3. Solution of Nonlinear Equations

- Introduction
- Bisection Method
- Newton's Method
- Secant Method
- Fixed Points and Functional Iteration
- Computing Roots of Polynomials
- Homotopy and Continuation Methods

Definition and examples

x is a *zero* (or *root*) of a function f if $f(x) = 0$

examples

- $f(x) = e^x$ has no zeros
- $f(x) = e^x - e^{-x}$ has one zero
- $f(x) = e^x - e^{-x} - 3x$ has three zeros
- $f(x) = \cos x$ has infinitely many zeros

cf., one linear equation in one variable $ax = b$

- a unique solution if $a \neq 0$
- no solution if $a = 0$, $b \neq 0$
- any $x \in \mathbf{R}$ is a solution if $a = b = 0$

Characteristics of algorithms for nonlinear equations

how f is described

- user provides subroutine to compute $f(x)$ (and possibly $f'(x)$) at x
- called 'black box' or 'oracle' model for describing f
- evaluating f and f' can be expensive ($e.g.$, require a circuit simulation)

limitations of algorithms

- there exist no algorithms that are guaranteed to find all solutions
- most algorithms find at most one solution
- need prior information from the user: $e.g.$, an interval that contains a zero, or a point near a solution

methods for solving nonlinear equations are iterative

- generate a sequence of points $x^{(k)}$, $k = 0, 1, 2, \ldots$ that converge to a solution; $x^{(k)}$ is called the kth *iterate*; $x^{(0)}$ is the *starting point*
- computing $x^{(k+1)}$ from $x^{(k)}$ is called one *iteration* of the algorithm
- each iteration typically requires one evaluation of f (or f and f') at $x^{(k)}$
- algorithms need a stopping criterion, $e.g.$, terminate if

$$|f(x^{(k)})| \leq \text{specified tolerance}$$

- speed of the algorithm depends on:
  - the cost of evaluating $f(x)$ (and possibly, $f'(x)$)
  - the number of iterations

Analyzing speed of convergence

suppose $x^{(k)} \to x^\star$ with $f(x^\star) = 0$; how fast does $x^{(k)}$ go to $x^\star$?

error after k iterations:

- **absolute error**: $|x^{(k)} - x^\star|$
- **relative error**: $|x^{(k)} - x^\star|/|x^\star|$ (defined if $x^\star \neq 0$)
- **number of correct digits:**

$$\left\lfloor -\log_{10}\left(\frac{|x^{(k)} - x^\star|}{|x^\star|}\right)\right\rfloor$$

(defined if $x^\star \neq 0$ and $|x^{(k)} - x^\star|/|x^\star| \leq 1$)

rates of convergence of a sequence $x^{(k)}$ with limit $x^\star$

- linear convergence: there exists a $c \in (0,1)$ such that

$$|x^{(k+1)} - x^\star| \leq c\,|x^{(k)} - x^\star| \quad \text{for sufficiently large } k$$

- R-linear convergence: there exists $c \in (0,1)$, $M > 0$ such that

$$|x^{(k)} - x^\star| \leq M c^k \quad \text{for sufficiently large } k$$

- quadratic convergence: there exists a $c > 0$ s.t.

$$|x^{(k+1)} - x^\star| \leq c\,|x^{(k)} - x^\star|^2 \quad \text{for sufficiently large } k$$

- superlinear convergence: there exists a sequence c_k with $c_k \to 0$ s.t.

$$|x^{(k+1)} - x^\star| \leq c_k\,|x^{(k)} - x^\star| \quad \text{for sufficiently large } k$$

interpretation (if $x^\star \neq 0$): let

$$r^{(k)} = -\log_{10}(\frac{|x^{(k)} - x^\star|}{|x^\star|})$$

($i.e.$, $r^{(k)} \approx$ the number of correct digits at iteration k)

- linear convergence: we gain roughly $-\log_{10} c$ correct digits per step

$$r^{(k+1)} \geq r^{(k)} - \log_{10} c$$

- quadratic convergence: for k sufficiently large, number of correct digits roughly doubles in one step

$$r^{(k+1)} \geq -\log(c|x^\star|) + 2r^{(k)}$$

- superlinear convergence: number of correct digits gained per step increases with k

$$r^{(k+1)} - r^{(k)} \to \infty$$

examples (with $x^\star = 1$)

- $x^{(k)} = 1 + 0.5^k$ converges linearly (with $c = 1/2$):

$$\frac{|x^{(k+1)} - 1|}{|x^{(k)} - 1|} = \frac{2^k}{2^{k+1}} = \frac{1}{2}$$

- $x^{(k)} = 1 + 0.5^{2^k}$ converges quadratically (with $c = 1$)

$$\frac{|x^{(k+1)} - 1|}{|x^{(k)} - 1|^2} = \frac{(2^{2^k})^2}{2^{2^{k+1}}} = 1$$

- $x^{(k)} = 1 + (1/(k+1))^k$ converges superlinearly

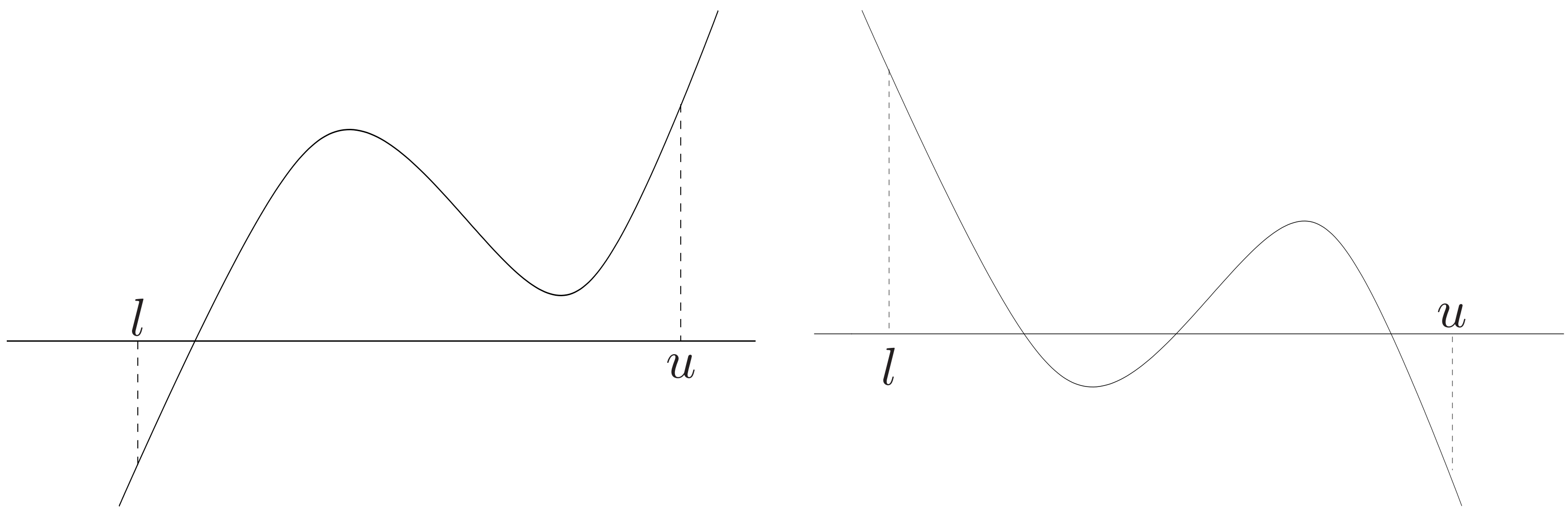
$$\frac{|x^{(k+1)} - 1|}{|x^{(k)} - 1|} = \frac{(k+1)^k}{(k+2)^{k+1}} \rightarrow 0$$

k	$1+0.5^k$	$1+0.5^{2^k}$	$1+(1/(k+1)^k)$
0	2.00000000000000	1.50000000000000	2.00000000000000
1	1.50000000000000	1.25000000000000	1.50000000000000
2	1.25000000000000	1.06250000000000	1.11111111111111
3	1.12500000000000	1.00390625000000	1.01562500000000
4	1.06250000000000	1.00001525878906	1.00160000000000
5	1.03125000000000	1.00000000023283	1.00012860082305
6	1.01562500000000	1.00000000000000	1.00000849985975
7	1.00781250000000	1.00000000000000	1.00000047683716
8	1.00390625000000	1.00000000000000	1.00000002323057
9	1.00195313125000	1.00000000000000	1.00000000100000
10	1.00097656250000	1.00000000000000	1.00000000003855

- sequence 1: we gain roughly $-\log_{10}(c) = 0.3$ correct digits per step
- sequence 2: number of correct digits roughly doubles at each step
- sequence 3: number of correct digits gained per step increases slowly (from 0.5 initially to 2 near the end)

Bisection method

$f : \mathbf{R} \rightarrow \mathbf{R}$, continuous



if $f(l)f(u) < 0$, then the interval $[l, u]$ contains at least one zero

Intermediate Value Theorem: Let $f \in \mathbf{C}([a, b])$ and assume p is a value between $f(a)$ and $f(b)$, that is

$$f(a) \leq p \leq f(b), \quad \text{or} \quad f(b) \leq p \leq f(a)$$

then there exists a point $c \in [a, b]$ for which $f(c) = p$

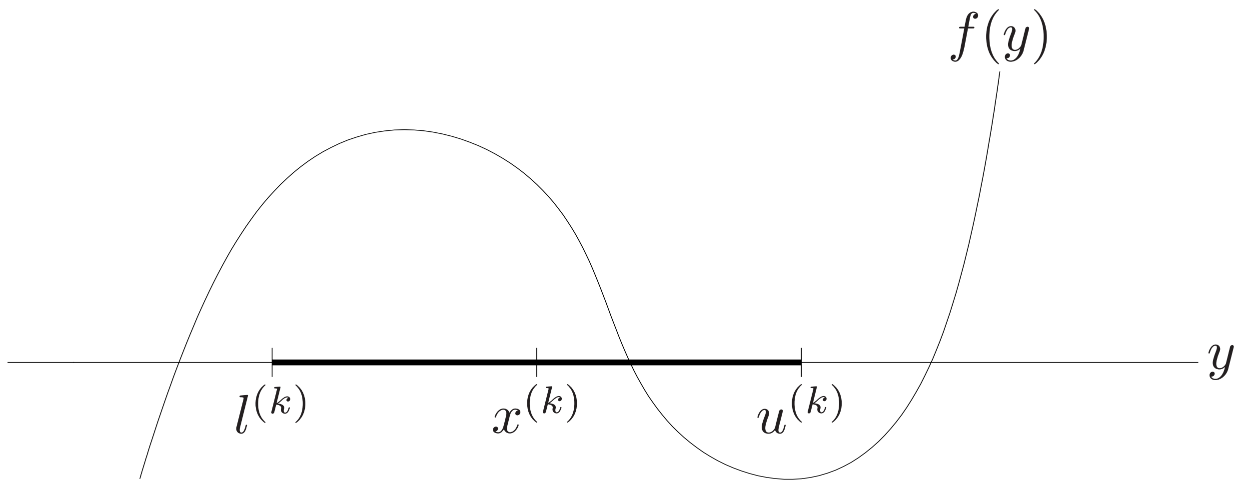
idea sketch

to find $x^\star$, let x be the midpoint of $[l, u]$

$$x = \frac{1}{2}(u + l)$$

assume $f(l) \neq 0$, then there are three possibilities:

1. $f(l)f(x) < 0 \Longrightarrow x^\star$ is between l and x
2. $f(l)f(x) > 0 \Longrightarrow x^\star$ is between x and u
3. $f(l)f(x) = 0 \Longrightarrow f(x) = 0$ and $x^\star = x$



given l, u with $l < u$ and $f(l)f(u) < 0$; a required tolerance $\delta, \epsilon > 0$
repeat

1. $x := (l + u)/2$.
2. Compute $f(x)$.
3. **if** $f(x) = 0$, **return** x.
4. **if** $f(x)f(l) < 0$, $u := x$, **else**, $l := x$.

until $u - l < \epsilon$ or $|f(x)| < \delta$

one function evaluation per iteration

remarks

- to avoid numerical error, calculate midpoint by $x = l + (u - l)/2$
- effectively determine $f(l)f(x) < 0$ via

$$\mathbf{sign}(f(l)) \neq \mathbf{sign}(f(x))$$

  since the multiplication could cause an underflow or overflow

- always put a maximum number of steps to avoid an infinite loop

convergence rate

- $u^{(k)} - l^{(k)}$ measures our uncertainty in localizing a zero $x^\star$:

$$|x^{(k)} - x^\star| \leq u^{(k)} - l^{(k)}$$

- uncertainty is halved at each iteration:

$$\begin{aligned} u^{(k)} - l^{(k)} &= \left(\frac{1}{2}\right)^k (u^{(0)} - l^{(0)}) \\ |x^{(k)} - x^\star| &\leq \left(\frac{1}{2}\right)^k (u^{(0)} - l^{(0)}) \end{aligned}$$

