Nonlinear Equations

(in 1D)

Numerical Analysis Spring 2021

A. Donev, Courant

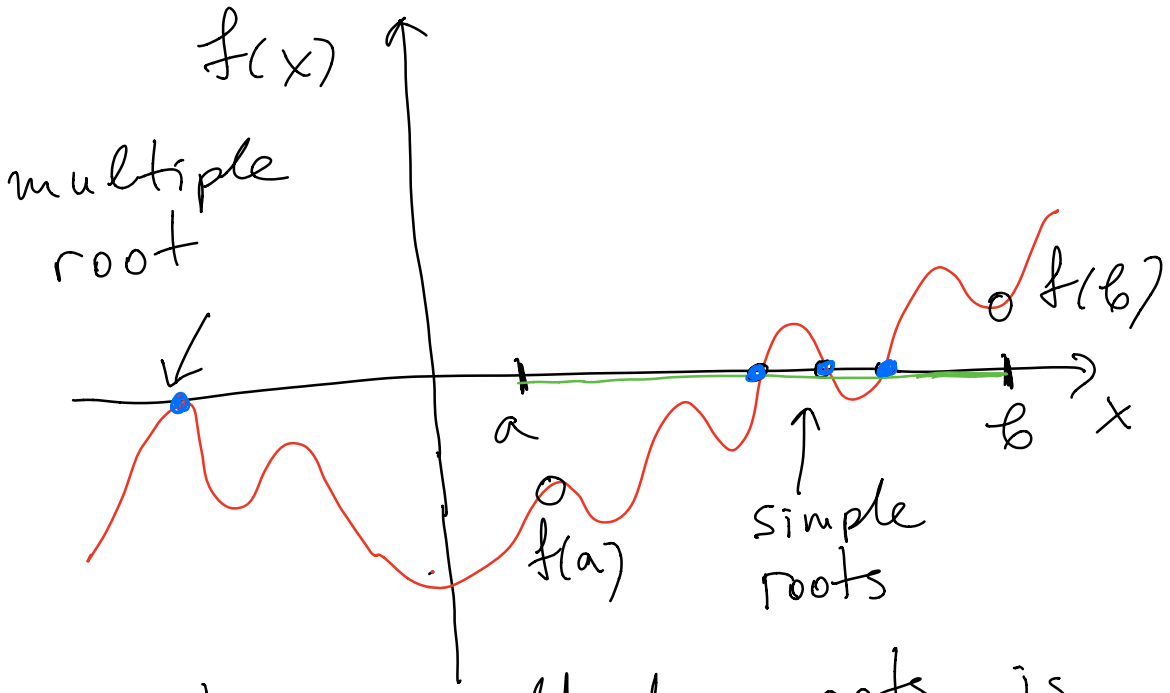
Consider solving a nonlinear equation like

$$\cos x + x^2 - 7 = 5$$

No closed form solution exists, so the only thing we can do is to solve numerically (on a computer)

Solve $f(x) = 0$ on $x \in [a, b]$

By this we mean find as accurately as possible at least one solution in $[a, b]$, assuming one exists.



Finding multiple roots is harder, so we focus on simple roots & assume there is at least one root in $[a,b]$ & f is continuous

Theorem If f is continuous on $[a,b]$ and $f(a)\cdot f(b) < 0$ (i.e., $f(a)$ & $f(b)$ have opposite sign) then $\exists\, x \in (a,b)$ s.t. $f(x)=0$

"Proof": It's "obvious" but if you insist apply the Intermediate Value Theorem $\square$

