Chapter 3: Nonlinear equations in one variable

Dhavide Aruliah (modified from U. Ascher)

UBC Computer Science

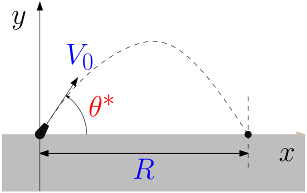
Department of Computer Science
University of British Columbia
dhavide@cs.ubc.ca
http://www.cs.ubc.ca/∼dhavide/cpsc302/

Goals of this chapter

- to develop practical methods for a basic, simply stated problem
  - includes favourites such as fixed point iteration, Newton's method
- to develop important algorithmic concepts in numerical computing
- to study basic algorithms for minimizing a function in one variable

Example: Range of a cannonball

To what elevation should the cannon be raised to hit the target?

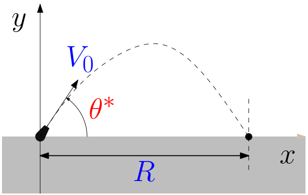


Parameters:

- g is acceleration of gravity (in ms^{-2}): known
- V_0 is initial speed (in ms^{-1}): known
- R is distance to target (in m): known
- θ^* is required elevation (in radians): unknown

- Determine elevation θ^* needed to hit target at horizontal distance R (values of parameters V_0, R, and g given a priori)
- Integrating Newton's 2nd law yields single nonlinear equation in θ^*

Remarks: Range of a cannonball



$$f(\theta^*) = 2\sin\theta^*\cos\theta^* - \frac{Rg}{{V_0}^2}$$
$$= 0$$

- Idealisation: no air resistance
- No solution if $\frac{Rg}{{V_0}^2} > 1$
- Solution nonunique ($\sin$ & $\cos$ periodic)
- Meaningful θ^* satisfies $0 < \theta^* < \pi/2$
- Analytic formula for θ^*:

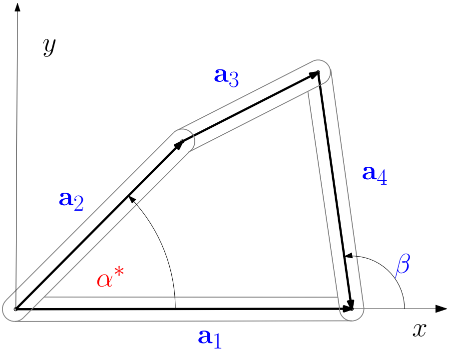
$$\theta^* = \frac{1}{2}\arcsin\left(\frac{Rg}{{V_0}^2}\right)$$

(uses $2\sin\theta\cos\theta = \sin(2\theta)$)

Example: Configuration of rods

Parameters:

- $a_k = |\mathbf{a}_k|$ is length of rod k (in m): known ($k = 1:4$)
- β is angle of rod 4 from horizontal (in radians): known
- α^* is angle of rod 2 from horizontal (in radians): unknown



Determine angle α^* from values a_1, a_2, a_3, a_4, and β

Example: Configuration of rods

- Orientations of rigid rods in plane implies

$$\mathbf{a}_1 - \mathbf{a}_2 - \mathbf{a}_3 - \mathbf{a}_4 = \mathbf{0}$$

- Rod 1 fixed on x axis implies Freudenstein equation

$$\frac{a_1}{a_2}\cos(\beta) - \frac{a_1}{a_4}\cos(\alpha^*) - \cos(\beta - \alpha^*) = -\frac{a_1^2 + a_2^2 - a_3^2 + a_4^2}{2a_2a_4}$$

(derive as an exercise in trigonometry or analytic geometry)