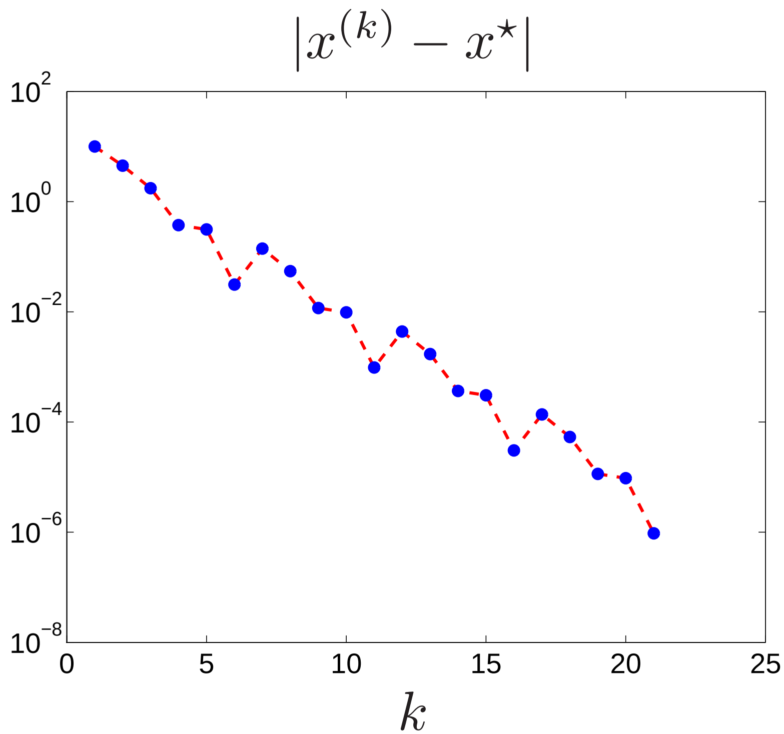
  $i.e.$, R-linear convergence with $c = 1/2$, $M = u^{(0)} - l^{(0)}$

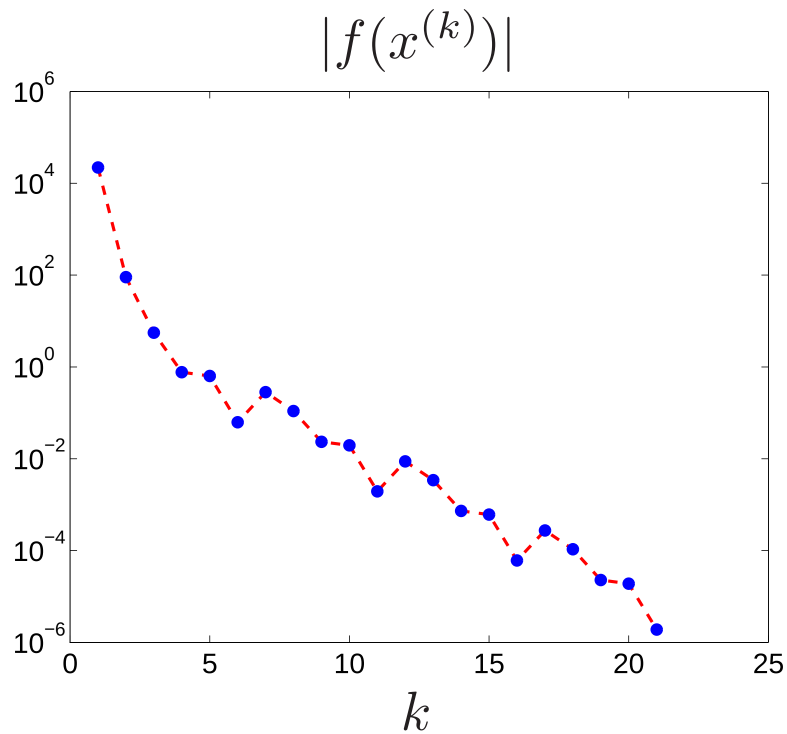
- number of iterations required for $u^{(k)} - l^{(k)} \leq \epsilon$ or $|x^{(k)} - x^\star| \leq \epsilon$:

$$k \ \geq \ \log_2 \frac{u^{(0)} - l^{(0)}}{\epsilon}$$

example: $f(x) = e^x - e^{-x}$

- unique zero $x^\star = 0$
- start bisection method with $l = -1$, $u = 21$





conclusions

- the bisection method is also known as *the method of interval halving*
- bisection is known as a *global* method, $i.e.$, always converges no matter how far you start from the actual root
- it cannot find roots when the function is tangent to the axis
- convergence is slow compared to other methods

Newton's method

$f : \mathbf{R} \to \mathbf{R}$, differentiable

---

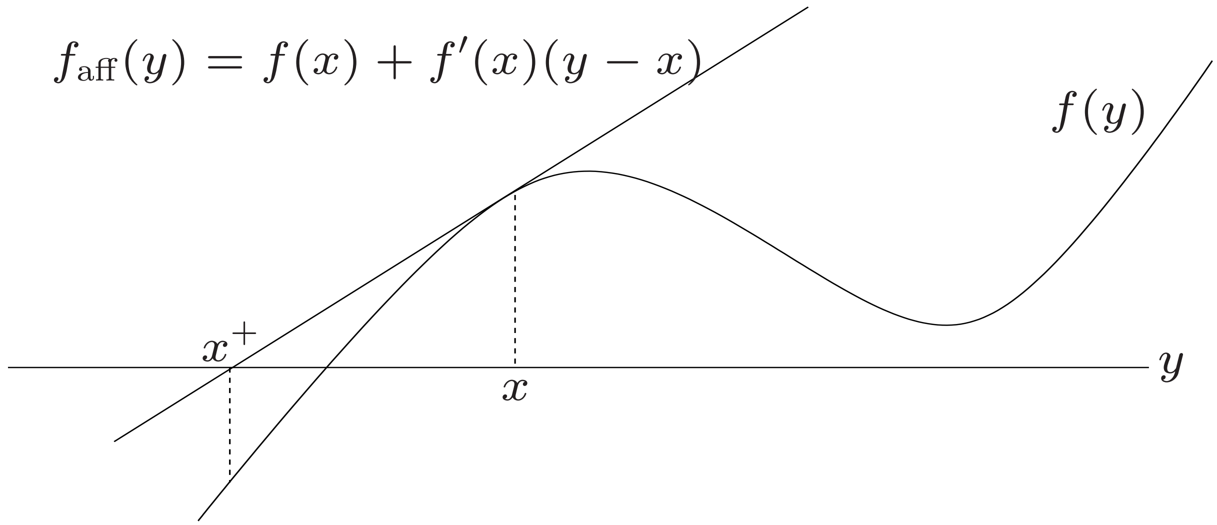
given initial x, required tolerance $\epsilon > 0$
repeat

1. Compute $f(x)$ and $f'(x)$.
2. **if** $|f(x)| \leq \epsilon$, **return** x.
3. $x := x - f(x)/f'(x)$.

until maximum number of iterations is exceeded.

- each iteration requires one evaluation of f and f'
- there exist other (more sophisticated) stopping criteria
- we assume $f'(x^{(k)}) \neq 0$, all k

interpretation (with notation $x = x^{(k)}$, $x^+ = x^{(k+1)}$)



- make affine approximation of f around x using Taylor series expansion:

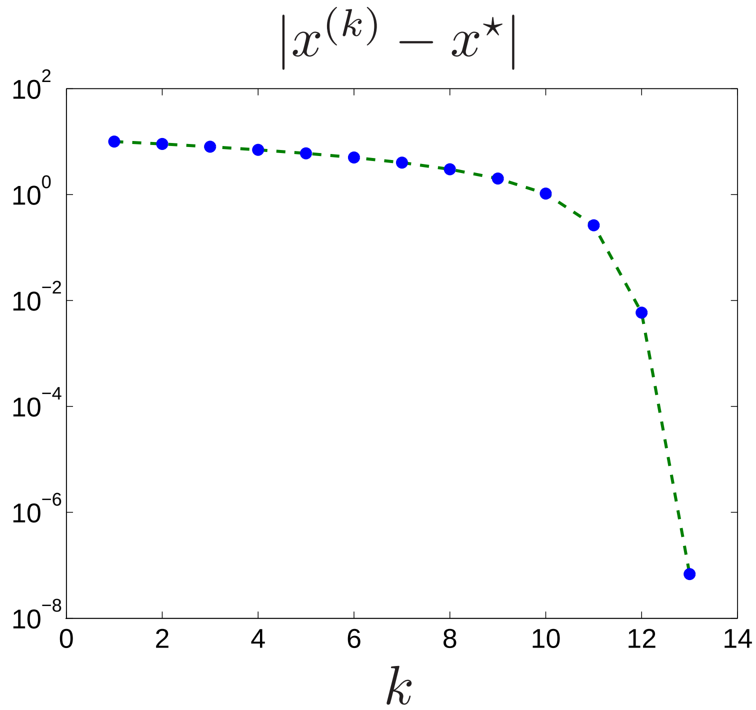
$$f_{\text{aff}}(y) = f(x) + f'(x)(y - x)$$

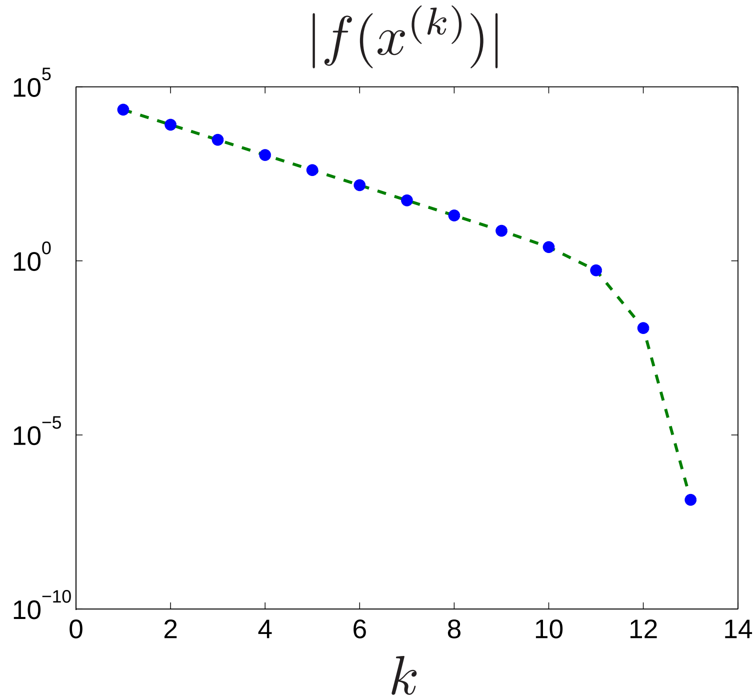
- solve the linearized equation $f_{\text{aff}}(y) = 0$ and take the solution y as x^+:

$$x^+ = x - f(x)/f'(x)$$

Examples

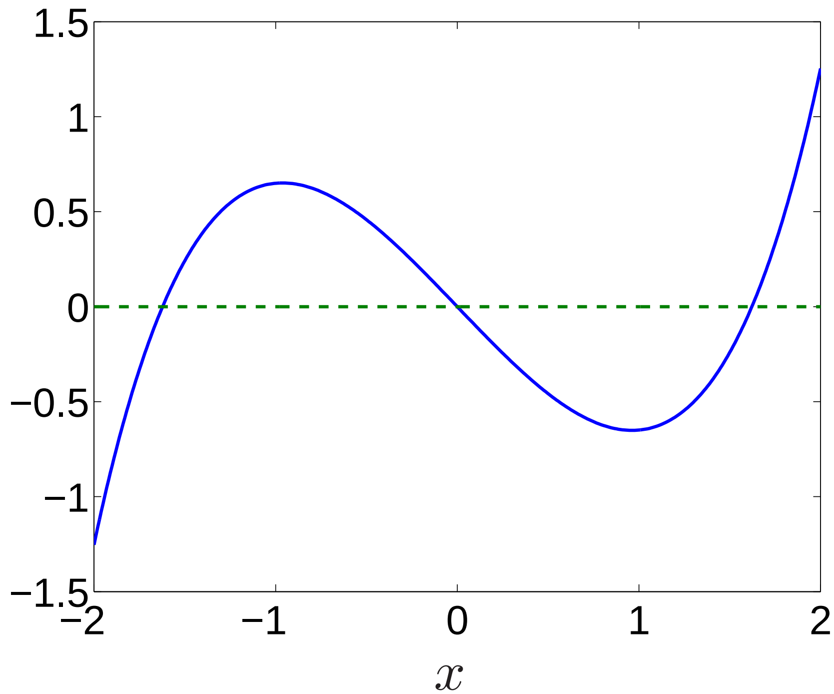
- $f(x) = e^x - e^{-x}$, start at $x^{(0)} = 10$