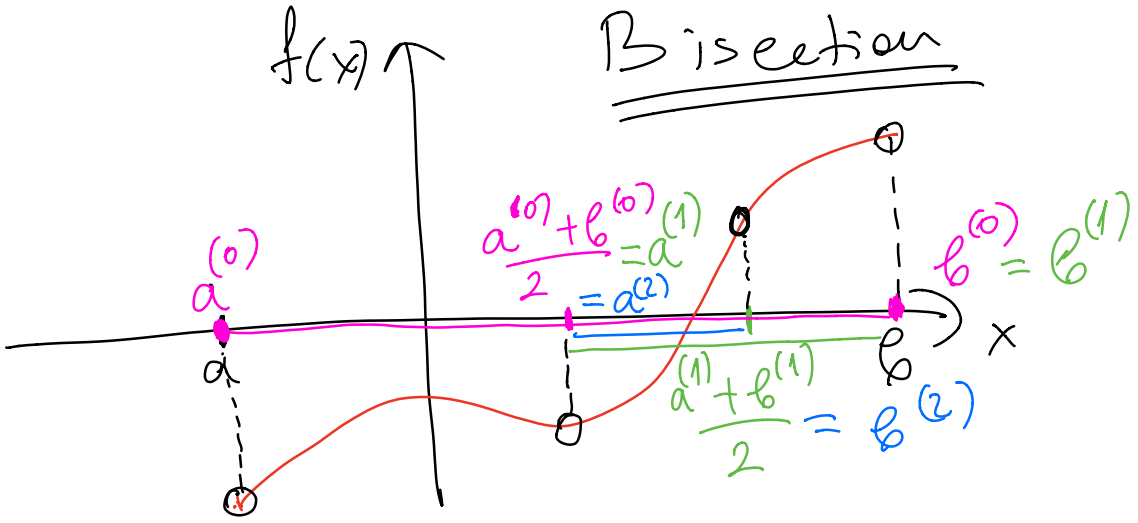
Now, assume that we are given an $f(x)$, meaning, given a real x we can evaluate $f(x)$ using some computer code, and an interval $[a,b]$ such that $f(a) \cdot f(b) < 0$

(Q1) Why don't I say $f(a)f(b) \leq 0$?

Can we devise an algorithm that will give us a root of f on $[a,b)$?

Answer: Yes, use bisection

Bisection



Algorithm Bisection

Input $[a, b]$, $n \in \mathbb{Z}^+$

Output x such that $f(x) \approx 0$ (approximately equal)

Set $a^{(0)} = a$, $b^{(0)} = b$

For $k = 0, 1, 2, \ldots, n-1$ do

$$x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}$$

$$f^{(k)} = f(x^k)$$

If $f^{(k)} f(a^{(k)}) < 0$

$b^{(k+1)} = x^{(k)}$; $a^{(k+1)} = a^{(k)}$

else

$b^{(k+1)} = b^{(k)}$; $a^{(k+1)} = x^{(k)}$

end if

end for

Return $x \approx x^{(n)} = \frac{a^{(n)} + b^{(n)}}{2}$

or $x \in [a^{(n)}, b^{(n)}]$

[Matlab demo Bisection.m]

How accurate is x ?

What is absolute error

$\varepsilon^{(n)} = |x^{(n)} - x|$

where $f(x) = 0$

We know

$$x \in \left[a^{(n)}, b^{(n)} \right]$$

$$\Rightarrow \quad |x^{(n)} - x| \leq \frac{b^{(n)} - a^{(n)}}{2}$$

Observe: $b^{(n)} - a^{(n)} = \frac{b^{(0)} - a^{(0)}}{2^n}$

$$\Rightarrow \quad \underbrace{\varepsilon^{(n)}}_{\text{error estimate}} = |x^{(n)} - x| \leq \frac{b-a}{2^{n+1}}$$

This means the error approximately halves each iteration. This is guaranteed, which means the method is very robust, but it's not very fast.

given ε $< \dfrac{b-a}{2^{n+1}}$

$$2^{n+1} < \frac{b-a}{\varepsilon}$$

$$n+1 = \left\lfloor \log_2\left(\frac{b-a}{\varepsilon}\right) \right\rfloor$$

"floor" in Matlab

$$\Rightarrow \quad n = \left\lceil \log_2\left(\frac{b-a}{\varepsilon}\right) \right\rceil$$

ceil in Matlab

So given a target error tolerance ε we can directly estimate how many iterations to take.

[Numerical analyst job complete]

Newton & Secant methods

Can we do better than bisection? Bisection only used the sign of f, not the value of $f(x)$.

"Definition": A function $f(x)$ is a smooth function on $[a,b]$ if it can be approximated well by a low order polynomial (linear, quadratic, cubic) over the whole interval $[a,b]$

In most of this class, we will assume to work with functions that are sufficiently smooth. This is more than just being sufficiently differentiable, which we may require as an assumption. Smoothness also requires high-order derivatives not to be very large. In some sense

[See Functional Analysis]

Our error / convergence estimates will only be accurate for smooth $f(x)$.