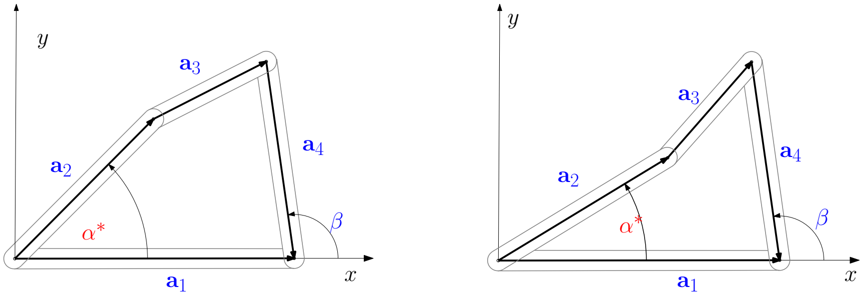
- Zero-finding problem: find angle α^* such that $f(\alpha^*) = 0$, where

$$f(\alpha) := \frac{a_1}{a_2}\cos(\beta) - \frac{a_1}{a_4}\cos(\alpha) - \cos(\beta - \alpha) + \frac{{a_1}^2 + {a_2}^2 - {a_3}^2 + {a_4}^2}{2a_2a_4}$$

Remarks: Configuration of rods

$$f(\alpha^*) = \frac{a_1}{a_2}\cos(\beta) - \frac{a_1}{a_4}\cos(\alpha^*) - \cos(\beta - \alpha^*) + \frac{{a_1}^2 + {a_2}^2 - {a_3}^2 + {a_4}^2}{2a_2a_4} = 0$$

- Given any 5 of 6 parameters, different zero-finding problem arises
- Solution may have rods overlapping; possibly unphysical
- f may have several zeros α^* for fixed $a_1, a_2, a_3, a_4, \beta$



Zero-finding problems

Zero of nonlinear equation

Given $f : [a,b] \subset \mathbb{R} \to \mathbb{R}$, determine $x^* \in [a,b]$ such that $f(x^*) = 0$

- Solutions of nonlinear equations are zeros
- Classical terminology: "roots" of nonlinear equations
  terminology originates from roots of polynomials, e.g.,

$$f(x) := x^2 - 2 \Rightarrow \left[f(x^*) = 0 \Rightarrow x^* = \pm\sqrt{2} \right]$$

- Terminology: zero-finding or root-finding

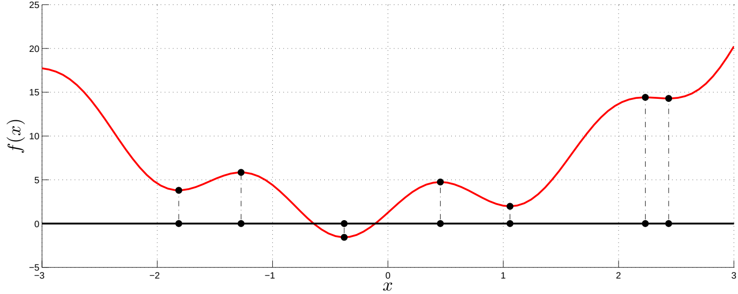
Related problem: minimisation of functions

- Many practical problems involve *optimisation*
- Minimum of f coincides with zero of f' where $f'' > 0$

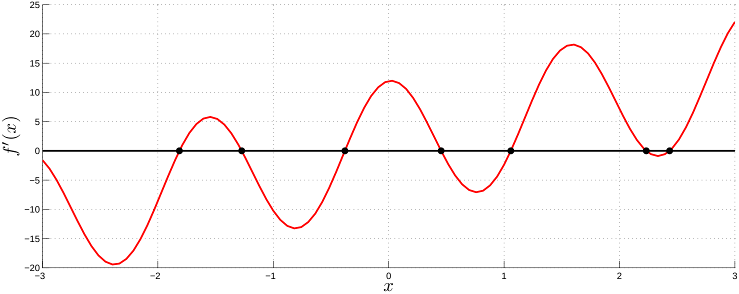
$\Rightarrow$ Optimisation closely linked with zero-finding

- Terminology: zero of f' is *critical point* of f

Graph of f including local extrema of f



Graph of f' including zeros of f'



Remarks

- Any nonlinear equation in x can be expressed as $f(x) = 0$

  "Find $\alpha \in \mathbb{R}$ such that $2 - \alpha = \cos\alpha$"

  equivalent to

  "Find a zero of f where $f(x) = x - \cos x - 2$"