asymptotic convergence is much faster than bisection method

- $f(x) = e^x - e^{-x} - 3x$

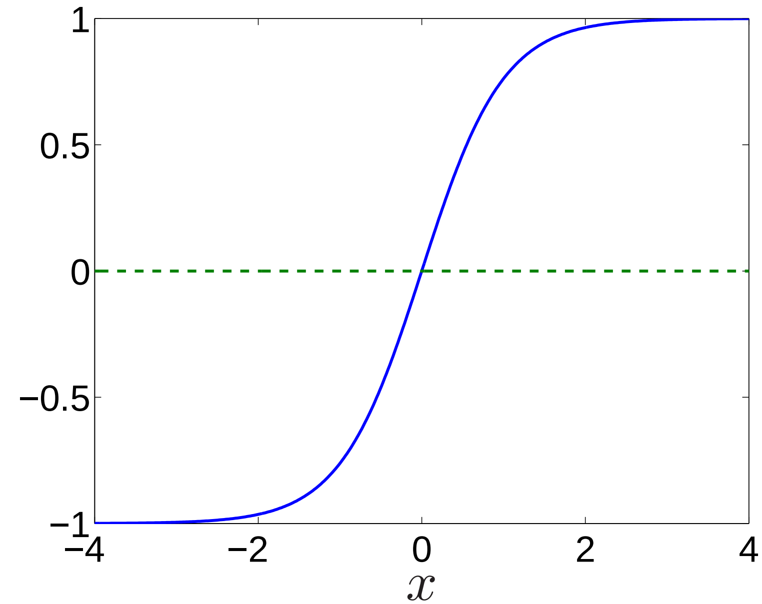


  - start at $x^{(0)} = -1$: converges to $x = -1.62$
  - start at $x^{(0)} = -0.8$: converges to $x = 1.62$
  - start at $x^{(0)} = -0.7$: converges to $x = 0$

  converges to a different solution depending on the starting point

- $f(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ (unique root at $x = 0$)
  - start at $x^{(0)} = 0.9$:

$$\begin{aligned} x^{(1)} &= -5.7\ 10^{-1} \\ x^{(2)} &= 1.3\ 10^{-1} \\ x^{(3)} &= -1.6\ 10^{-3} \\ x^{(4)} &= 2.5\ 10^{-9} \\ x^{(5)} &= -3.0\ 10^{-17} \end{aligned}$$



converges very rapidly

  - start at $x^{(0)} = 1.1$:

$$x^{(1)} = 1.1\ 10^0, \qquad x^{(2)} = 1.2\ 10^0, \qquad x^{(3)} = -1.7\ 10^0,$$
$$x^{(4)} = 5.7\ 10^0, \qquad x^{(5)} = -2.3\ 10^4$$

does not converge

error analysis: let $f \in \mathbf{C}^2([a,b])$ with $f(x^\star) = 0$ for $x^\star \in [a,b]$

- expand f in a Taylor series about $x = x^{(k)}$ and evaluate at $x = x^\star$

$$0 = f(x^\star) = f(x^{(k)}) + (x^\star - x^{(k)})f'(x^{(k)}) + \frac{1}{2}(x^\star - x^{(k)})^2 f''(\xi^{(k)})$$

  where $\xi^{(k)}$ is between $x^{(k)}$ and $x^\star$

- divide both sides by $f'(x^{(k)})$ and re-arrange, we have

$$x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} - x^\star = \frac{1}{2}(x^\star - x^{(k)})^2 \frac{f''(\xi^{(k)})}{f'(x^{(k)})}$$

- assume the convergence; $f'(x^{(k)}) \approx f'(x^\star)$ and $f''(\xi^{(k)}) \approx f''(x^\star)$

$$x^{(k+1)} - x^\star \;\approx\; \frac{1}{2}(x^\star - x^{(k)})^2 \frac{f''(x^\star)}{f'(x^\star)}$$

  the error at one step is like the *square* of the error at the previous step

Theorem of Newton's Method

assume

- $f \in \mathbf{C}^2(I)$ and I is an open interval
- $f(x^\star) = 0$ for some $x^\star \in I$ and that $f'(x^\star) \neq 0$
- $x^{(k)}$ is defined by the Newton's iteration

then for $x^{(0)}$ sufficiently close to $x^\star$ we have that

$$\lim_{k \to \infty} x^{(k)} = x^\star$$

and

$$\lim_{k \to \infty} \frac{x^\star - x^{(k+1)}}{(x^\star - x^{(k)})^2} = -\frac{f''(x^\star)}{f'(x^\star)}$$

Proof.

- define J a ball around $x^\star$ with radius $\epsilon > 0$

$$J = \{x \mid |x - x^\star| \leq \epsilon\}$$

  with ϵ small enough so that $J \subset I$ and f is not vanished on J

- J is closed and f'' is continous on J; there exists c such that

$$c = \frac{\max_{x\in J} |f''(x)|}{2 \min_{x\in J} |f'(x)|} \quad \text{and} \quad c < \infty$$

- since $\xi^{(0)} \in J$, the Newton error for the first iteration satisfies

$$|x^\star - x^{(1)}| \leq |x^\star - x^{(0)}|^2 \frac{f''(\xi^{(0)})}{2f'(x^{(0)})} \quad \leq \quad c|x^\star - x^{(0)}|^2$$

- choose $x^{(0)}$ so that $|x^\star - x^{(0)}| < 1/c$, then we have

$$|x^\star - x^{(1)}| \leq c|x^\star - x^{(0)}|^2 \ < \ |x^\star - x^{(0)}|$$

  which forces $x^{(1)} \in J$

- apply the Newton's method recursively; the entire sequences is in J and

$$|x^{(k+1)} - x^\star| \ = \ \frac{1}{2}(x^\star - x^{(k)})^2 \frac{|f''(\xi^{(k)})|}{|f'(x^{(k)})|} \ \leq \ c|x^\star - x^{(k)}|^2$$

  which shows the *quadratic convergence*

- define the error $e^{(k)} = x^\star - x^{(k)}$; we can show

$$|e^{(k)}| \leq (1/c)(ce^{(0)})^{2^k}$$

  and $e^{(k)} \to 0$ as $k \to \infty$ provided that

$$ce^{(0)} = c|x^\star - x^{(0)}| < 1 \qquad (x^{(0)} \text{ is closed enough to } x^\star)$$

- $x^{(k)} \to x^\star$ and $\xi^{(k)} \to x^\star$ (since $\xi^{(k)}$ is between $x^\star$ and $x^{(k)}$)
- continuity on f shows that

$$
\begin{aligned}
\lim_{k\to\infty} \frac{x^\star - x^{(k+1)}}{(x^\star - x^{(k)})^2} &= -\lim_{k\to\infty} \frac{f''(\xi^{(k)})}{f'(x^{(k)})} \\
&= -\frac{f''(\lim_{k\to\infty} \xi^{(k)})}{2f'(\lim_{k\to\infty} x^{(k)})} \\
&= \frac{f''(x^\star)}{f'(x^\star)}
\end{aligned}
$$

conclusions

- Newton's method works very well if we start near a solution
- it may not work at all if we start too far from a solution
- if there are multiple solutions, it may converge to a different solution depending on the starting point; it does not necessarily converge to the solution closest to the starting point
- also known as Newton–Raphson Iteration
- convergence is quadratic (only a few iterations required to get solution close to root)
- Newton's method is combined with other slower methods to ensure convergence

Computation of the Square Root

given a a positive number, finding $\sqrt{a}$ is equivalent to finding a root of

$$f(x) = x^2 - a = 0$$

applying the Newton's iteration to $f(x)$ gives

$$x^{(k+1)} = \frac{1}{2}\left(x^{(k)} + \frac{a}{x^{(k)}}\right)$$

if we pick $x^{(0)} > 0$ then the relative error satisfies

$$\left|\frac{x^{(k+1)} - \sqrt{a}}{\sqrt{a}}\right| \leq 2\left(\frac{x^{(0)} - \sqrt{a}}{2\sqrt{a}}\right)^{2^k}$$

the error decreases very rapidly

Proof. the Newton error equation is

$$x^{(k+1)} - \sqrt{a} = (x^{(k)} - \sqrt{a})^2 \frac{f''(\xi^{(k)})}{f'(x^{(k)})} = \frac{(x^{(k)} - \sqrt{a})^2}{2x^{(k)}}$$

so the relative error satisfies

$$\left| \frac{x^{(k+1)} - \sqrt{a}}{\sqrt{a}} \right| = \left(\frac{x^{(k)} - \sqrt{a}}{\sqrt{a}} \right)^2 \left| \frac{\sqrt{a}}{2x^{(k)}} \right|$$

- if $x^{(0)} > 0$, the Newton iteration gives $x^{(k)} > 0$ for all k
- the error equation says that $\sqrt{a} \leq x^{(k)}$ for all $k \geq 1$
- hence, $|\sqrt{a}/x^{(k)}| \leq 1$ and from the relative error equation, we have

$$\left| \frac{x^{(k+1)} - \sqrt{a}}{\sqrt{a}} \right| \leq \frac{1}{2} \left(\frac{x^{(k)} - \sqrt{a}}{\sqrt{a}} \right)^2$$

- iterate the above inequality recursively, we get the desired result

example: compute $\sqrt{9} = 3$ using $x^{(0)} = 1$

the Newton's iteration

$$x^{(k+1)} = 0.5(x^{(k)} + 9/x^{(k)})$$

generate the sequences:

k	$x^{(k)}$
0	1.0000000000
1	5.0000000000
2	3.4000000000
3	3.0235294118
4	3.0000915541
5	3.0000000014
6	3.0000000000

get 10 digits correct by only 6 iterations

Newton's Method for Convex Function

convex function: f is convex if and only if $\mathbf{dom}(f)$ is convex and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y), \quad \forall x, y$$

with $0 \leq \theta \leq 1$

- if f is differentiable, then f is convex if and only if

$$f(y) \geq f(x) + f'(x)(y-x), \quad \forall y, x$$

- if f is twice differentiable, then f is convex if and only if

$$f''(x) \geq 0$$

- examples: e^{ax}, x, x^2, $|x|$, $-\log(x)$, $x\log(x)$, $\|x\|$

assumptions:

- $f \in \mathbf{C}^2(\mathbf{R})$
- f is increasing, *i.e.*, $f'(x) \geq 0$ for all x
- f is convex
- f has a zero at $x^\star$

Result: if f satisfies the above assumptions, then $x^\star$ is unique, and the Newton Iteration will converge to $x^\star$ from *any* starting point

to apply Newton method, we also assume $f' \neq 0$ for all x

the uniqueness of $x^\star$ is evident as f is increasing; cannot cross zero twice

Proof: the error equation of the Newton's iteration is

$$x^{(k+1)} - x^\star \;=\; \frac{1}{2}(x^\star - x^{(k)})^2 \frac{f''(\xi^{(k)})}{f'(x^{(k)})}$$

denote $e^{(k)} = x^{(k)} - x^\star$ the error at the kth iteration

- f is convex and increasing, so $f'' \geq 0$ and $f' > 0$ (assume $f' \neq 0$) $\forall x$
- the error equation says $x^{(k)} \geq x^\star$ for all $k \geq 1$ and since f is inscreasing,

$$f(x^{(k)}) \geq f(x^\star) = 0$$

- the Newton iterations:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad e^{(k+1)} = e^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

  says that both $e^{(k)}$ and $x^{(k)}$ are decreasing sequences

- moreover, $e^{(k)}$ and $x^{(k)}$ are bounded below (by 0 and $x^\star$)
- therefore, the limits of both sequences exist and given by

$$e^\star = \lim_{k\to\infty} e^{(k)}, \quad z = \lim_{k\to\infty} x^{(k)}$$

- take the limit to the Newton's iteration

$$\begin{aligned} \lim_{k\to\infty} e^{(k+1)} &= \lim_{k\to\infty} - \lim_{k\to\infty} \frac{f(x^{(k)})}{f'(x^{(k)})} \\ e^\star &= e^\star - \frac{f(z)}{f'(z)} \end{aligned}$$

- hence, $f(z) = 0$ and we can conclude that $z = x^\star$

Newton's Method for Systems of Nonlinear Equations

consider a function $f : \mathbf{R}^n \to \mathbf{R}^n$

let $x^\star = x + h$ and use the affine approximation of f about x

$$0 = f(x^\star) = f(x+h) \approx f(x) + Df(x)h$$

where $Df(x)$ is the Jacobian matrix of f, *i.e.*, $Df(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j}$

then, solve h from

$$h = -Df(x)^{-1} f(x)$$

provided that the Jacobian matrix is nonsingular

Newton's method is summarized by

$$x^{(k+1)} = x^{(k)} - [Df(x^{(k)})]^{-1} f(x^{(k)})$$

which follows the same treatment for single equation

Secant method

idea:

- Newton's method requires a formula for $f'(x)$
- use an approximation to the derivative in the Newton formula

$$f'(x^{(k)}) \approx \frac{f(x^{(k)} - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

the approximation comes directly from the definition of f' as a limit

- iteration for the secant method is

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \left(\frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} \right)$$

Secant Algorithm

$f : \mathbf{R} \to \mathbf{R}$, continuous

---

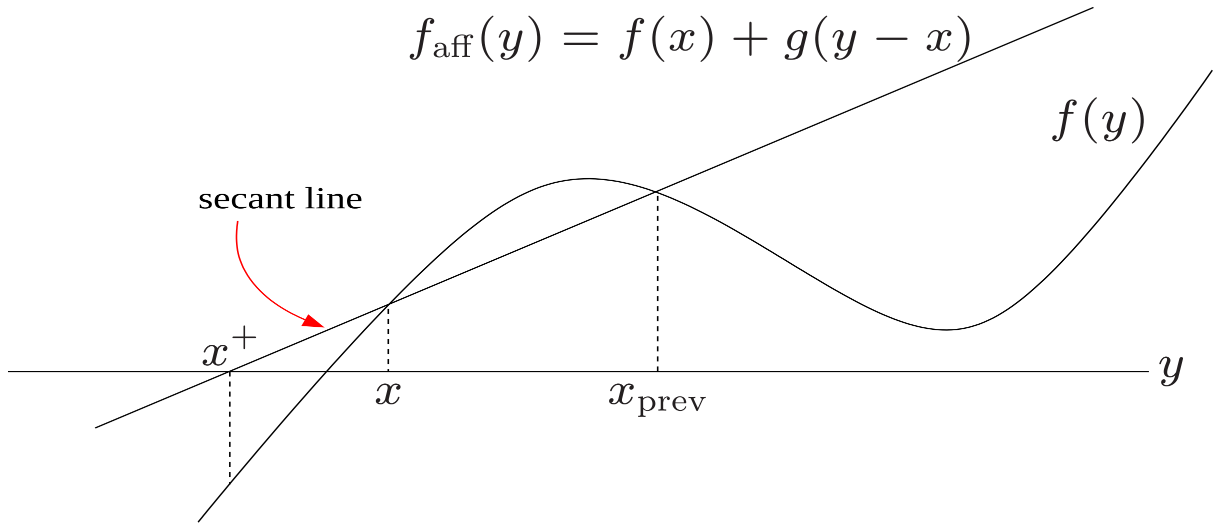
given two initial points x, x_{prev}, required tolerance $\epsilon > 0$
repeat

1. Compute $f(x)$
2. **if** $|f(x)| \le \epsilon$, **return** x.
3. $g := (f(x) - f(x_{\text{prev}}))/(x - x_{\text{prev}})$.
4. $x_{\text{prev}} := x$.
5. $x := x - f(x)/g$.

until maximum number of iterations is exceeded.

- first iteration requires two evaluations of f (at x and x_{prev})
- subsequent iterations require one evaluation (at x)
- we assume $g \neq 0$

interpretation (with notation: $x = x^{(k)}$, $x^+ = x^{(k+1)}$, $x_{\text{prev}} = x_{\text{prev}}^{(k)}$)



- affine approximation f_{aff} with $f_{\text{aff}}(x) = f(x)$, $f_{\text{aff}}(x_{\text{prev}}) = f(x_{\text{prev}})$:

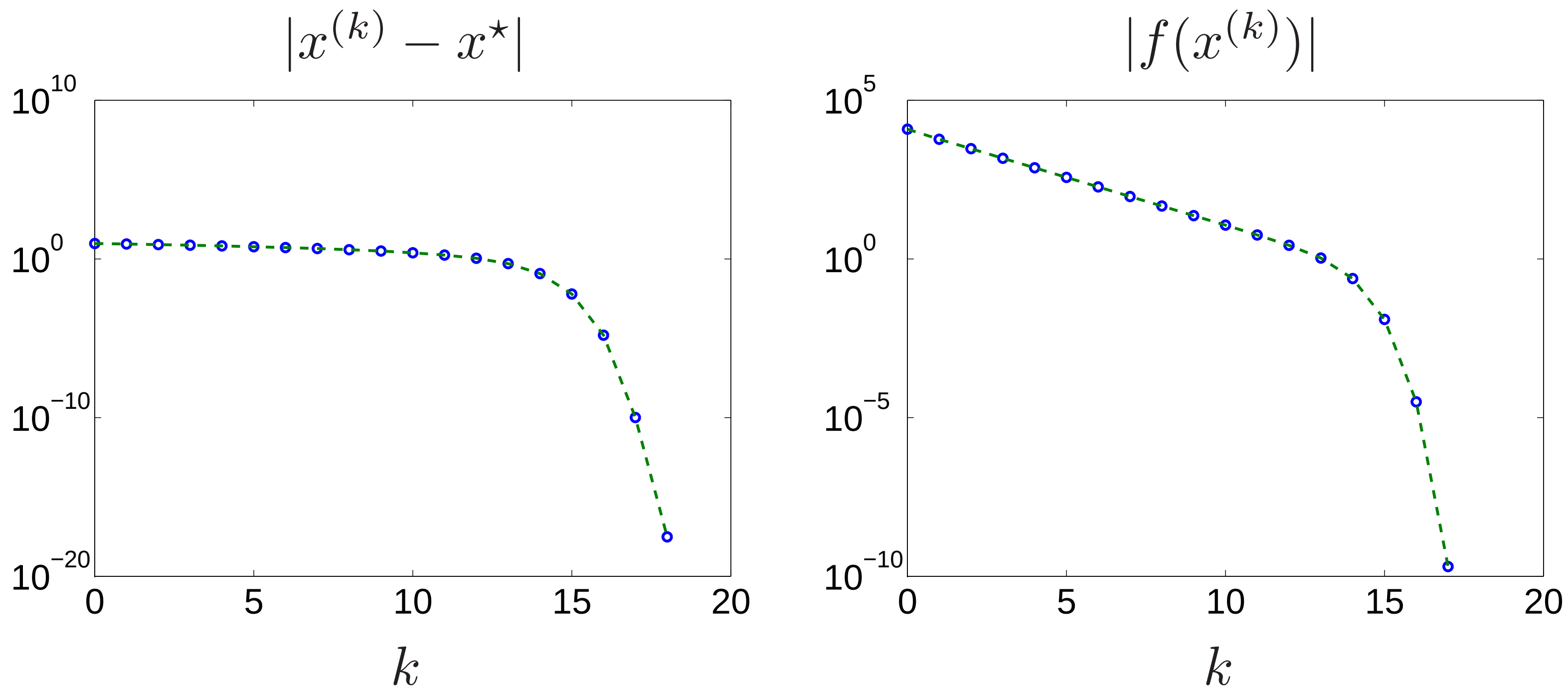
$$f_{\text{aff}}(y) = f(x) + g(y - x) \quad \text{with } g = \frac{f(x) - f(x_{\text{prev}})}{x - x_{\text{prev}}}$$

- solve linear equation $f_{\text{aff}}(y) = 0$ and take the solution as new iterate x^+:

$$x^+ = x - f(x)/g$$

Examples

- $f(x) = e^x - e^{-x}$, start at $x^{(0)} = 10$, $x^{(0)}_{\text{prev}} = 11$



fast asymptotic convergence, but slower than Newton method

- other examples: secant method works well if we start near a solution; may not converge otherwise

Error Analysis of Secant Method

define $e^{(k)} = x^{(k)} - x^\star$

- from the definition of the secant method and with some algebra:

$$e^{(k+1)} = x^{(k+1)} - x^\star = \frac{f(x^{(k)})e^{(k-1)} - f(x^{(k-1)})e^{(k)}}{f(x^{(k)}) - f(x^{(k-1)})}$$

- factoring out $e^{(k)}e^{(k-1)}$ and inserting $(x^{(k)} - x^{(k-1)})/(x^{(k)} - x^{(k-1)})$

$$e^{(k+1)} = \left(\frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}\right)\left(\frac{\frac{f(x^{(k)})}{e^{(k)}} - \frac{f(x^{(k-1)})}{e^{(k-1)}}}{x^{(k)} - x^{(k-1)}}\right)e^{(k)}e^{(k-1)}$$

(error equation)

- by Taylor's Theorem

$$f(x^{(k)}) = f(x^\star + e^{(k)}) = f(x^\star) + e^{(k)}f'(x^\star) + \frac{1}{2}(e^{(k)})^2 f''(x^\star) + \mathcal{O}((e^{(k)})^3)$$

- use $f(x^\star) = 0$ and divide both sides by $e^{(k)}$

$$f(x^{(k)})/e^{(k)} = f'(x^\star) + \frac{1}{2}e^{(k)}f''(x^\star) + \mathcal{O}((e^{(k)})^2)$$

- changing the index to $k-1$

$$f(x^{(k-1)})/e^{(k-1)} = f'(x^\star) + \frac{1}{2}e^{(k-1)}f''(x^\star) + \mathcal{O}((e^{(k-1)})^2)$$

- subtract the above two equations and neglect the higher order terms

$$f(x^{(k)})/e^{(k)} - f(x^{(k-1)})/e^{(k-1)} \approx \frac{1}{2}\left(e^{(k)} - e^{(k-1)}\right) f''(x^\star)$$

- since $x^{(k)} - x^{(k-1)} = e^{(k)} - e^{(k-1)}$

$$\frac{f(x^{(k)})/e^{(k)} - f(x^{(k-1)})/e^{(k-1)}}{x^{(k)} - x^{(k-1)}} \approx \frac{1}{2}f''(x^\star)$$

- use the above result and

$$\frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} \approx \frac{1}{f'(x^\star)}$$

in the error equation, we obtain

$$e^{(k+1)} \approx \frac{1}{2}\frac{f''(x^\star)}{f'(x^\star)}e^{(k)}e^{(k-1)} = ce^{(k)}e^{(k-1)}$$

- assume the method has α-order convergence, *i.e.*,

$$|e^{(k+1)}| \sim A|e^{(k)}|^\alpha$$

hence, we have

$$|e^{(k)}| \sim A|e^{(k-1)}|^\alpha, \qquad |e^{(k-1)}| \sim (A^{-1}|e^{(k)}|)^{1/\alpha}$$

- substitute the above result to get the asymptotic values of $e^{(k)}$:

$$A^{1+1/\alpha}|c|^{-1} \sim |e^{(k)}|^{1-\alpha+1/\alpha}$$

- the LHS is a nonzero constant while $k \to \infty$, so the exponent of $e^{(k)}$ must be zero

$$1 - \alpha + 1/\alpha = 0 \quad \Longrightarrow \quad \alpha = (1+\sqrt{5})/2 \approx 1.62$$

- the convergence rate of secant method is *super linear*

Convergence of Newton and secant methods

Newton method: if $f'(x^\star) \neq 0$ and $x^{(0)}$ is sufficiently close to $x^\star$, then Newton's method converges and there exists a $c > 0$ such that

$$|x^{(k+1)} - x^\star| \leq c\, |x^{(k)} - x^\star|^2$$

$i.e.$, quadratic convergence

secant method: if $f'(x^\star) \neq 0$ and $x^{(0)}$ is sufficiently close to $x^\star$, then the secant method converges and there exists a $c > 0$ such that

$$|x^{(k+1)} - x^\star| \leq c\,|x^{(k)} - x^\star|^r$$

where $r = (1 + \sqrt{5})/2 \approx 1.6$

$i.e.$, superlinear convergence

Summary

bisection method

- does not require derivatives
- user must provide initial interval $[l, u]$ with $f(l)f(u) < 0$
- R-linear convergence

Newton's method

- requires derivatives
- user must provide starting point near a solution
- quadratic convergence

secant method

- does not require derivatives
- user must provide two starting points near a solution
- superlinear convergence

