Nonlinear equations and systems

An equation is **nonlinear** if it involves nonlinear components of the unknown, such as powers of the unknown (with exponent other than unit, e.g. x^3, $x^{1/4}$, x^{-2} or $\frac{1}{x^2}$, etc.) or other nonlinear functions of the unknown (e.g. $\sin x$, $\log x$, e^x, etc.). In general, a non-linear equation takes the form

$$f(x) = 0,$$

where $f\colon \mathbb{R} \to \mathbb{R}$ is a nonlinear function of $x \in \mathbb{R}$.
A number x such that

$$f(x) = 0$$

is called a **root** (or **zero** or **solution**) of the equation (or of f). A nonlinear equation may have one or more roots, or no roots.

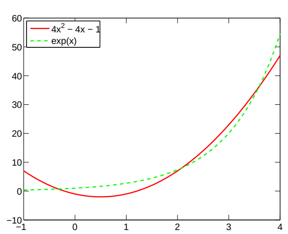
Nonlinear equations and systems -- examples of nonlinear equations

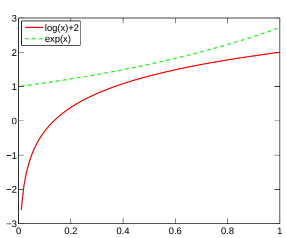
Example:

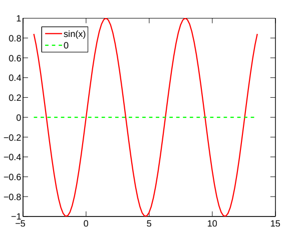
Let $f(x) = 4x^2 - 4x - 1 - e^x$. The equation $f(x) = 0$ has 3 roots. We can visualize where the roots lie, if we plot the quadratic $4x^2 - 4x - 1$ and the function e^x. The roots of $f(x) = 0$ are the points where the two graphs match.

Let $f(x) = \ln x - e^x$. The equation $f(x) = 0$ has no roots.

Let $f(x) = \sin x$. The equation $f(x) = 0$ has roots $x = \kappa\pi$, κ integer (infinite number of roots).







Nonlinear equations and systems -- systems of nonlinear equations

A nonlinear system of equations is a set of equations of which at least one (and usually most or all) is (are) nonlinear, and which involve an unknown vector. Non-linearity may be taking place between different components of the unknown vector (e.g $x_1\, x_2$, $x_1 \sin x_2$, etc.) or within the same component (e.g. x_1^3). In general, a nonlinear system of equations takes the form

$$f(x) = 0,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear vector function of $x \in \mathbb{R}^n$, i.e.

$$f = \begin{bmatrix} f_1(x_1, x_2, \cdots, x_n) \\ f_2(x_1, x_2, \cdots, x_n) \\ \cdots \\ f_n(x_1, x_2, \cdots, x_n) \end{bmatrix}$$

and 0 is the zero vector in $\mathbb{R}^n$.
A vector $x \in \mathbb{R}^n$ such that

$$f(x) = 0$$

is called a **root** (or **zero** or **solution**) of the system. A nonlinear system may have one or more roots, or no roots.

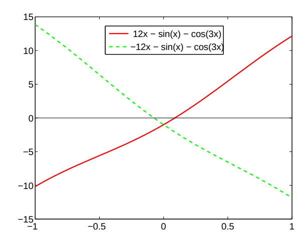
Nonlinear equations and systems -- systems of nonlinear equations

Example: The 2×2 nonlinear system

$9x_1^2 - x_2^2 = 0$

$4x_2 - \sin x_1 - \cos x_2 = 0$

has two solutions.



General nonlinear systems

There are also non-square nonlinear systems: in those, the function f has m components (component functions) and n variables, i.e. $f\colon \mathbb{R}^n \to \mathbb{R}^m$. Then, the arising system has m equations and n unknowns, i.e. it is $m \times n$.

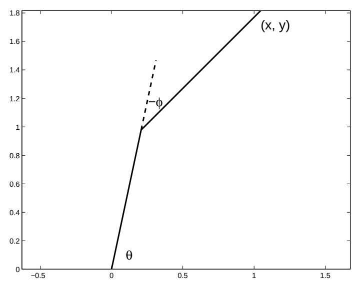
We will first study simple equations (1×1 systems), then $n \times n$ (square) systems.