- Solutions generally not unique

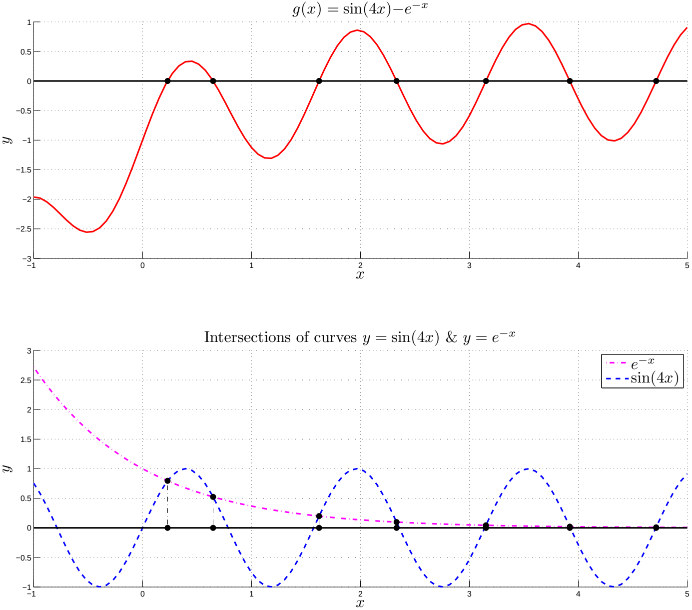
  e.g., $f(x) = -x^3 + 6x^2 - 4x - 8$ has 3 real zeros

  e.g., $f(x) = \sin 4x - e^{-x}$ has infinitely many zeros

- Solutions not generally attainable in closed form, i.e., as formulas
- Solutions could be real or complex

  e.g., $f(x) = x^3 - 2x^2 + 4x - 8 \Rightarrow$ zeros $x^* = 2$ & $x^* = \pm 2i$

Example: Zeros of $g(x) = \sin 4x - e^{-x}$



Reformulations of nonlinear equations

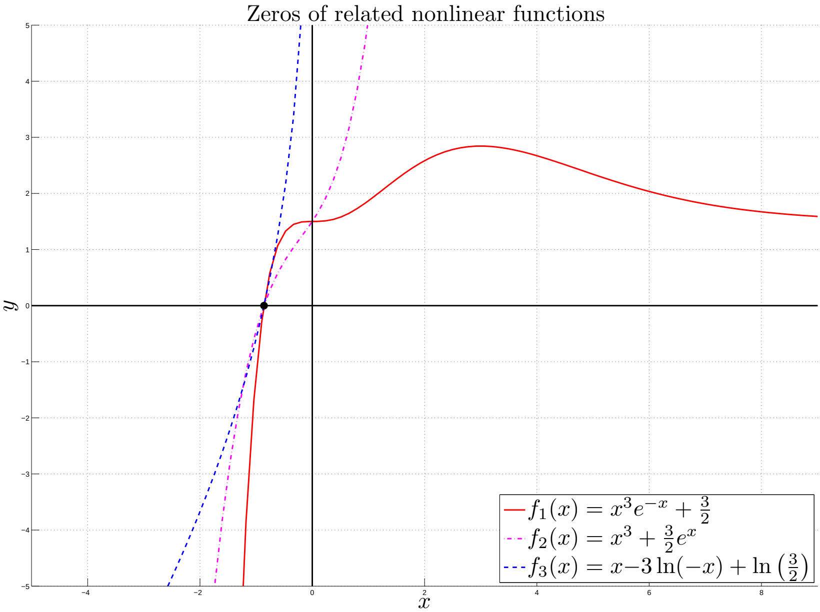
- Nonlinear equations typically admit reformulations
- Applying invertible functions to both sides preserves solutions
- e.g., consider the functions below:

$$f_1(x) = x^3 e^{-x} + \frac{3}{2} \qquad g_1(x) = -x^3 + 6x^2 - 4x - 8$$

$$f_2(x) = x^3 + \frac{3}{2} e^x \qquad g_2(x) = x - 6 + \frac{4}{x} + \frac{8}{x^2}$$

$$f_3(x) = x - 3\ln(-x) + \ln\left(\frac{3}{2}\right)$$

  - f_1, f_2, and f_3 have the same zeros
  - g_1 and g_2 have the same zeros
- Transformations can help or hinder solver performance!