Fixed Point Iteration

Idea: consider Newton's method as applied to $f(x) = x^2 - a$

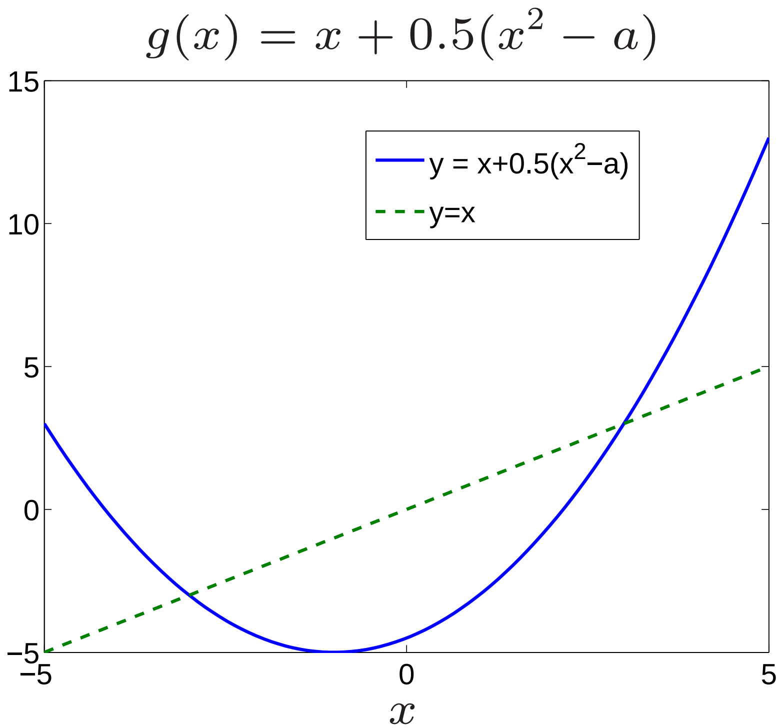
$$x^{(k+1)} = \frac{1}{2}\left(x^{(k)} + \frac{a}{x^{(k)}}\right)$$

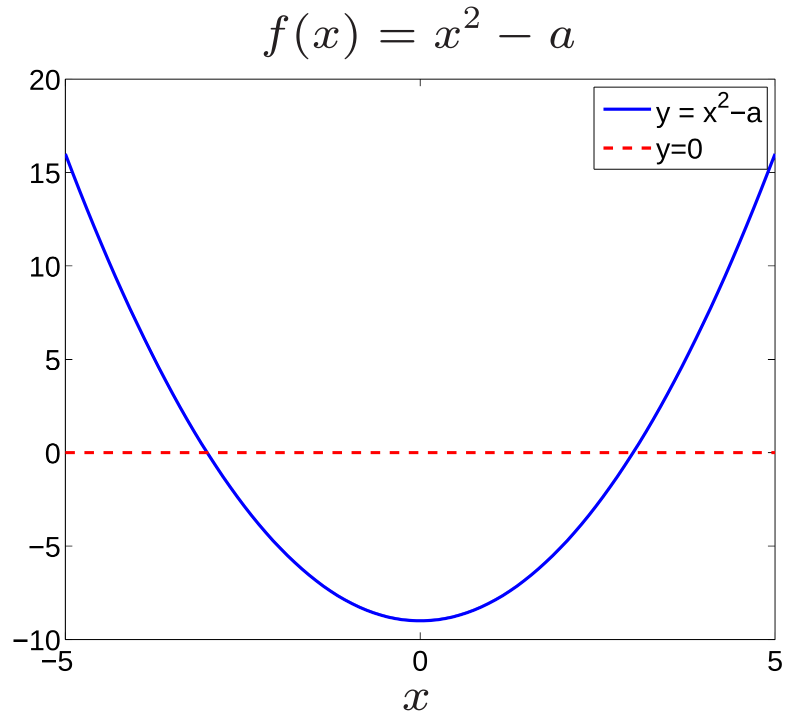
as $k \to \infty$, we know that $x^{(k)} \to \sqrt{a}$

write this more abstractly as

$$x^{(k+1)} = g(x^{(k)}) \quad \text{for} \quad g(x) = \frac{1}{2}(x + ax^{-1})$$

- $f(x^\star) = 0 \Longleftrightarrow x^\star = g(x^\star)$
- $x^\star$ is a **fixed point** of the function g
- functions $g = a/x$ or $g = a + x - x^2$ yield the same fixed point





- the root is where the curve crosses the x axis
- the fixed point is where the curve crosses the line $y = x$

Functional Iteration

a sequence of points computed by a formula of the form

$$x_{n+1} = g(x_n)$$

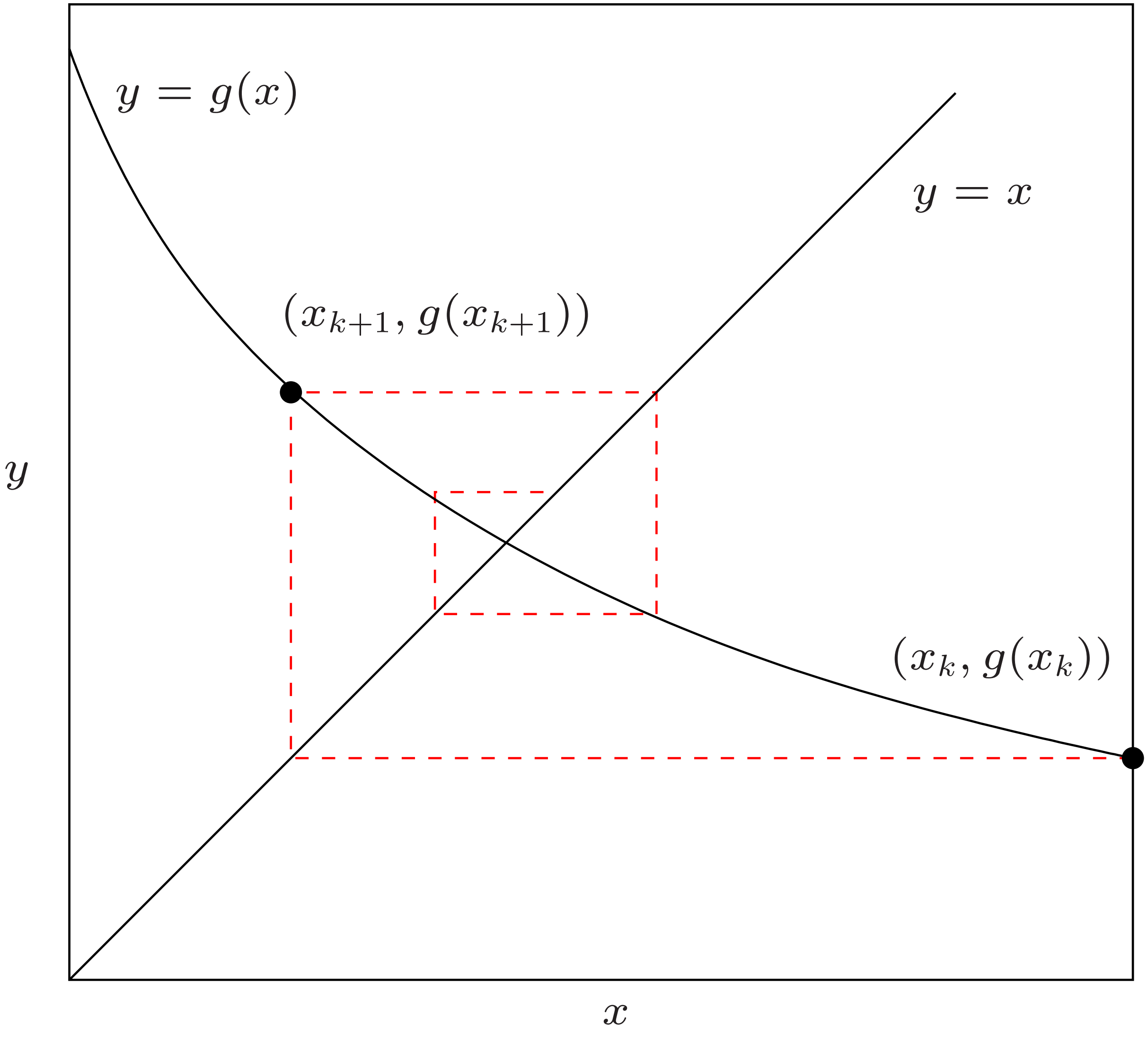
is called **functional iteration**

example: iterate these functions with $a = 9$ and $x^{(0)} = 1$

$$g_1 = x + \frac{1}{2}(x^2 - a), \quad g_2 = a/x, \quad g_3 = a + x - x^2, \quad g_4 = 0.5(x + a/x)$$

k	$x(k) + 0.5(x(k)^2 - a)$	$a/x(k)$	$a + x(k) - x(k)^2$	$0.5(x(k) + a/x(k))$
0	1.0000	1.0000	1.0000	1.0000
1	−3.0000	9.0000	9.0000	5.0000
2	−3.0000	1.0000	$-6.3000e+01$	3.4000
3	−3.0000	9.0000	$-4.0230e+03$	3.0235
4	−3.0000	1.0000	$-1.6189e+07$	3.0001
5	−3.0000	9.0000	$-2.6207e+14$	3.0000

Graphical interpretation



suppose a sequence x_n converges with $\lim_{n\to\infty} x_n = z$

if g is *continuous* then

$$g(z) = g(\lim_{n\to\infty} x_n) = \lim_{n\to\infty} g(x_n) = \lim_{n\to\infty} x_{n+1} = z$$

we call z a **fixed point** of the function g

one may ask the following questions:

- existence and uniqueness of a fixed point
- if the iteration converges, how fast does it converge ?

Contractive Mapping

a mapping (or function) g is said to be *contractive* on $S \subseteq \mathbf{dom}\, g$ if

there exists a scalar $\lambda < 1$ such that

$$|g(x) - g(y)| \leq \lambda |x - y|$$

for all $x, y \in S$

- loosely speaking, a contractive function is a *non-expansive* map
- every contractive mapping is Lipschitz continuous
- if g is continuously differentiable on $[a, b]$ with

$$\max_{x \in [a,b]} |g'(x)| < 1$$

  then g is contractive on $[a, b]$ (by Mean-Value Theorem)

examples: $g(x) = e^{-x}, \cos x$ on $[0, 1]$

Contractive Mapping Theorem

if $g : S \to S$ is contractive for all $x \in S$, then

- g has a *uniqued* fixed point in S
- this fixed point is a limit of every sequence

$$x_{n+1} = g(x_n) \quad \text{with} \quad x_0 \in S$$

Fixed Point Existence and Convergence

let $g \in \mathbf{C}([a, b])$ with $a \leq g(x) \leq b$ for all $x \in [a, b]$

1. g has at least one fixed point $x \in [a, b]$

2. if g is contractive on $[a, b]$ then

   (a) $x^\star$ (root of $f(x) = 0$) is unique

   (b) the iteration
   $$x^{(n+1)} = g(x^{(n)})$$
   converges to $x^\star$ for any initial guess $x^{(0)} \in [a, b]$

   (c) the error estimate obeys
   $$|x^\star - x^{(k)}| \leq \frac{\lambda^k}{1 - \lambda}|x^{(1)} - x^{(0)}|$$
   R-linear convergence with $c = \lambda$ and $M = |x^{(1)} - x^{(0)}|/(1 - \lambda)$

- Proof 1: define $h(x) = g(x) - x$ and use the intermediate value theorem
- Proof 2a-2b: a direct result from contractive mapping theorem
- Proof 2c: since g is contractive

$$|x^\star - x^{(k)}| = |g(x^\star) - g(x^{(k-1)})| \leq \lambda |x^\star - x^{(k-1)}|$$

write this recursively, we obtain

$$|x^\star - x^{(k)}| \leq \lambda^k |x^\star - x^{(0)}|$$

and apply the following result:

$$\begin{aligned} |x^\star - x^{(0)}| = |x^\star - g(x^{(0)}) + x^{(1)} - x^{(0)}| &\leq |g(x^\star) - g(x^{(0)})| + |x^{(1)} - x^{(0)}| \\ &\leq \lambda |x^\star - x^{(0)}| + |x^{(1)} - x^{(0)}| \end{aligned}$$

from which it follows that

$$|x^\star - x^{(0)}| \leq \frac{1}{1-\lambda} |x^{(1)} - x^{(0)}|$$

example 1 apply the theorem on $g_4(x) = 0.5(x + a/x)$ with $a = 9$

- $2 \leq g_4(x) \leq 4$ for all $x \in [2, 4]$
- $g_4'(x) = 1/2 - a/2x^2$ and $|g_4'(x)| < 1$ on $[2, 4]$
- hence $g_4(x)$ is contractive on $[2, 4]$
- there's a fixed point in $[2, 4]$ and the iteration converges

example 2 apply the theorem on $g_3(x) = a + x - x^2$ with $a = 9$

- $|g_3'(x)| < 1$ on $(0, 1)$ (says g_3 is contractive on $[\frac{1}{4}\ ,\ \frac{1}{2}]$)
- but $g_3(x)$ does not satisfy $1/4 \leq g_3(x) \leq 1/2$
- no fixed poin in $[1/4, 1/2]$ (in fact the $\sqrt{a} = 3$ is not in this interval)

Local Convergence for Fixed Point Iteration

assumptions:

- $x^\star$ is a fixed point
- g be a continuously differentiable in an open interval of $x^\star$
- $|g'(x^\star)| < 1$

then for all x_0 sufficiently close to $x^\star$, the iteration

$$x_{n+1} = g(x_n)$$

converges,

$$\lim_{n\to} \frac{x^\star - x_{n+1}}{x^\star - x_n} = g'(x^\star)$$

and

$$|x^\star - x_n| \leq \frac{\lambda^n}{1-\lambda}|x_1 - x_0|$$

for some $\lambda < 1$

Proof.