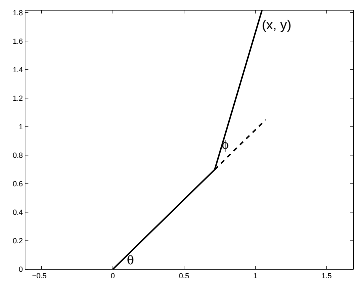
Nonlinear equations and systems -- why?

Nonlinear equations arise often in many applications.

Example: Assume we have a robot arm with two joints connecting two parts of length a and b, respectively, to each other and to a solid base. Let θ be the angle between the x-axis and the first arm part, and ϕ be the angle between the first and second arm parts. (Note the $-\phi$ in the left figure.) The tip of the arm is in position with coordinates

$$x = a\cos(\theta) + b\cos(\theta + \phi)$$
$$y = a\sin(\theta) + b\sin(\theta + \phi)$$





To program the arm to reach a given location (x, y), we need to solve the above 2×2 system of nonlinear equations with respect to θ and ϕ.

Multiplicity of nonlinear equations' roots

If $f \in \mathbb{C}^m$ (i.e. f is m times differentiable with continuous derivatives) and we have $f(x^*) = 0$, $f'(x^*) = 0, \cdots, f^{(m-1)}(x^*) = 0$, **but** $f^{(m)}(x^*) \neq 0$, for some $m \geq 1$, then x^* is a root of **multiplicity** m. If $m > 1$, then x^* is a **multiple** root, while, if $m = 1$, then x^* is a **simple** root.

Examples:
$x^2 - 2x + 1 = 0$ has a double root (multiplicity 2) at $x = 1$.
$x^3 - 3x^2 + 3x - 1 = 0$ has a triple root (multiplicity 3) at $x = 1$.
$\sin(x) - 1 = 0$ has a double root (multiplicity 2) at $x = \dfrac{\pi}{2}$.

Note: The term «multiple», when used in the context of function roots, may have a two-fold meaning:
«f has multiple roots», means one of the two
(i) «some roots of f have multiplicity more than 1», and
(ii) «f has more than 1 roots».
With caution, we can infer the correct meaning from the context.

Fixed points and roots of functions -- systolic or contractive functions

The point x is called **fixed point** of the function $g(x)$, when $x = g(x)$.
A function $g\colon \mathbb{R}^n \to \mathbb{R}^n$ is called **contractive** (or **systolic** or just **contraction**) in a set $S \subset \mathbb{R}^n$, if there exists a constant (scalar) λ, with $0 \leq \lambda < 1$, such that

$$\|g(x) - g(z)\| \leq \lambda \|x - z\|, \quad \forall\ x, z \in S. \qquad (1)$$

Geometrically, for $n = 1$, relation (1) means that the slope of the chord that joins any two points in S on the graph of a contraction g is less than or equal to λ.

If g is differentiable in S, and $n = 1$, relation (1) is equivalent to

$$|g'(x)| \leq \lambda, \quad \forall\ x \in S. \qquad (2)$$

Let $f(x) = 0$ the nonlinear equation to be solved, and let x a root.
Assume we have found (constructed) a function g, such that, when x is root of $f(x) = 0$, then x is a fixed point of $g(x)$, and vice-versa. That is, g is constructed so that

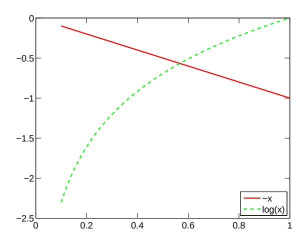
$$f(x) = 0 \iff x = g(x) \qquad (3)$$

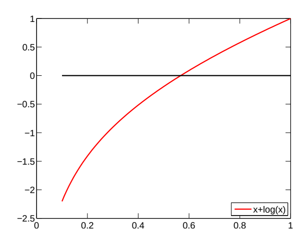
Then, finding of a root of f is equivalent to finding a fixed point of g.

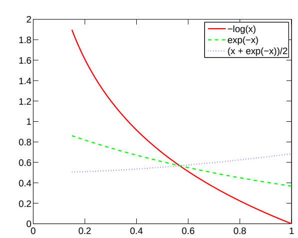
Fixed points and roots of functions

In several situations, it is more convenient to find a fixed point of g instead of a root of f. Thus, finding roots is highly connected to finding fixed points.

For a given function f, there can be many functions g such that (3) holds, i.e. $f(x) = 0 \iff x = g(x)$.
Examples:
Let $f(x) = x + \ln x$. The following g's satisfy (3): $g(x) = -\ln x$, $g(x) = e^{-x}$, $g(x) = \dfrac{x + e^{-x}}{2}$.