Zeros of related nonlinear functions
5
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−2
0
2
4
6
8
x
y
$f_1(x) = x^3 e^{-x} + \frac{3}{2}$
$f_2(x) = x^3 + \frac{3}{2} e^x$
$f_3(x) = x - 3\ln(-x) + \ln\left(\frac{3}{2}\right)$

Roots of continuous functions

- *How many roots?* Depends not only on function f but also on interval $[a, b]$
  Example: $f(x) = \sin(x)$ has one root in $[-\pi/2, \pi/2]$, two roots in $[-\pi/4, 5\pi/4]$, and no roots in $[\pi/4, 3\pi/4]$.
- Why study nonlinear problems before linear ones?!
  - Single variable linear equation too simple (e.g., $ax = b \Rightarrow$ solution: $x = b/a$) However, systems of linear equations (Chapters 5, 7) have many complications.
  - Several important general methods can be described in a simple context.
  - Several important algorithm properties can be defined and used in a simple context.

Desirable algorithm properties

Generally for nonlinear problems, must use an iterative method: starting with initial iterate (guess) x_0, generate sequence of iterates $x_1, \; x_2, \; \ldots, \; x_k, \ldots$ that hopefully converge to a root x^*.

Desirable properties of a contemplated iterative method are:

- *Efficient*: requires a small number of function evaluations.
- *Robust*: fails rarely, if ever. Announces failure if it does fail.
- Requires *minimal additional information* such as the derivative of f.
- Requires f to satisfy only *minimal smoothness properties*.
- *Generalizes easily* to systems of nonlinear equations in many unknowns.

Like many other wish lists, this one is hard to fully satisfy...

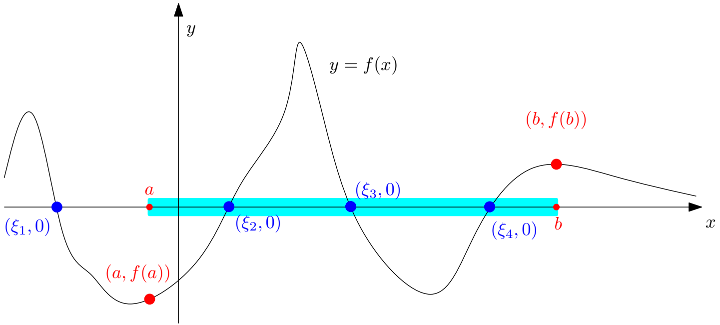
Bisection method properties

- Simple
- Safe, robust
- Requires only that f be continuous
- Slow
- Hard to generalize to systems

The Intermediate value theorem

Theorem (Intermediate value theorem)

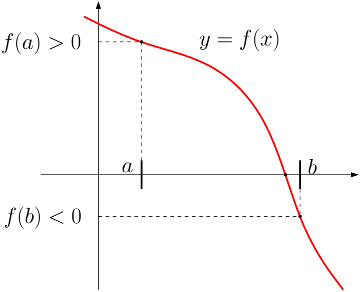
Given a continuous function $f : [a, b] \to \mathbb{R}$ *such that* $f(a)f(b) < 0$, *then there exists at least one* $\xi \in (a, b)$ *such that* $f(\xi) = 0$.