Note: If $f(x)$ is continuously differentiable, it will be almost linear over a sufficiently small interval $[a, b]$
(think calculus as $b-a \to dx$)

Idea for "all" of NA:
Approximate smooth functions by low-degree polynomials

If $f(x) \approx p(x)$ on $[a, b]$
then we can solve

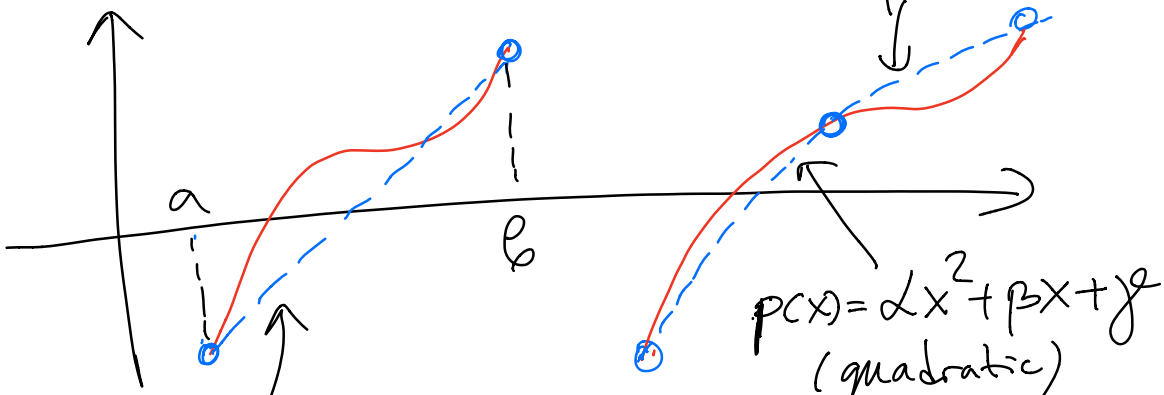
$$p(x) = 0$$

instead. If p is a linear or quadratic function then we have explicit solution!

How do we approximate $f(x)$ by $p(x)$?

Two ideas common in all of numerical analysis:

(1) Evaluate $f(x)$ at a few points & fit a polynomial through them:



$$p(x) = f(a) + \frac{x-a}{b-a} f(b)$$

↑ linear

(2) Approximate $f(x)$ by it's Taylor series around the current guess for the answer

---

Reminder

If $f(x)$ is infinitely differentiable at x_0,

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$$

$$= f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 + \dots$$

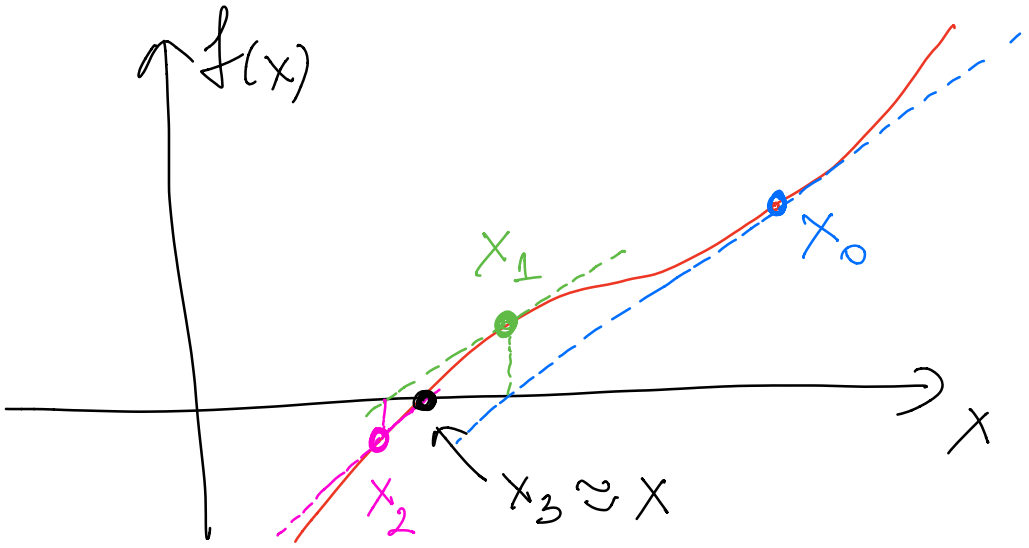
Taylor series with remainder

$$f(x) = \sum_{n=0}^{k} \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + \frac{f^{(k+1)}(\xi)}{(k+1)!}(x-x_0)^{n+1}$$

for some ξ between x_0 & x

Newton's method

Approximate $f(x)$ by it's tangent at x_k, i.e., first-order Taylor series



$$f(x) \simeq f_k(x) = f(x_k) + f'(x_k)(x - x_k)$$

Solve $f_k(x) = 0 \Rightarrow$

$$x \simeq x_k - \frac{f(x_k)}{f'(x_k)}$$

Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} .$$

E.g. $f(x) = x^2 - c$

(square root calculation)

$f'(x) = 2x$ $\Rightarrow$

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

$$= \frac{x_k}{2} + \frac{c}{2x_k}$$

which is the Babylonian method!