- since g is continously differentiable, we can find an closed interval J centered at $x^\star$ such that

$$|g'(x)| \leq \lambda < 1, \qquad \forall x \in J$$

- from the definition of fixed point iteration

$$|x^\star - x_1| = |g(x^\star) - g(x_0)| \leq \lambda |x^\star - x_0|$$

- x_1 is closer to $x^\star$ than x_0, and so are all the rest of the iterates
- thus, $g(x) \in J$ for all $x \in J$
- the rest of the proof follows similarly to the result in page 2-53

Higher order rate of convergence

assumptions:

- g is p times continuously differentiable
- $x^\star$ is a fixed point, *i.e.*, $x^\star = g(x^\star)$

if

$$g'(x^\star) = g''(x^\star) = \cdots = g^{(p-1)}(x^\star) = 0$$

but $g^{(p)} \neq 0$ then the iteration

$$x_{n+1} = g(x_n)$$

converges with order p for x_0 sufficiently close to $x^\star$

Proof.

- since $g'(x^\star) = 0 < 1$, the iteration converges for x_0 close to $x^\star$
- by Taylor's Theorem,

$$g(x_n) = g(x^\star) + (x_n - x^\star)g'(x^\star) + \cdots + \frac{(x_n - x^\star)^{p-1}}{(p-1)!}g^{(p-1)}(x^\star) + \frac{(x_n - x^\star)^p}{p!}g^{(p)}(\xi_n)$$

where ξ_n is between x_n and $x^\star$

- since all derivative terms are zero except that in the remainder term,

$$g(x_n) - g(x^\star) = \frac{(x_n - x^\star)^p}{p!}g^{(p)}(\xi_n)$$

- so that the iteration them implies

$$\frac{(x_{n+1} - x^\star)}{(x_n - x^\star)^p} = \frac{1}{p!} g^{(p)}(\xi_n)$$

from which the convergence with order p follows

Newton's method: equivalent to a fixed point iteration with

$$g(x) = x - \frac{f(x)}{f'(x)}$$

note that

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

with $f(x^\star) = 0$ we can see that $g'(x^\star) =$ 1 -1 $=$ 0

Newton's method has (local) order of convergence of at least 2

Computing Roots of Polynomials

consider the polynomial p

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

if $a_n \neq 0$, p is of *degree* n

Factor Theorem: a polynomial of degree n can be written as a product of n linear factors

$$p(z) = (z - r_1)(z - r_2) \cdots (z - r_n) q_n$$

- r_k for $k = 1, \ldots, n$ corresponds to a root of $p(z)$
- q_n is a constant
- **multiplicity** of r_k is the number of factor $z - r_k$ in $p(z)$

questions:

- how can we find the roots of p ?
- is there any root at all ?

Theorem 1. *Every nonconstant polynomial has at least one root in the complex field.*

(This says nothing about the existence of real roots!)

Theorem 2. *A polynomial of degree n has exactly n roots in the complex plane; each root is counted to its multiplicity.*

is it possible to restrict these roots to a limited area?

Localization Theorem

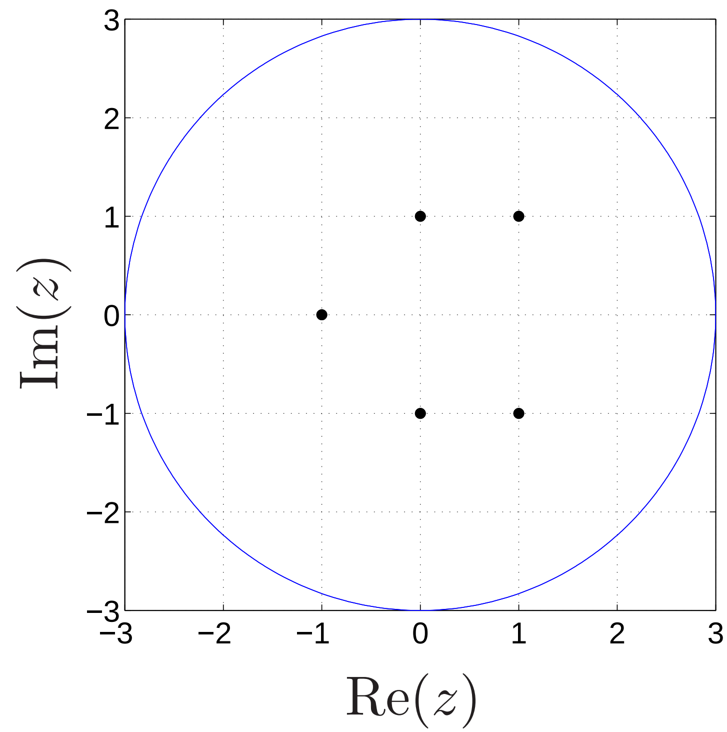
Theorem 3. *All roots of the polynomial* p *lie in the open disk whose center is at the origin of the complex plane and whose radius is*

$$\rho = 1 + \frac{1}{|a_n|} \max_{0 \le k < n} |a_k|$$

- the radius is always greater than 1, $i.e.$, $\rho > 1$
- the radius ρ is an upper bound of the moduli of all roots

$$|r_k| \le \rho, \qquad k = 1, \ldots, n$$

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



the radius of the disk from the localization theorem is

$$\rho = 1 + \frac{1}{|a_5|} \max_{0 \le k < 5} |a_k| = 1 + 2/1 = 3$$

all the roots of p are: $1 \pm i, -1, \pm i$

we can get a more specific region of all roots of $p(z)$

let $q(z)$ be a polynomial with the order reverse to $p(z)$

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

in another word, $q(z) = z^n p(1/z)$

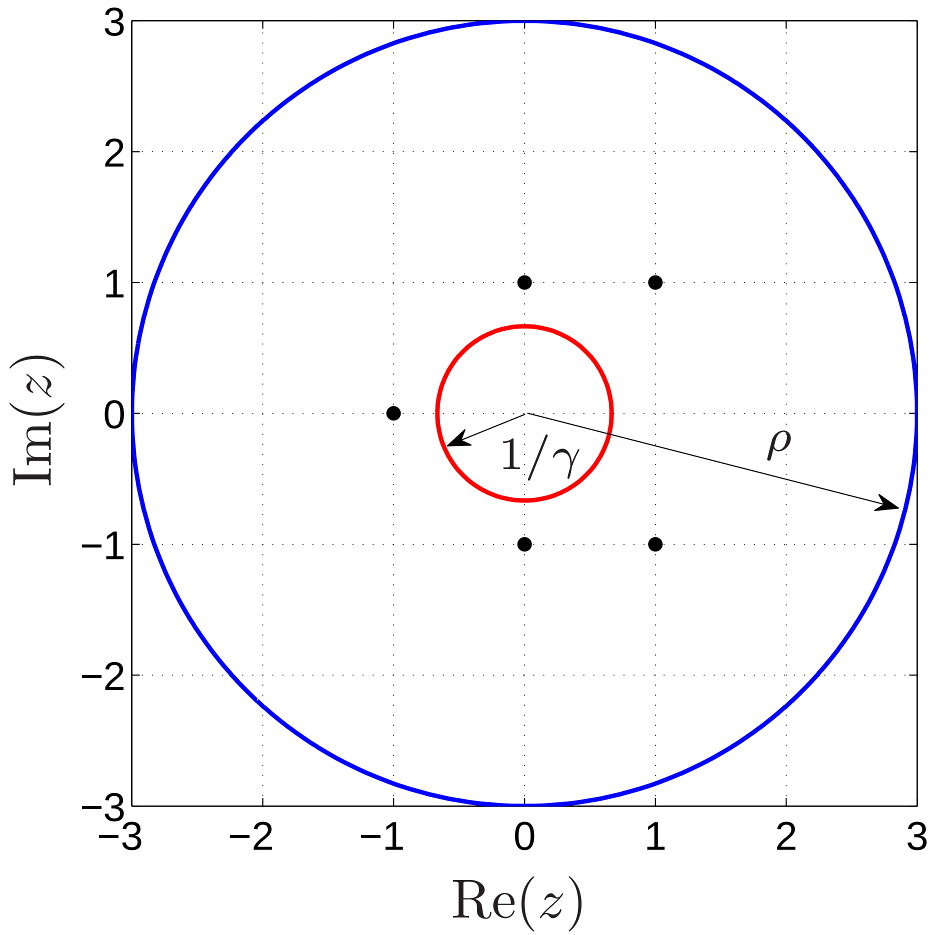
therefore, if r is a root of p, then $1/r$ is a root of q

$$p(r) = 0 \quad \Longleftrightarrow \quad q(1/r) = 0$$

this fact yields the following theorem

Theorem 4. *if all the roots of q are in the disk $\{z \mid |z| \leq \gamma\}$, then all the nonzero roots of p are outside the disk $\{z \mid |z| < 1/\gamma\}$.*

example: $p(z) = z^5 - z^4 + z^3 + z^2 + 2$, $\rho = 3$



$q(z) = 2z^5 + z^3 + z^2 - z + 1, \qquad \gamma = 1 + \frac{1}{2}\max\{1, 1, -1, 1\} = 1 + 1/2 = 3/2$

all roots of $p(z)$ lie in the ring $2/3 \leq |z| \leq 3$

Evaluating a Polynomial

to evaluate a polynomial

$$p(z) = a_0 + a_1 z + \cdots + a_n z^n$$

it requires n multiplications, n additions, and forming $z^2, z^3, \ldots$

a more effficient way to do is to write $p(z)$ as

$$p(z) = a_0 + z(a_1 + z(a_2 + \cdots + z(a_{n-1} + a_n z) \cdots))$$

- this is called **nested multiplication**
- requires only n multiplication and n additions
- an algorithm form of nested multiplication is known as **Horner's rule**

Horner's Rule

$$p(z) = a_0 + z(a_1 + z(a_2 + \cdots + z(a_{n-1} + a_n z) \cdots))$$

---

given a point z, and polynomial coefficients a_j, $j = 0, \ldots, n$
set $p = a_n$
for $k = n - 1$ **downto** 0

$$p = a_k + pz$$

endfor
return p

Horner's Rule for Evaluating a Derivative

Horner's rule can be modified to compute $p'(z)$ as well

$$p'(z) = a_1 + 2a_2 z + 3a_3 z^3 + \cdots + n a_n z^{n-1}$$

which can be written in a nested multiplication as

$$p'(z) = a_1 + z(2a_2 + z(3a_3 + \cdots + z((n-1)a_{n-1} + n a_n z)\cdots))$$

---

given a point z and polynomial coefficients a_j, $j = 0, \ldots, n$
set $d = n a_n$
for $k = n-1$ **downto** 1
    $d = k a_k + dz$

endfor
return d

More Efficient Horner's Algorithm

Idea: store intermediate values for computing derivatives

to evaluate $p(z)$ at z_0, consider the polynomial

$$q(z) = \frac{p(z) - p(z_0)}{z - z_0}$$

where

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

let

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

comparing coefficients yields

$$\begin{aligned} p(z) &= (z - z_0) q(z) + p(z_0) \\ &= \underbrace{p(z_0) - z_0 b_0}_{a_0} + \underbrace{(b_0 - z_0 b_1)}_{a_1} z + \ldots + \underbrace{(b_{n-2} - z_0 b_{n-1})}_{a_{n-1}} z^{n-1} + \underbrace{b_{n-1}}_{a_n} z^n \end{aligned}$$

the two sets of coefficients are related by

$$\begin{aligned}
b_{n-1} &= a_n \\
b_{n-2} &= a_{n-1} + z_0 b_{n-1} \\
&\vdots \\
b_0 &= a_1 + z_0 b_1 \\
p(z_0) &= a_0 + z_0 b_0
\end{aligned}$$

nested computation: $p(z_0) = a_0 + z_0(a_1 + z_0(a_2 + \cdots (a_{n-1} + z_0 a_n) \cdots))$

---

given a point z, and polynomial coefficients a_j, $j = 0, \ldots, n$
set $b_{n-1} = a_n$
for $k = n-1$ **downto** 0
    $b_{k-1} = a_k + z b_k$
endfor
return b_i for $i = -1, 0, \ldots, n-1$ (where $b_{-1} = a_0 + z b_0 = p(z)$)