Bracketing a zero

- Assume f is a continuous function
- Any interval $[a, b]$ for which either
    - $f(a) > 0$ and $f(b) < 0$; or
    - $f(a) < 0$ and $f(b) > 0$

  is said to bracket a zero of f
- Interval can be called a bracket
- Equivalent criterion: $f(a)f(b) < 0$



Assumption of continuity essential for interval to bracket a zero

Example

Assumption of continuity essential for interval to bracket a zero

- Consider g defined by

$$g(x) = x - \cot x$$

- Sample g uniformly yields
- It seems g has three zeros:
  - One in $[0.75, 1.25]$
  - One in $[2.75, 3.25]$
  - One in $[3.25, 3.75]$
- g discontinuous in $[2.75, 3.25]$

$\Rightarrow$ IVT does not guarantee zero in interval $[2.75, 3.25]$

x	$g(x)$
0.75	-0.3234
1.25	0.9177
1.75	1.9311
2.25	3.0573
2.75	5.1718
3.25	-5.9383
3.75	2.3144

Exercise: construct table in MATLAB
Exercise: plot table data in MATLAB
Exercise: plot $g(x)$ vs. x in MATLAB with table data neatly

Bisection method development

- Given $a < b$ such that $f(a) \cdot f(b) < 0$, there must be a root in uncertainty interval $[a, b]$ (provided f is continuous).
- Let's save the input (initial interval of uncertainty): $\hat{a} = a,\ \hat{b} = b$.
- At each iteration: compute midpoint $p = \frac{a+b}{2}$, evaluate $f(p)$.
  Check sign of $f(a) \cdot f(p)$.
  If positive, set $a \leftarrow p$, if negative set $b \leftarrow p$.
  Note: only one evaluation of function f required per iteration.
- Reduces length of uncertainty interval by factor 2 at each iteration.
  Setting $x_n = p$, error after n iterations satisfies

$$|x^* - x_n| \leq \frac{\hat{b} - \hat{a}}{2} \cdot 2^{-n}.$$

- Stopping criterion: $\frac{\hat{b} - \hat{a}}{2} \cdot 2^{-n} \leq \texttt{atol}$ (`atol` is (absolute) tolerance)
- Number of iterations n can be determined a priori:
  unusual in algorithms for nonlinear problems.

bisect function

```
function [p,n] = bisect(func,a,b,fa,fb,atol)
if (a >= b) | (fa*fb >= 0) | (atol <= 0)
   disp('something wrong with the input:  quitting');
   p = NaN; n=NaN;
   return
end
n = ceil ( log2 (b-a) - log2 (2*atol));
for k=1:n
   p = (a+b)/2;
   fp = feval(func,p);
   if fa * fp < 0
     b = p;
     fb = fp;
   else
     a = p;
     fa = fp;
   end
end
p = (a+b)/2;
```

Bisection accuracy and iteration count

To find a root of the function $f(x) = (x-2)(x^2+x+1)$ in the interval $[1, 2.5]$, 20 bisection iterations were applied, giving an absolute error of ≈2.4e-7.
How many additional iterations are required to achieve absolute error ≈3.0e-8?

(A) 1
(B) 2
(C) 3
(D) 4

A B C D

Fixed point iteration

This is an intuitively appealing approach which often leads to simple algorithms for complicated problems.

1. Write given problem

   $$f(x) = 0$$

   as

   $$g(x) = x,$$

   so that $f(x^*) = 0$ iff $g(x^*) = x^*$.
2. Iterate:

   $$x_{k+1} = g(x_k), \quad k = 0, 1, \ldots,$$

   starting with guess x_0.

It's all in the choice of the function g.

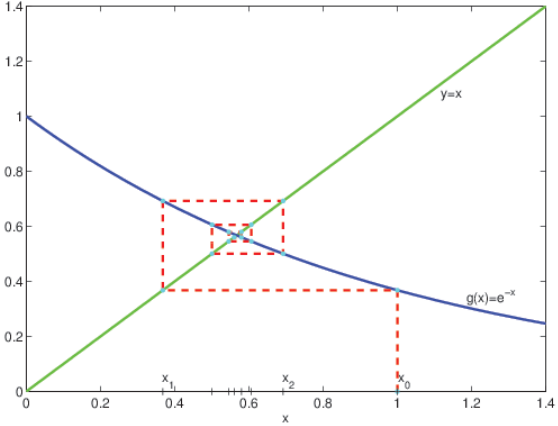
Choosing the function g

- Note: there are many possible choices g for the given f: this is a family of methods.
- Examples:

$$\begin{aligned} g(x) &= x - f(x), \\ g(x) &= x + 2f(x), \\ g(x) &= x - f(x)/f'(x) \quad (\text{assuming f}' \text{ exists and f}'(\text{x}) \neq 0). \end{aligned}$$