Remember that the method converged fast. Why?

General idea: When analyzing the error / convergence of method based on Taylor series, use Remainder Theorem

Error $e_k = x_k - x$

$$f(x) = 0 = f(x_k) + (x - x_k) f'(x_k)$$

remainder $\longrightarrow + \frac{1}{2}(x - x_k)^2 f''(\xi)$

ξ between x_k and x

Divide both sides by $f'(x_k) \neq 0$

& rearrange

$$\left(x_k - \frac{f(x_k)}{f'(x_k)} \right) - x = -\frac{1}{2}(x - x_k)^2 \frac{f''(\xi)}{f'(x_k)}$$

$$x_{k+1} - x = e_{k+1} = -\frac{1}{2} e_k^2 \cdot \frac{f''(\xi)}{f'(x_k)}$$

$$\frac{e_{k+1}}{e_k^2} = -\frac{f''(\xi)}{2 f'(x_k)}$$

If the method converges,

$$x_k \to x$$

and $\xi \to x \quad \Rightarrow$

$$\frac{|e_{k+1}|}{e_k^2} \to \frac{f''(x)}{2f'(x)} = C$$

$$\Rightarrow |e_{k+1}| \approx C \cdot e_k^2$$

Error gets squared every iteration, not halved like for bisection. This leads to fast convergence after the error is small

If $e_k << 1 \Rightarrow e_k^2 << e_k$

($<<$: much less than)

This explains why the Babylonian method converged fast.

But when does Newton method converge?

General answer hard to give but we can give a sufficient condition

We want $|e_{k+1}| < |e_k|$

(error decreases)

$$\left|\frac{f''(\xi)}{2f'(x_k)}\right| e_k^2 < |e_k|$$

If $|e_k| < \left|\frac{2f'(x_k)}{f''(\xi)}\right|$

then error decreases

Imagine that

$$\left| \frac{f''(y_1)}{f'(y_2)} \right| \leq A \qquad \forall y_1, y_2 \in I$$

$$I = [x - \delta, x + \delta]$$

(see theorem 1.8 in Suli)

and $x_0 \in I \Rightarrow$

$$|e_0| = |x_0 - x| < \delta$$

If $|e_0| < \frac{1}{A} < \left| \frac{f'(x_1)}{f''(\xi)} \right|$

$$\Rightarrow |e_1| < \frac{|e_0|}{2}$$

and error decreases strictly by at least half and eventually much faster

Conclusion:

If we give Newton's method an initial guess closer to a simple root x than $\approx \left| \frac{f'(x)}{f''(x)} \right|$, it will converge rapidly.

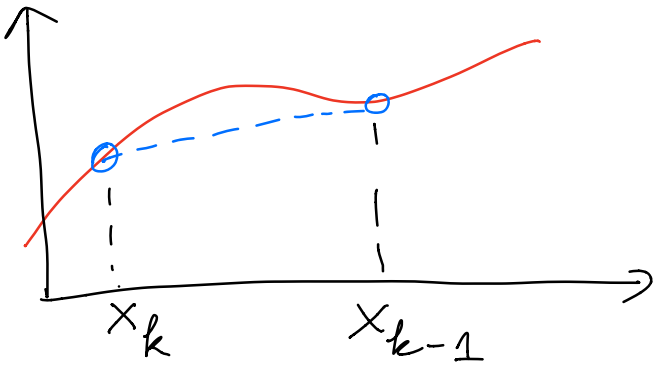
Otherwise, it may converge very slowly, erratically, or diverge.

Idea: Use bisection first to find an initial guess that is good & then switch to Newton

This is called safe guarded Newton method but we will not go into the details in class.