Horner's Algorithm in a Table

to calculate by hand, we can form the coefficients in a table as

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_0
z_0		$z_0 b_{n-1}$	$z_0 b_{n-2}$	$\cdots$	$z_0 b_0$
	b_{n-1}	b_{n-2}	b_{n-3}	$\cdots$	$\boxed{b_{-1}}$

example: evaluate $p(3)$ where

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

	1	-4	7	-5	-2
3		3	-3	12	21
	1	-1	4	7	$\boxed{19}$

hence, $p(3)$ is equal to 19

Deflation

a process of removing a linear factor from $p(z)$ is called **deflation**

- if z_0 is a root of $p(z)$ then $p(z) = (z - z_0)q(z)$
- removing factor $z - z_0$ gives $q(z)$ which can be determined by b_j's

example: deflate $p(z)$ in page 2-72 given that 2 is a root of $p(z)$

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

thus, we have

$$p(z) = z^4 - 4z^3 + 7z^2 - 5z - 2 = (z-2)(z^3 - 2z^2 + 3z + 1)$$

Evaluating the Derivatives

suppose that $p(z)$ can be written in the form

$$p(z) = c_n(z - z_0)^n + c_{n-1}(z - z_0)^{n-1} + \cdots + c_0$$

and hence $q(z)$ is of the form,

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

- Taylor's Theorem says that $c_k = p^{(k)}(z_0)/k!$
- $c_0 = p(z_0)$ and obtained by applying Horner's algorithm to p at z_0
- $c_1 = p'(z_0)$ and obtained by applying Horner's algorithm to q at z_0
- repeat Horner's algorithm until we can get all coefficients c_k's

example: expand the Taylor series around the point 3 of the polynomial

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

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

hence, the Taylor series is

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

Complete Horner's Algorithm

the algorithm in page 2-74 can be implemented in pseudocode as follows

---

given a point z, and polynomial coefficients a_i, $i = 0, \ldots, n$
for $k = 0$ **upto** $n - 1$
    for $j = n - 1$ **downto** k
        $a_j = a_j + z a_{j+1}$
    endfor
endfor
return a_i for $0 \leq i \leq n$

---

where the coefficients c_k overwrite the input coefficients a_k

Last note: Horner's algorithm can be used to compute $f(x)$ and $f'(x)$ in Newton's method when applied to polynomial functions

Bairstow's Method

Theorem 5. *If all coefficients of $p(z)$ are real, and if w is a nonreal root of $p(z)$, then so is $\bar{w}$. In addition, $(z-w)(z-\bar{w})$ is a real quadratic factor of $p(z)$.*

basically, a complex root of a real polynomial must occur in complex conjugate pair

Idea:

- Newton's method requires complex arithmetic to find a complex root
- using Bairstow's method, we can employs only *real* arithmetic
- the method uses Newton's iteration to search for quadratic factors

Quotient and Remainder

if a *real* polynomial $p(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0$

is divided by the *quadratic* polynomial

$$d(z) = z^2 - uz - v$$

then the quotient and remainder

$$\begin{aligned} q(z) &= b_n z^{n-2} + b_{n-1} z^{n-3} + \cdots + b_3 z + b_2 \\ r(z) &= b_1 (z - u) + b_0 \end{aligned}$$

such that $p(z) = q(z)d(z) + r(z)$ can be computed recursively by setting

$$b_{n+1} = b_{n+2} = 0$$

and using

$$b_k = a_k + u b_{k+1} + v b_{k+2} \qquad n \geq k \geq 0$$

Bairstow's method uses Newton's iteration to find u, v such that

$$\begin{aligned} b_0(u,v) &= 0 \\ b_1(u,v) &= 0 \end{aligned}$$

- d (the quadratic polynomial) will then become a factor of p
- once u, v are found, the roots of d are readily obtained
- the Newton's iteration for solving $b_0(u,v) = 0$ and $b_1(u,v) = 0$ is

$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ v_k \end{bmatrix} - \begin{bmatrix} \dfrac{\partial b_1(u_k,v_k)}{\partial u} & \dfrac{\partial b_1(u_k,v_k)}{\partial v} \\ \dfrac{\partial b_0(u_k,v_k)}{\partial u} & \dfrac{\partial b_0(u_k,v_k)}{\partial v} \end{bmatrix}^{-1} \begin{bmatrix} b_1(u_k,v_k) \\ b_0(u_k,v_k) \end{bmatrix}$$

- the partial derivatives and $b_1(u_k,v_k)$, $b_0(u_k,v_k)$ are computed from the recurrence equation in page 2-78

Laguerre Iteration

$p(z)$: a real polynomial

---

given initial z, required tolerance $\epsilon > 0$
repeat

1. Compute $A = \frac{p'(z)}{p(z)}$.
2. Compute $B = A^2 - \frac{p''(z)}{p(z)}$.
3. Compute $C = (A \pm \sqrt{(n-1)(nB - A^2)})/n$.
   (the sign is chosen so that $|C|$ is largest)
4. **if** $|1/C| \leq \epsilon$, **return** z.
5. $z := z - 1/C$.

until maximum number of iterations is exceeded.

- each iteration requires one evaluation of p, p' and p''
- evaluating p, p' and p'' at z can be done by Horner's algorithm

interpretation: define

- r_i, $i = 1, 2, \ldots, n$ the n real roots of $p(z)$
- $1/v_i$ the distance from a point z (on real axis) to the ith root

$$v_i = (z - r_i)^{-1}$$

by writing $p(z)$ as a product of linear factors

$$p(z) = (z - r_1)(z - r_2) \cdots (z - r_n)$$

and consider

$$\ln |p(z)| = \ln |z - r_1| + \cdots + \ln |z - r_n|$$

we can verify that

$$\begin{aligned} A &= \tfrac{d}{dz} \ln |p(z)| &= \tfrac{p'(z)}{p(z)} &= \textstyle\sum_{i=1}^n \tfrac{1}{z - r_i} &= \textstyle\sum_{i=1}^n v_i \\ B &= -\tfrac{d^2}{dz^2} \ln |p(z)| &= \left(\tfrac{p'(z)}{p(z)}\right)^2 - \tfrac{p''(z)}{p(z)} &= \textstyle\sum_{i=1}^n \tfrac{1}{(z - r_i)^2} &= \textstyle\sum_{i=1}^n v_i^2 \end{aligned}$$

Fact: for any *real* $z \neq r_j$ for all j, the numbers $(z - r_j)^{-1}$ lies in the interval whose endpoints are

$$(A \pm \sqrt{(n-1)(nB - A^2)})/n$$

hence, C^{-1} is an estimate of the distance from z to the nearest root

Proof. in fact, for *any real numbers* v_i with

$$A = \sum_{i=1}^{n} v_i, \qquad B = \sum_{i=1}^{n} v_i^2$$

we can consider

$$A^2 - 2Av_1 + v_1^2 = (A - v_1)^2 = (v_2 + v_3 + \cdots + v_n)^2$$

by Cauchy Schwarz inequality

$$A^2 - 2Av_1 + v_1^2 \leq (n-1)(v_2^2 + v_3^2 + \cdots + v_n^2) = (n-1)(B - v_1^2)$$

rewrite the inequality as a quadratic polynomial in v_1

$$nv_1^2 - 2Av_1 + A^2 - (n-1)B \leq 0$$

define

$$q(x) = nx^2 - 2Ax + A^2 - (n-1)B$$

so we have $q(v_1) \leq 0$

for large $|x|$, evidently $q(x) > 0$, so v_1 must lie between two roots of q:

$$(A \pm \sqrt{(n-1)(nB - A^2)})/n$$

which are the endpoints given in 2-82

- $nB - A^2 \geq 0$ (by Cauchy Schwarz inequality), so the endpoints are real
- the result also holds for v_i, $i = 2, \ldots, n$
- use $v_i = (z - r_i)^{-1}$ and we finish the proof

example: $p(z) = z^4 - 8z^3 - 25z^2 + 44z + 60$

$p(z)$ has all simple roots $r = -1, 2, -3$ and 10

k	z	z	z	z
0	-20.000000	100.000000	4.000000	-2.000000
1	-4.369910	10.416379	2.272328	-1.242866
2	-3.041839	10.000039	2.001053	-1.002888
3	-3.000003	10.000000	2.000000	-1.000000
4	-3.000000	10.000000	2.000000	-1.000000

remarks:

- used in several softwares packages
- for polynomials with all real roots, converges from any starting point
- third-order convergence near simple roots
- if p has multiple roots, the convergence rate is linear

example: $p(z) = z^3 - 4z^2 + 6z - 4$

$p(z)$ has *complex* roots $r = 2$ and $1 \pm i$

k	z	z	z
0	1000000.000000	$100.000000 - 2000.000000i$	5.000000
1	$1.333334 + 0.943020i$	$1.332561 - 0.942549i$	$1.285968 + 0.256216i$
2	$1.003279 + 1.000001i$	$1.003260 - 0.999979i$	$1.833103 - 0.298087i$
3	$1.000000 + 1.000000i$	$1.000000 - 1.000000i$	$1.989546 - 0.006191i$
4	$1.000000 + 1.000000i$	$1.000000 - 1.000000i$	$2.000000 + 0.000000i$

remarks:

- when a starting point is complex or p has complex zeros, the iteration is performed in complex arithmetic
- A, B and C are now complex numbers
- fast convergence though a starting point is quite far from the root

Homotopy and Continuation Methods

- let $f : X \to Y$ be a function and we find the roots of an equation

$$f(x) = 0$$

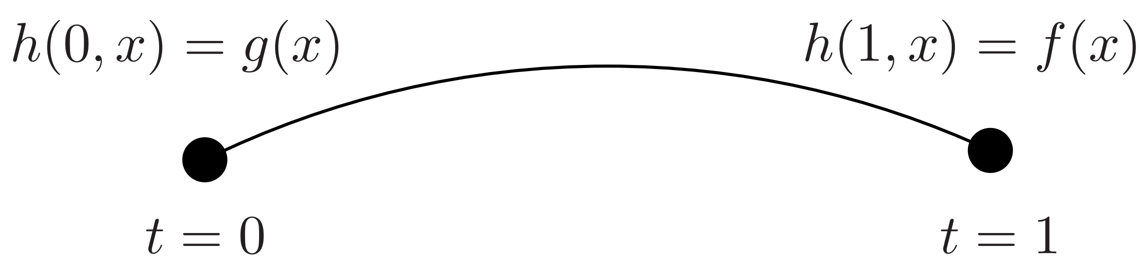
- for a function g, we define a homotopy $h : [0,1] \times X \to Y$ as

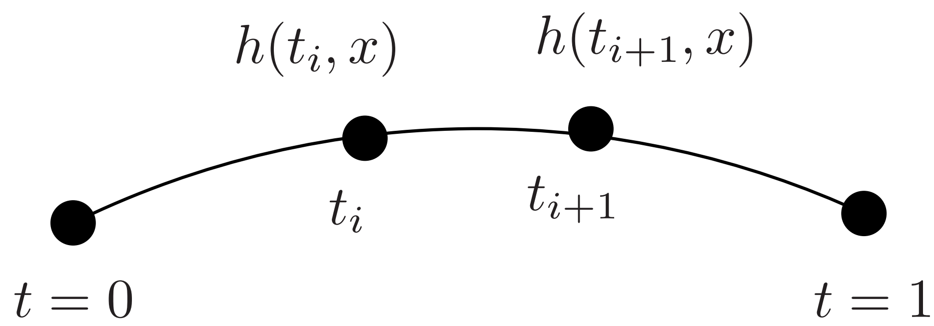
$$h(t,x) = tf(x) + (1-t)g(x),$$

where t runs over the interval $[0,1]$

if $t = 0$, then $h(0,x) = g(x) = 0 \to$ **easy problem**

If $t = 1$, then $h(1,x) = f(x) = 0 \to$ **original problem**





if $h(t, x) = 0$ has a unique root for each $t \in [0, 1]$

then the root is a function of t, denoted by

$$\{x(t) : 0 \le t \le 1\}$$

example: let $g(x) = f(x) - f(x_0)$

$$\begin{aligned} h(t, x) &= tf(x) + (1 - t)\left[f(x) - f(x_0)\right], \\ &= f(x) + (t - 1)f(x_0) \end{aligned}$$

$x(0)$ will be the solution of the problem when $t = 0$ (simple problem)