  The first two choices are simple, the last one has potential to yield fast convergence (we'll see later).
- Want resulting method to
  - be simple;
  - converge; and
  - do it rapidly.

Graphical illustration, $x = e^{-x}$, starting from $x_0 = 1$



Fixed Point Theorem

If $g \in C[a,b],\ g(a) \geq a$ *and* $g(b) \leq b$*, then there is a fixed point* x^* *in the interval* $[a,b]$*.*
If, in addition, the derivative g' *exists and there is a constant* $\rho < 1$ *such that the derivative satisfies*

$$|g'(x)| \leq \rho \quad \forall\, x \in (a,b),$$

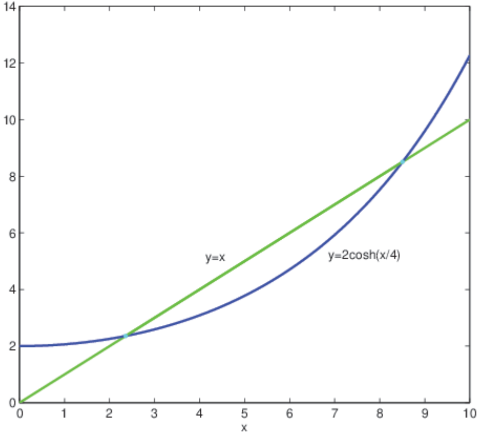
then the fixed point x^* *is unique in this interval.*

Convergence of fixed point iteration

- Assuming $\rho < 1$ as for the fixed point theorem, obtain
$$|x_{k+1} - x^*| = |g(x_k) - g(x^*)| \leq \rho |x_k - x^*|.$$
- This is a **contraction** by factor ρ.
- So
$$|x_{k+1} - x^*| \leq \rho |x_k - x^*| \leq \rho^2 |x_{k-1} - x^*| \leq \ldots \leq \rho^{k+1} |x_0 - x^*| \to 0.$$
- The smaller ρ the faster convergence is.

Example: $\cosh$ with two roots

$$f(x) = g(x) - x, \quad g(x) = 2\cosh(x/4)$$



Fixed point iteration with g

For tolerance 1.e-8:

- Starting at $x_0 = 2$ converge to x_1^* in 16 iterations.
- Starting at $x_0 = 4$ converge to x_1^* in 18 iterations.
- Starting at $x_0 = 8$ converge to x_1^* (even though x_2^* is closer to x_0).
- Starting at $x_0 = 10$ obtain **overflow** in 3 iterations.

Note: bisection yields both roots in 27 iterations.

Rate of convergence

- Suppose we want $|x_k - x^*| \approx 0.1|x_0 - x^*|$.
- Since $|x_k - x^*| \leq \rho^k |x_0 - x^*|$, want

$$\rho^k \approx 0.1,$$

i.e., $k \log_{10} \rho \approx -1$.

- Define the **rate of convergence** as

$$rate = -\log_{10} \rho.$$

- Then it takes about $k = \lceil 1/rate \rceil$ iterations to reduce the error by more than an order of magnitude.

Return to cosh example

- Bisection: $rate = -\log_{10} 0.5 \approx .3 \quad \Rightarrow \quad k = 4$.
- For the root x_1^* of fixed point example, $\rho \approx 0.31$ so

$$rate = -\log_{10} 0.31 \approx .5, \quad \Rightarrow \quad k = 2.$$

- For the root x_2^* of fixed point example, $\rho > 1$ so

$$rate = -\log_{10}(\rho) < 0, \quad \Rightarrow \quad \text{no convergence.}$$

Convergence of fixed point iteration

For a positive parameter λ, the quadratic equation $\lambda x^2 - 2x - 1 = 0$ has the root $x = \frac{1+\sqrt{1+\lambda}}{\lambda}$. Consider applying fixed point iteration to the equation $x = g(x)$ with $g(x) = \frac{1}{2}\left(\lambda x^2 - 1\right)$. Assume that the initial iterate x_0 is near x . For which values of $\lambda > 0$ will this iteration converge?