Secant method

Newton's method was based on approxmating $f(x)$ with its first-order Taylor series (tangent). How about approximating $f(x)$ based on two points?



$$f(x) \approx p(x) = f(x_k) + \frac{x - x_k}{x_{k-1} - x_k}\left(f(x_{k-1}) - f(x_k)\right)$$

Check: $\begin{cases} p(x_k) = f(x_k) \\ p(x_{k-1}) = f(x_{k-1}) \end{cases}$

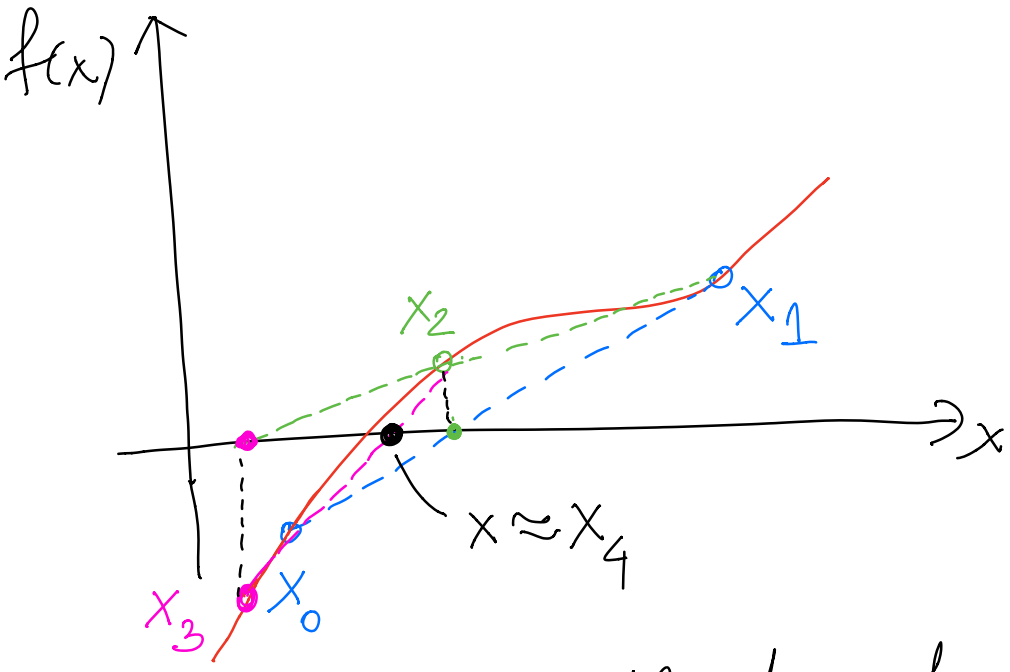
$$f(x_{k+1}) \approx p(x_{k+1}) = 0$$

Solve for x_{k+1} (linear eq.)

$$x_{k+1} = x_k - f_k\left(\frac{x_k - x_{k-1}}{f_k - f_{k-1}}\right)$$

Secant method

Here $f_k = f(x_k)$ (notation)



One way to think of secant method is as Newton's method with the approximation

$$f'(x_k) \approx \frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k}$$

which is good if f is smooth and $x_{k-1} \approx x_k$.

Called finite-difference approximation

Note we need two points to start the iteration.

It is therefore very natural to combine secant with bisection: instead of using midpoint use secant point as new guess.

This method of Regula Falsi is guaranteed to converge to a root but maybe it is faster than bisection?

[Demo Secant.m]

Conclusion: True Secant converges faster than bisection or Regula Falsi.

But how much faster?

Bisection: $|e_{k+1}| \leq \frac{|e_k|}{2}$ (linear)

Newton: $|e_{k+1}| \to \left|\frac{f''(x)}{2f'(x)}\right| |e_k|^2$ (quadratic)

Secant (we will not prove):

$$|e_{k+1}| \to M \cdot |e_k|^q$$

where $q = \frac{1}{2}(1+\sqrt{5}) \approx 1.6$ is Golden Ratio

We say bisection converges at a linear rate, Newton at quadratic rate, and secant at

order of convergence ≈ 1.6

Aside: For secant method, one can prove (see (4.17) in "Practice" book)

$$|e_{k+1}| \to \left|\frac{f''(x)}{2f'(x)}\right| |e_k e_{k-1}|$$

which makes the relationship with Newton's method more obvious.

Homework

Try & (maybe) analyze Steffensen's method

$$x_{k+1} = x_k - \frac{f_k^2}{f(x_k + f_k) - f_k}$$