Existence and uniqueness of root and fixed point

Mathematically proving the existence of a root and/or fixed point of a given function is not usually a simple task, let alone the proof of their uniqueness.
There are though some theorems that give *sufficient* conditions under which the existence and/or uniqueness of roots or fixed points is guaranteed.

Reminder:

Intermediate Value Theorem (IVP):
If
− f is continuous in $[a, b]$,
then
− for all (scalar) $\gamma \in (f(a), f(b))$, there exists $\xi \in (a, b)$ such that $f(\xi) = \gamma$.

Mean Value Theorem (MVP):
If
− f is continuous in $[a, b]$ and differentiable in (a, b)
then
− there exists $\xi \in (a, b)$ such that $f'(\xi) = \dfrac{f(b) - f(a)}{b - a}$

Theorems for existence of root and fixed point

Let g and f be connected by the relation $g(x) = x \iff f(x) = 0$. The following theorems give sufficient (but not necessary) conditions for the respective conclusions.

Theorem 1 (Bolzano): **Existence** of a root of $f(x)$
If
− $f(x)$ is continuous in $[a, b]$
− $f(a) \cdot f(b) < 0$,
then
− there exists at least one root of $f(x)$ in (a, b).
Proof based on Intermediate Value Theorem.

Theorem 2: **Existence** of a fixed point of $g(x)$
If
− $g(x)$ is continuous $I = [a, b]$
− $g(x) \in I, \forall\, x \in I$ (i.e. $g(x)$ maps I to itself, $g(x)\colon I \to I$),
then
− there exists at least one fixed point of $g(x)$ in I.
Proof based on Theorem 1.

Theorems for existence and/or uniqueness of root and fixed point

Theorem 3: **Uniqueness** of a root of $f(x)$
If
− $f(x)$ is differentiable in $I = (a, b)$
− $f'(x) \ne 0, \forall\, x \in I$
− there exists a root of $f(x)$ in I
then
− the root is unique in I.
Proof based on Mean Value Theorem.

Theorem 4: **Existence** and **uniqueness** of a fixed point of $g(x)$
If
− g is contraction in $I = [a, b]$
− $g(x) \in I, \forall\, x \in I$ (i.e. $g(x)$ maps I to itself),
then
− there exists a unique fixed point x^* of $g(x)$ in I.
Proof based on Theorem 2 and contraction properties.

Numerical methods for solving nonlinear equations

Most nonlinear equations and systems cannot be solved by standard mathematical techniques, that is, we cannot arrive to an analytic formula for their roots. Numerical techniques are used to approximate roots. The numerical techniques for solving nonlinear equations and systems are iterative. (Hence, the term **nonlinear solver** usually refers to an iterative method for the solution of nonlinear equations or systems.)

The general strategy of nonlinear solvers is that we start with some initial guess $x^{(0)}$ and compute successive approximations $x^{(k)}$ to the root (solution) x, for $k = 1, \cdots$ until some **stopping criterion** is satisfied. (Note: some nonlinear solvers may need more than one initial guess to start.)

The most common initial guess is $x^{(0)} = 0$.

The most common stopping criteria are
$\dfrac{\| f(x^{(k)})\|}{\| f(x^{(0)})\|} \le \varepsilon$ (relative), or $\| f(x^{(k)})\| \le \varepsilon$ (absolute),
where ε is the desired precision of the approximation, or
$\dfrac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k+1)}\|} \le \varepsilon$, or $\|x^{(k+1)} - x^{(k)}\| \le \varepsilon$.

Numerical methods for solving nonlinear equations

The quantity (vector if $n > 1$) $f(x^{(k)})$ is called **residual** of f at $x^{(k)}$.
(Absolute) norm of residual: $\| f(x^{(k)}) \|$; Relative norm of residual: $\| f(x^{(k)}) \| / \| f(x^{(0)}) \|$.

None of the stopping criteria are "perfect", that is, $x^{(k)}$ may satisfy a certain stopping criterion, but may still be far from the root.