(A) $\lambda \in (0, 1)$

(B) $\lambda = 1$

(C) The iteration converges for any $\lambda > 0$.

(D) The iteration diverges for any $\lambda > 0$.

A B C D

Newton's method

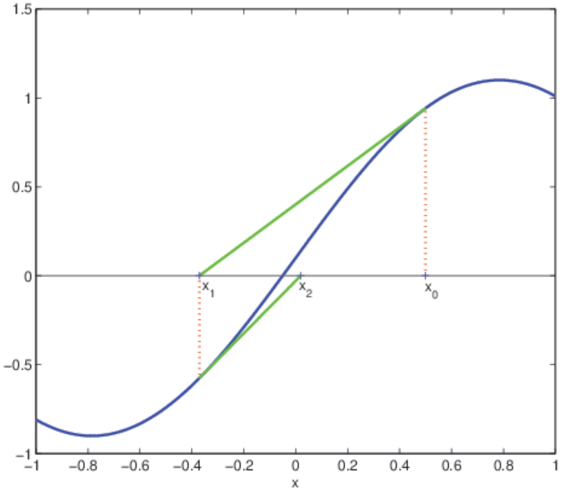
This fundamentally important method is everything that bisection is not, and vice versa:

- Not so simple
- Not very safe or robust
- Requires more than continuity on f
- Fast
- Automatically generalizes to systems

Derivation

- By Taylor series,
$$f(x) = f(x_k) + f'(x_k)(x - x_k) + f''(\xi(x))(x - x_k)^2/2.$$
- So, for $x = x^*$
$$0 = f(x_k) + f'(x_k)(x^* - x_k) + \mathcal{O}\left((x^* - x_k)^2\right).$$
- The method is obtained by neglecting nonlinear term, defining $0 = f(x_k) + f'(x_k)(x_{k+1} - x_k)$, which gives the iteration step
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots.$$

A geometric interpretation



Next iterate is x-intercept of the tangent line to f at current iterate.

Example: cosh with two roots

- The function
$$f(x) = 2\cosh(x/4) - x$$
has two solutions in the interval $[2, 10]$.
- Newton's iteration is
$$x_{k+1} = x_k - \frac{2\cosh(x_k/4) - x_k}{0.5\sinh(x_k/4) - 1}.$$
- For absolute tolerance 1.e-8:
  - Starting from $x_0 = 2$ requires 4 iterations to reach x_1^*.
  - Starting from $x_0 = 4$ requires 5 iterations to reach x_1^*.
  - Starting from $x_0 = 8$ requires 5 iterations to reach x_2^*.
  - Starting from $x_0 = 10$ requires 6 iterations to reach x_2^*.
- Tracing the iteration's progress:

k	0	1	2	3	4	5
$f(x_k)$	-4.76e-1	8.43e-2	1.56e-3	5.65e-7	7.28e-14	1.78e-15

- Note that the number of significant digits essentially doubles at each iteration (until the 5th, when roundoff error takes over).

Speed of convergence

A given method is said to be

- **linearly convergent** if there is a constant $\rho < 1$ such that

$$|x_{k+1} - x^*| \leq \rho |x_k - x^*| \ ,$$

  for all k sufficiently large;

- **quadratically convergent** if there is a constant M such that

$$|x_{k+1} - x^*| \leq M |x_k - x^*|^2 \ ,$$

  for all k sufficiently large;

- **superlinearly convergent** if there is a sequence of constants $\rho_k \to 0$ such that

$$|x_{k+1} - x^*| \leq \rho_k |x_k - x^*|,$$

  for all k sufficiently large.