$x(1)$ is the solution to the original problem ($f(x) = 0$)

continuation method: determine the curve $x(t)$ by computing points

$$x(t_0), x(t_1), \ldots, x(t_m)$$

on the curve

if the function $t \mapsto x(t)$ and the function h are differentiable

differentiating $0 = h(t, x(t))$ respect to t gives

$$0 = h_t(t, x(t)) + h_x(t, x(t))x'(t)$$

the differential equation describing the path:

$$x'(t) = -\left[h_x(t, x(t))\right]^{-1} h_t(t, x(t)), \quad x(0) = x_0$$

(where $x(0)$ is supposedly known)

example: find the roots of system equation

$$0 = f(x) = \begin{bmatrix} \xi_1^2 - 3\xi_2^2 + 3 \\ \xi_1\xi_2 + 6 \end{bmatrix}, \quad x = (\xi_1, \xi_2) \in \mathbb{R}^2.$$

define a homotopy

$$h(t, x) = f(x) + (t - 1)f(x_0)$$

select $x_0 = (1, 1)$

$$h_x = f'(x) = \begin{bmatrix} \partial f_1/\partial \xi_1 & \partial f_1/\partial \xi_2 \\ \partial f_2/\partial \xi_1 & \partial f_2/\partial \xi_2 \end{bmatrix} = \begin{bmatrix} 2\xi_1 & -6\xi_2 \\ \xi_2 & \xi_1 \end{bmatrix}$$

$$h_t = f(x_0) = \begin{bmatrix} f_1(x_0) \\ f_2(x_0) \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \end{bmatrix}$$

the inverse of $f'(x)$ is

$$h_x^{-1} = [f'(x)]^{-1} = \frac{1}{2\xi_1^2 + 6\xi_2^2} \begin{bmatrix} \xi_1 & 6\xi_2 \\ -\xi_2 & 2\xi_1 \end{bmatrix}.$$

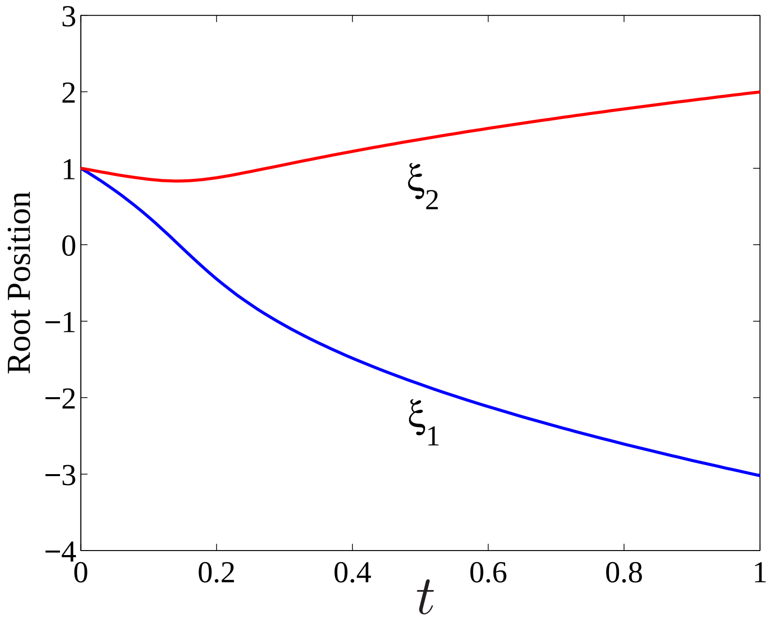
the differential equation describing the solution path is

$$\begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} = -h_x^{-1} h_t = -\frac{1}{2\xi_1^2 + 6\xi_2^2} \begin{bmatrix} \xi_1 & 42\xi_2 \\ -\xi_2 & 14\xi_1 \end{bmatrix}$$

use the Euler method, *i.e.*,

$$x(t+\delta) = x(t) + x'(t)\delta, \quad \delta = 0.01$$

to solve the differential equation numerically



obtain $x(1) = (-3.019, 1.997)$

use this as a starting point in Newton's method

(notice that f has a root $(-3, 2)$)

Continuously Differentiable Solution

Theorem 6. *If $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and if $\| [f'(x)]^{-1} \| \leq M$ on $\mathbb{R}^n$, then for any $x_0 \in \mathbb{R}^n$ there is a unique curve*

$$\{x(t) : 0 \leq t \leq 1\},$$

in $\mathbb{R}^n$ such that

$$f(x(t)) + (t-1)f(x_0) = 0,$$

with $0 \leq t \leq 1$

the function $t \mapsto x(t)$ is a **continuously differentiable solution** of the initial-value problem

$$x' = -[f'(x)]^{-1} f(x_0)$$

where $x(0) = x_0$

Solution of Homogeneous Equation

Lemma 7. *Let A be an $n \times (n+1)$ matrix. A solution of the homogeneous equation $Ax = 0$ is given by*

$$x_j = (-1)^{j+1} \det(A_j),$$

where A_j is A without column j.

Proof. augment the ith row of A with A; call this matrix as B

- B is obviously singular since it contains two equal rows
- expand the determinant of B by the elements in its top row:

$$0 = \det B = \sum_{j=1}^{n+1} (-1)^{j+1} a_{ij} \det(A_j) = \sum_{j=1}^{n+1} a_{ij} x_j$$

- this is true for $i = 1, 2, \dots, n$, we have $Ax = 0$. □

Tracing the Path

another way of tracing the path $x(t)$ given by Garcia and Zangwill (1981)

suppose that $x \in \mathbb{R}^n$ and $t \in [0, 1]$

a vector $y \in \mathbb{R}^{(n+1)}$ is defined by

$$y = (t, \xi_1, \xi_2, \ldots, \xi_n),$$

where $\xi_1, \xi_2, \ldots, \xi_n$ are the components of x

hence, our equation is

$$h(t, x) = 0 \quad \Longrightarrow \quad h(y) = 0 \quad \Longrightarrow \quad h(y(s)) = 0$$

where we allow y to be a function of an *independent variable* s

differentiate respect to s, we have

$$h'(y(s))y'(s) = 0$$

the vector $y(s)$ has $n+1$ components, with denote by $\eta_1, \eta_2, \ldots, \eta_{n+1}$

from the lemma, we get

$$\eta_j' = (-1)^{j+1} \det(A_j) \quad (1 \leq j \leq n+1),$$

where $A = h'(y(s))$ and A_j is A without column j

example: find the roots of system equation

$$0 = f(x) = \begin{bmatrix} \xi_1^2 - 3\xi_2^2 + 3 \\ \xi_1\xi_2 + 6 \end{bmatrix}, \quad x = (\xi_1, \xi_2) \in \mathbb{R}^2$$

consider the homotopy $h(t,x) = f(x) + (t-1)f(x_0)$ with $x_0 = (1,1)$

$$h(t,x) = \begin{bmatrix} \xi_1^2 - 3\xi_2^2 + 2 + t \\ \xi_1\xi_2 - 1 + 7t \end{bmatrix}$$

differentiate with respect to s

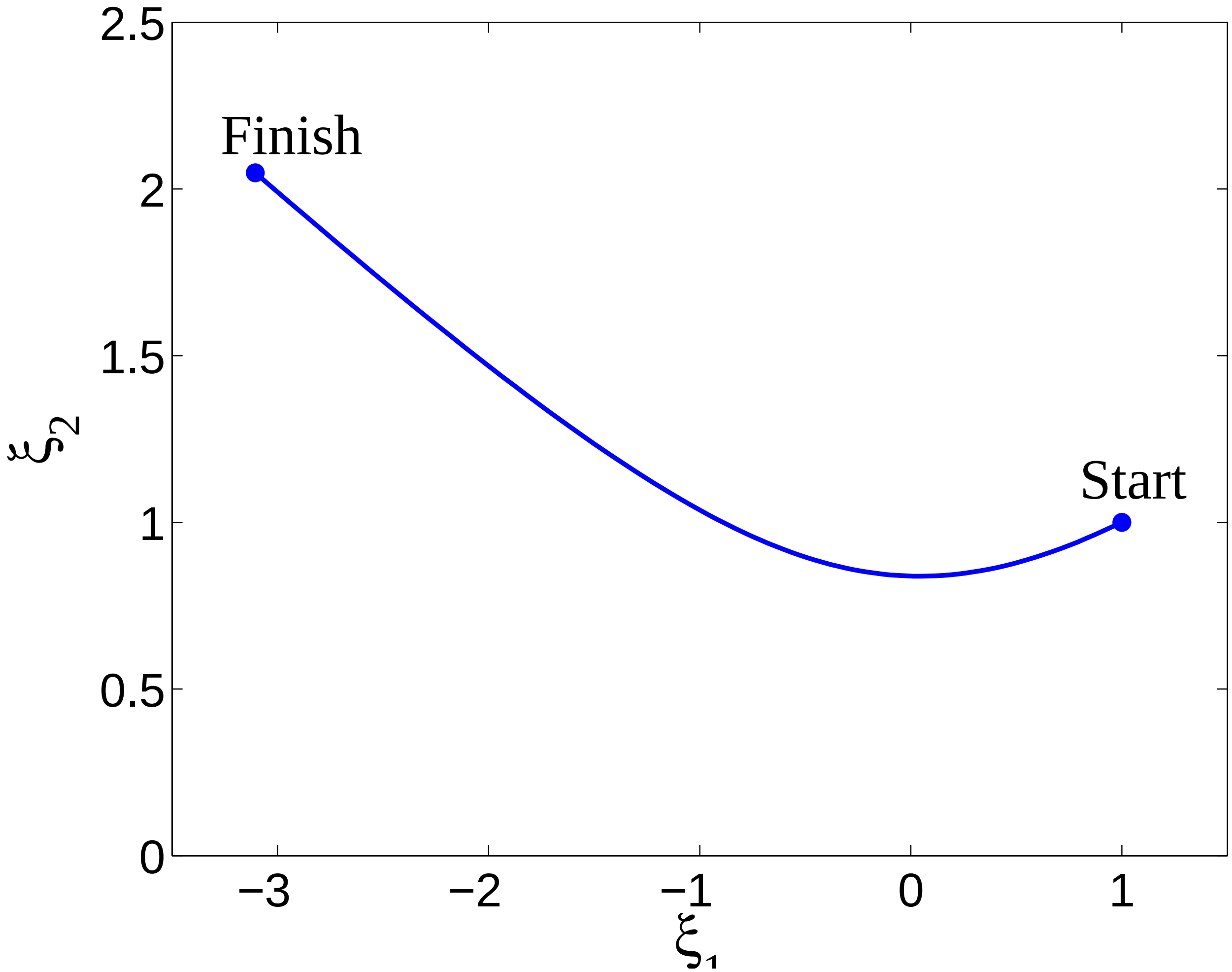
$$h'(y(s))y'(s) = \begin{bmatrix} 1 & 2\xi_1 & -6\xi_2 \\ 7 & \xi_2 & \xi_1 \end{bmatrix} \begin{bmatrix} t' \\ \xi_1' \\ \xi_2' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

solve the linear equation to obtain the differential equations

$$\left\{ \begin{array}{rcrrcl} t' & = & 2\xi_1^2 + 6\xi_2^2, & t(0) & = & 0, \\ \xi_1' & = & -\xi_1 - 42\xi_2, & \xi_1(0) & = & 1, \\ \xi_2' & = & \xi_2 - 14\xi_1, & \xi_2(0) & = & 1. \end{array} \right.$$

use the Euler method to solve the diff. equation ($\delta s = 0.001$), we obtain

$$\begin{aligned} (s, t, \xi_1, \xi_2) &= (0.089, 0.999, -3.024, 2.004), \\ (s, t, \xi_1, \xi_2) &= (0.090, 1.042, -3.105, 2.048). \end{aligned}$$



points near $t = 1$ can be used to start a Newton iteration

Relation to Newton's Method

consider the homotopy

$$h(t, x) = f(x) - e^{-t} f(x_0),$$

where t runs from 0 to ∞

we seek a curve or path, $x = x(t)$, on which

$$0 = h(t, x(t)) = f(x(t)) - e^{-t} f(x_0).$$

differentiating h w.r.p to t leads to a diff. eq. describing the path:

$$\begin{aligned} 0 &= f'(x(t))x'(t) + e^{-t} f(x_0) = f'(x(t))x'(t) + f(x(t)) \\ x'(t) &= -\left[f'(x(t))\right]^{-1} f(x(t)) \end{aligned}$$

integrating with **Euler's method** with step size **1** results in **Newton's method**:

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

Conclusions

- homotopy can be any continuous connection between two functions
- we solve the problem $f(x) = 0$ by solving the series of the problems

$$h(t_0, x) = 0,\ h(t_1, x) = 0,\ \ldots,\ h(t_m, x) = 0$$

- homotopy method requires the numerical solution systems of differential equation.
- Newton's method is a subproblem of homotopy method

References

Lecture notes on

Nonlinear equations with one variable, EE103, L. Vandenberhge, UCLA

Lecture notes on

Solution of Nonlinear Equations, EE507, D. Banjerdponchai, Chulalongkorn University

Chapter 2 in

J. F. Epperson, *An Introduction to Numerical Methods and Analysis*, John Wiley & Sons, 2007

Chapter 3 in

D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd edition, Brooks & Cole, 2002