- For $n = 1$, it can be shown that if $| f(x^{(k)}) | \le \varepsilon$, then $| x^{(k)} - x^* | \approx \dfrac{\varepsilon}{| f'(x^*) |}$. Therefore, if $| f'(x^*) |$ is close to 0, then $| x^{(k)} - x^* |$ may be large, while $| f(x^{(k)}) | \le \varepsilon$.
- When an iterative method converges slowly, we may have $\| x^{(k)} - x^{(k-1)} \| \le \varepsilon$, while $\| x^{(k)} - x^* \|$ is large.

Iterative methods for nonlinear equations do not always **converge** (i.e. a criterion may never be reached). There are cases though that they converge rapidly, and give satisfactory approximation in a few iterations.

Then, there are more issues rising: how do we avoid divergent iterations; how do we construct rapidly convergent iterations; how do we quantify the speed of convergence, and more.

Nonlinear solvers

The general form of a nonlinear solver is
guess $x^{(0)}$ (and possibly $x^{(-1)}$ or more)
for $k = 1, \cdots,$ maxit
  compute $x^{(k)}$ using previous approximations and information from f
  if stopping criterion satisfied, exit, endif
endfor

If the computation of $x^{(k)}$ uses only the immediate previous approximation $x^{(k-1)}$, then the method is a **one-step** method, and needs only one initial guess. If the computation of $x^{(k)}$ uses $x^{(k-1)}$ and $x^{(k-2)}$, then the method is a **two-step** method and needs two initial guesses, etc.

An important issue that arises is how is $x^{(k)}$ computed using previous approximations and f, i.e., how we derive a formula for $x^{(k)}$.

We will introduce techniques for constructing nonlinear solvers, (i.e. ways to compute $x^{(k)}$), describe some commonly used nonlinear solvers, study sufficient conditions for their convergence, and introduce ways to measure the speed of convergence.

Convergence speed of iterative methods

A sequence (of numbers, vectors, etc.) $x^{(0)}, x^{(1)}, \cdots, x^{(n)}, \cdots$ converges to α, if, for all $\varepsilon > 0$, there exists n_0, such that, for all $n \ge n_0$, we have $\| x^{(n)} - \alpha \| < \varepsilon$.

We are not, though, only interested in whether a sequence converges, but also how fast it converges. This is important when assessing the efficiency of several iterative methods that produce sequences of approximations converging to the solution of a problem.

A measure of the speed of convergence is the **rate** (a.k.a. **order**) of convergence.

A sequence (of numbers, vectors, etc.) $x^{(0)}, x^{(1)}, \cdots, x^{(n)}, \cdots$ converges to α with order $\beta \ge 1$, if there exists a constant $\kappa > 0$ such that

$$\lim_{k \to \infty} \frac{\| x^{(k+1)} - \alpha \|}{\| x^{(k)} - \alpha \|^{\beta}} = \kappa$$

The constant κ is called the **asymptotic error constant**.

- If $\beta = 1$, then $\kappa \le 1$ is a necessary condition for convergence, and $\kappa < 1$ is a necessary and sufficient condition for convergence.
- If $\beta > 1$, then the constant κ plays a small role, assuming the initial approximation $x^{(0)}$ to α is close enough to α.

Convergence speed of iterative methods

Convergence at rate α, with asymptotic error constant κ:

$$\lim_{k \to \infty} \frac{\| x^{(k+1)} - \alpha \|}{\| x^{(k)} - \alpha \|^{\beta}} = \kappa$$

Typical and special cases
If $\beta = 1$, $\kappa = 1$, sublinear convergence
If $\beta = 1$, $\kappa < 1$, linear convergence
If $\beta = 1$, $\kappa = 0$, superlinear convergence
If $\beta = 1$, $\kappa = 0$, and at the same time
 $\beta > 1$, $\kappa = \infty$, superlinear convergence (special case)
If $\beta > 1$, $\kappa > 0$, superlinear convergence at rate β
If $\beta = 2$, $\kappa > 0$, quadratic convergence

- The rate of convergence does not have to be an integer.
- The term order of convergence is also used to describe the speed at which the error of discretization methods converges to zero as the discretization is refined, but this definition is different, and should not be confused with the rate of convergence of a sequence or an iterative method.

The bisection method

The bisection method for approximating a solution of $f(x) = 0$ is based on the fact that a continuous f function that changes sign in an interval $[L, R]$ must have at least one zero (more specifically, an odd number of roots) in that interval.