Convergence theorem for Newton's method

If $f \in C^2[a,b]$ and there is a root x^* in $[a,b]$ such that $f(x^*) = 0,\ f'(x^*) \neq 0$, then there is a number δ such that, starting with x_0 from anywhere in the neighborhood $[x^* - \delta, x^* + \delta]$, Newton's method converges quadratically.

Idea of proof:

- Expand $f(x^*)$ in terms of a Taylor series about x_k;
- divide by $f'(x_k)$, rearrange, and replace $x_k - \frac{f(x)}{f'(x_k)}$ by x_{k+1};
- find the relation between $e_{k+1} = x_{k+1} - x^*$ and $e_k = x_k - x^*$.

Secant method

- One potential disadvantage of Newton's method is the need to *know and evaluate* the derivative of f.
- The secant method circumvents the need for explicitly evaluating this derivative.
- Observe that near the root (assuming convergence)

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

- So, define Secant iteration

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, \ldots.$$

- Note the need for two initial starting iterates x_0 and x_1: a *two-step method*.

Example: cosh with two roots

$$f(x) = 2\cosh(x/4) - x.$$

Same absolute tolerance 1.e-8 and initial iterates as before:

- Starting from $x_0 = 2$ and $x_1 = 4$ requires 7 iterations to reach x_1^*.
- Starting from $x_0 = 10$ and $x_1 = 8$ requires 7 iterations to reach x_2^*.

k	0	1	2	3	4	5	6
$f(x_k)$	2.26	-4.76e-1	-1.64e-1	2.45e-2	-9.93e-4	-5.62e-6	1.30e-9

Observe superlinear convergence: much faster than bisection and simple fixed point iteration, yet not quite as fast as Newton's iteration.

Newton's method as a fixed point iteration

- If $g'(x^*) \neq 0$ then fixed point iteration converges linearly, as discussed before, as $\rho > 0$.
- Newton's method can be written as a fixed point iteration with

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

From this we get $g'(x^*) = 0$.

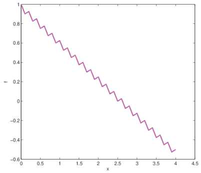
- In such a situation the fixed point iteration may converge faster than linearly: indeed, Newton's method converges quadratically under appropriate conditions.

Minimizing a function in one variable

- Optimization is a vast subject, only some of which is covered in Chapter 9. Here, we just consider the simplest situation of minimizing a smooth function in one variable.
- **Example**: find $x = x^*$ that minimizes

$$\phi(x) = 10\cosh(x/4) - x.$$

- From the figure below, this function has no zeros but does appear to have one minimizer around $x = 1.6$.



Conditions for optimum and algorithm

- Necessary condition for an optimum:
  Suppose $\phi \in C^2$ and denote $f(x) = \phi'(x)$. Then a zero of f is a critical point of ϕ, i.e., where

  $$\phi'(x^*) = 0.$$

  To be a minimizer or a maximizer, it is necessary for x^* to be a critical point.
- Sufficient condition for an optimum:
  A critical point x^* is a minimizer if also $\phi''(x^*) > 0$.
- Hence, an algorithm for finding a minimizer is obtained by using one of the methods of this chapter for finding the roots of $\phi'(x)$, then checking for each such root x^* if also $\phi''(x^*) > 0$.
- Note: rather than finding all roots of ϕ' first and checking for minimum condition later, can do things more carefully and wisely, e.g. by sticking to steps that decrease $\phi(x)$.

Example

To find a minimizer for

$$\phi(x) = 10\cosh(x/4) - x,$$

1. Calculate gradient

$$f(x) = \phi'(x) = 2.5\sinh(x/4) - 1$$

2. Find root of $\phi'(x) = 0$ using any of our methods, obtaining

$$x^* \approx 1.56014.$$

3. Second derivative

$$\phi''(x) = 2.5/4\cosh(x/4) > 0 \quad \text{for all } x,$$

so x^* is a minimizer.