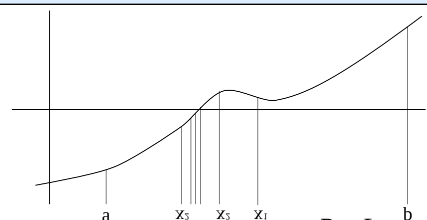
The bisection method is applicable if f is continuous, and if two points L and R can be found, such that $f(L) f(R) < 0$.

The bisection method approximates one of the zeros of f in $[L, R]$ by halving the interval at each iteration.

The bisection method
Let L and R be such that $sign(f(L)) \neq sign(f(R))$
for $k = 1$ to maxit
  $M = L + (R - L)/2$ /* $M = (L + R) / 2$ */
  if $|R - L| / 2 \leq \varepsilon$ or $f(M) = 0$, $x = M$, exit loop successfully, endif
  if $sign(f(L)) sign(f(M)) > 0$ then $L = M$
  else $R = M$
  endif
endfor

The bisection method

Note:
$sign(x) = 1$, if $x \geq 0$,
$sign(x) = -1$, if $x < 0$,



- The bisection method requires only one function evaluation per iteration.
- At the kth iteration bisection reaches a root of f within precision $\frac{R - L}{2^k}$, where $[L, R]$ is the initial interval the method is applied to.
- The bisection method always converges (whenever it is applicable).
- The bisection method converges linearly. This is slower than Newton's and slower than secant. However, bisection is useful for computing initial guesses for either Newton's or the secant method.
- If f has several roots in $[L, R]$, there is no guarantee as to which of the roots bisection converges to.
- The bisection method reaches a root within tolerance ε after at most $ceil(\log_2(\frac{b - a}{\varepsilon}))$ iterations.

Fixed-point (functional) iteration methods

Fixed-point iteration methods form a big class of root-finding methods, among which some of the most popular ones. Fixed-point iteration techniques are used to obtain the relation that computes the next approximation to the root x of a nonlinear equation.

Recall: A **fixed point** of a function g is an x such that $x = g(x)$.
Let $f(x) = 0$ be the nonlinear equation to be solved, and let x be a root.
Assume we construct a function g such that, when x is a root of $f(x) = 0$, then x is a fixed point of $g(x)$, and vice-versa. That is, g is constructed so that

$$f(x) = 0 \iff x = g(x) \tag{3}$$

Then a **fixed-point iteration** method for approximating the zero of f is:
guess $x^{(0)}$
for $k = 1$ to maxit
  $x^{(k)} = g(x^{(k-1)})$
  if stopping criterion is satisfied, $x = x^{(k)}$, exit loop successfully, endif
endfor

The above is a **one-step** iteration scheme.
Many convergence properties of the fixed-point iteration scheme are based on the properties of the function g.

Convergence of fixed-point iteration

Theorem 4b (extension of Theorem 4): **Convergence** of fixed-point iteration $x^{(k+1)} = g(x^{(k)})$.
If
− g is contraction in $I = [a, b]$ with constant λ,
− $g(x) \in I$, $\forall\, x \in I$ (i.e. $g(x)$ maps I to itself),
then
− there **exists** a **unique** fixed point x^* of g in I,
− $\forall x^{(0)} \in I$, the iteration scheme $x^{(k+1)} = g(x^{(k)})$ **converges** to x^*,
− $|x^{(k)} - x^*| \leq \lambda^k |x^{(0)} - x^*| \leq \lambda^k \max\{x^{(0)} - a,\ b - x^{(0)}\}$.

Proof based on Theorem 2, the contraction properties, and the definition of the fixed-point iteration scheme.

Convergence of fixed-point iteration

Theorem 5: **Convergence** of fixed-point iteration $x^{(k+1)} = g(x^{(k)})$.
If
− $g(x)$ has a fixed point x^*,
− $g(x)$ has continuous derivative in an open interval containing x^*,
− $|g'(x^*)| < 1$,
then
− there exists an open interval I that contains x^* (e.g. $I = (x^* - r^*, x^* + r^*)$, $r^* > 0$), such that, $\forall x^{(0)} \in I$, the iteration scheme $x^{(k+1)} = g(x^{(k)})$ **converges** to x^*.

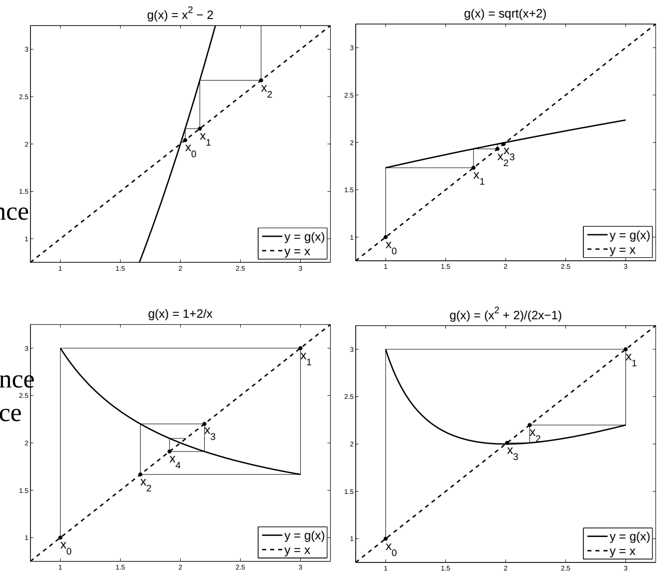
Proof based on the Mean Value Theorem.
Note: To pick r^*, we usually first pick $r_1 > 0$ and $r_2 > 0$, so that $|g'(x)| < 1$, $\forall x \in I = (x^* - r_1, x^* + r_2)$, then adjust the largest of r_1 and r_2 to the smallest. (i.e. if $r_1 > r_2$, set $r^* = r_2$, and if $r_1 < r_2$, set $r^* = r_1$).

Convergence of fixed-point iteration

For a given function f, there can be many functions g such that (3) holds, i.e. $f(x) = 0 \iff x = g(x)$.
Example:
Let $f(x) = x^2 - x - 2$. The following $g(x)$ satisfy (3):
(a) $g(x) = x^2 - 2$,
(b) $g(x) = \sqrt{x+2}$,
(c) $g(x) = 1 + 2/x$,
(d) $g(x) = (x^2 + 2)/(2x - 1)$.
However, the fixed-point iteration schemes arising from the above g's do not have the same convergence properties.

Convergence of fixed-point iteration

(a) divergence
(b) monotone convergence
(c) alternating convergence
(d) quadratic convergence



Rate of convergence of fixed-point iteration

Theorem 6:
If
− g has a fixed point α,
− $x^{(k+1)} = g(x^{(k)})$ converges to α,
− $g \in C^{\beta}$ near α,
− $g'(\alpha) = g''(\alpha) = \cdots = g^{(\beta-1)}(\alpha) = 0$ and $g^{(\beta)}(\alpha) \neq 0$,
then
− the rate of convergence of $x^{(k+1)} = g(x^{(k)})$ is β and
− the asymptotic error constant is $\frac{1}{\beta!}|g^{(\beta)}(\alpha)|$.

Note:
The above holds even if $\beta = 1$, in which case the last assumption reads
− $g'(\alpha) \neq 0$

The proof is based on Taylor's expansion and the definition of rate of convergence.

Newton’s method (Newton-Raphson method)

Newton’s method is the most well-known method for approximating a solution of $f(x) = 0$.

It is based on the idea of linearizing a nonlinear function, by replacing the function $f(x)$ by its tangent line at the current point of approximation $x^{(k)}$, assuming - of course - that the function is differentiable at $x^{(k)}$. The equation of the tangent of f at $x^{(k)}$ is

$$y = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) \tag{4}$$

Then approximate the root of $f(x) = 0$, i.e. the point where the graph of $f(x)$ hits the x-axis, by the point where the graph of the tangent hits the x-axis. We are, therefore, looking for the point x that makes $y = 0$ in (4). This will be the new approximation $x^{(k+1)}$ to the root of f. Thus

$$0 = f(x^{(k)}) + f'(x^{(k)})(x^{(k+1)} - x^{(k)}) \Rightarrow$$

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \tag{5}$$

Relation (5) gives **Newton’s iteration** for approximating the roots of a nonlinear equation $f(x) = 0$.

Newton’s method (Newton-Raphson method)

If we define

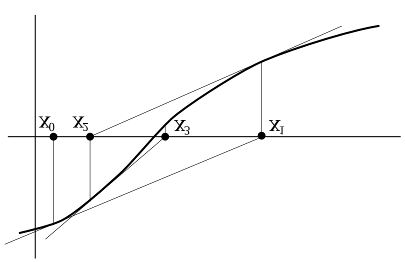
$$g(x) = x - \frac{f(x)}{f'(x)} \tag{6}$$

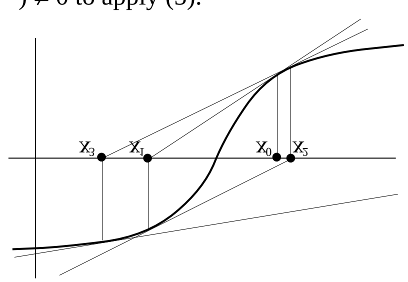
then Newton’s iteration becomes

$$x^{(k+1)} = g(x^{(k)})$$

i.e. it takes the form of a fixed-point iteration.

Note that f must be differentiable and $f'(x^{(k)}) \neq 0$ to apply (5).





Newton’s method (Newton-Raphson method)

Newton’s method

guess $x^{(0)}$

for $k = 1$ to maxit

$\quad x^{(k)} = x^{(k-1)} - \dfrac{f(x^{(k-1)})}{f'(x^{(k-1)})}$

$\quad$ if $|f(x^{(k)})| \le \varepsilon$, $x = x^{(k)}$, exit loop successfully, endif

endfor

- Newton’s method requires two function evaluations per iteration, one for f and one for f'.
- Newton’s method does not always converge. More specifically, it may converge when started at a certain initial guess, but may diverge if started at another initial guess. However, it always converges if f is twice differentiable and $x^{(0)}$ is chosen «close enough» to the root.
- When Newton’s method converges, it usually converges quadratically (rate 2), thus the number of correct digits of the root approximation approximately doubles at each iteration, thus a few (often 4-5) iterations suffice.

Newton’s method (Newton-Raphson method)

- Newton’s method converges slowly close to a multiple root. (It converges linearly with $\kappa = 1 - \dfrac{1}{m}$, where m the multiplicity of the root.) There exist modifications that improve convergence for this case.
- If f has several roots, there is no general guarantee as to which of the roots Newton’s converges to.

Modified Newton’s

Sometimes, the evaluation of the derivative is very time consuming (it may require many more flops than the evaluation of the function itself). A way to reduce the cost of a Newton iteration is to employ a previous derivative in the current iteration. That is, use

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

instead of (5). This gives rise to a *modified Newton’s method*. Modified Newton’s method (when it converges) converges much slower than Newton’s, thus it requires many more iterations. A compromise is to evaluate the derivative occasionally.

The secant method

The secant method for approximating a solution of $f(x) = 0$ can be derived by considering Newton's method and approximating the slope of the tangent of f at $x^{(k)}$ by the slope of the chord that subtends the graph of f at $x^{(k)}$ and $x^{(k-1)}$. Thus the method is based on linearizing a nonlinear function $f(x)$ by replacing $f(x)$ by the chord between $(x^{(k)}, f(x^{(k)})$ and $(x^{(k-1)}, f(x^{(k-1)})$. The slope of the chord is

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

Replacing $f'(x^{(k)})$ by the above slope in (5) we get

$$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} \tag{7}$$

Relation (7) gives the **secant iteration** for approximating the roots of a nonlinear equation $f(x) = 0$.

Note that we must have $f(x^{(k)}) \neq f(x^{(k-1)})$ to apply (7).

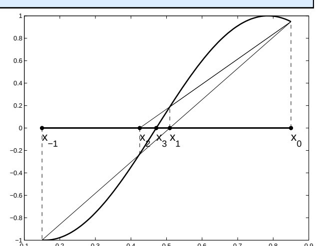
The secant iteration is a **two-step** scheme. Thus, the secant method requires two initial guesses. We usually make a guess $x^{(0)}$ and pick a $x^{(-1)}$ close to it, and such that $f(x^{(0)}) \neq f(x^{(-1)})$.

The secant method

The secant method
guess $x^{(0)}$ and $x^{(-1)}$
for $k = 1$ to maxit

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

if $|f(x^{(k)})| \leq \varepsilon$, $x = x^{(k)}$, exit loop successfully, endif
endfor



- The secant method requires only one function evaluation per iteration.
- The secant method does not always converge. It may converge when started at a certain initial guess, but may diverge if started at another initial guess. When it converges it does so a little slower than Newton's, but quite faster than bisection.
- When the secant method converges, it usually converges with rate $\beta = 1.618$ (the positive root of $x^2 - x - 1 = 0$, i.e. $\beta = (1 + \sqrt{5})/2$), thus the number of correct digits of the root approximation approximately increases by a factor of 1.6 at each iteration, thus a few (often 5-6) iterations suffice.
- The secant method method converges slowly close to a multiple root.
- If f has several roots, there is no general guarantee as to which of the roots secant converges to.

Newton's method for systems of nonlinear equations

Most methods for solving nonlinear equations in one variable cannot be extended in a straightforward way to the case of systems of nonlinear equations in several variables. However, Newton's method can.

Let $f: \mathbb{R}^n \to \mathbb{R}^n$ be a differentiable function (with respect to all variables), and let $\bar{f}(\bar{x}) = \bar{0}$ the system to be solved. The role of derivative in the case of one variable is played by the **Jacobian matrix** J in the case of several variables. The matrix J is defined by $(J(\bar{x}))_{ij} = \frac{\partial f_i}{\partial x_j}(\bar{x})$, where $\frac{\partial f_i}{\partial x_j}$ is the partial derivative of f_i with respect to x_j. Note that J is a $n \times n$ matrix, and that the role of division by $f'(x^{(k)})$ in the case of one variable is played by the multiplication by the inverse of J in the case of several variables. Thus, the new approximation in Newton's method for systems of nonlinear equations is given by

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - J^{-1}(\bar{x}^{(k)}) \bar{f}(\bar{x}^{(k)}). \tag{8}$$

Note that f must be differentiable at $\bar{x}^{(k)}$ and that $J(\bar{x}^{(k)})$ must be a non-singular matrix, in order to apply (8).

Also, since the elements of the inverse of J are not required explicitly, the multiplication of J^{-1} by the vector $\bar{f}(\bar{x}^{(k)})$ is computed by solving an $n \times n$ linear system.

Newton's method for systems of nonlinear equations

Newton's method for systems
guess $\bar{x}^{(0)}$
for $k = 1$ to maxit
solve $J(\bar{x}^{(k-1)})\bar{s}^{(k-1)} = -\bar{f}(\bar{x}^{(k-1)})$
$\bar{x}^{(k)} = \bar{x}^{(k-1)} + \bar{s}^{(k-1)}$
if $\|f(\bar{x}^{(k)})\| \leq \varepsilon$, then $\bar{x} = \bar{x}^{(k)}$, exit loop successfully, endif
endfor

Note: Often, the bar above x, f, s, etc, is omitted, as it should be clear from the context whether the quantities are scalars or vectors.

- Newton's method for systems requires the evaluation of a function of n variables (f) and the computation of the Jacobian matrix J at each iteration. The computation of J requires n^2 function evaluations ($\partial f_i / \partial x_j$), each of n variables.
  Furthermore, Newton's method for systems requires the solution of an $n \times n$ linear system at each iteration, something that costs about $\frac{n^3}{3}$ flops.

Newton's method for systems of nonlinear equations

- Newton's method for systems does not always converge. It may converge when started at a certain initial guess, but may diverge if started at another initial guess. However, it always converges if f is twice differentiable and $x^{(0)}$ is chosen «close enough» to the root.
- When Newton's method converges, it usually converges quadratically (rate 2), thus the number of correct digits of the root approximation approximately doubles at each iteration, thus a few (often 4-5) iterations suffice.
- Since the cost of each Newton iteration is high, several variants of the Newton method (*quasi-Newton* methods) have been developed aiming at reducing the cost, but preserving (or almost preserving) the high rate of convergence.
  There are two ways to reduce the cost of a Newton iteration: (a) avoid evaluating the Jacobian (n^2 partial derivatives), and (b) avoid solving a linear system.
  Some methods compute successive approximations to the Jacobian, and some other methods compute successive approximations to the inverse of the Jacobian.
  Broyden's method is one of the most well-known quasi-Newton methods, and can be considered an extension of secant to multiple dimensions. However, the study of quasi-Newton methods is out of the scope of this course.

Brief overview of numerical methods for solving nonlinear equations

Method / property	bisection	Newton	secant
guaranteed convergence	yes	no	no
(usual) convergence rate	1	2	1.618
no. of func. evals per it	1	2 (f, f')	1
requirements for application	continuity, change of sign	differentiability, $f'(x^{(k)}) \neq 0$	continuity, $f(x^{(k)}) \neq f(x^{(k-1)})$
number of initial guesses (past approximations) required	2	1	2
fixed-point iter. method	no	yes	no
extendable to nonlin. systems	no	Newton's	Broyden's
no. of func. evals per it		$n^2 + n$	n
flops per iteration		$O(n^3)$	$O(n^2)$

