Numerical analysis (6/7): Nonlinear (systems of) equations

University of Luxembourg

Philippe Marchner

Siemens Digital Industries Software, France

November 23th, 2022

Outline

Outline

Overview of the content

We will see methods to find solutions of nonlinear equations of the form

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

with $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$.
Most of the time, solutions are not known explicitly and we need numerical methods.
Finding zeros of a function has an important connection with optimization.

Objectives

- describe some of the most useful numerical methods
- study the convergence of these methods
- evaluate the efficiency i.e. the convergence speed and cost of the associated sequences
- adapt some methods to higher dimensional problems

Outline

Preliminary analysis

Problem description

- Sometimes we know how to explicitly solve some equations.
  For example
  $$x^2 - x - 1 = 0$$
  has two solutions: $\dfrac{1+\sqrt{5}}{2}$ and $\dfrac{1-\sqrt{5}}{2}$
- However, if one considers the equation
  $$\cos x = x,$$
  a mathematical theorem (which one ?) indicates that it has a unique solution between 0 and 1, but it cannot be explicitly written.
- Nevertheless, in scientific computing, an **approximation** of the solution will be sufficient with an error estimate if possible.

Preliminary analysis

Problem description

Let us consider the following 1D equation

$$f(x) = 0,\; x \in \mathbb{R}$$

where f is a real-valued function with one parameter.

- We assume that this equation admits (at least) one root r, such that $f(r) = 0$.
- The idea is to build a sequence (x_n) that converges towards r.
- Hence, the term x_n of the sequence will be an approximation of r, the accuracy depending on the choice of n.

Question

How to build the sequence (x_n) ?

Preliminary analysis

Problem description

Let us consider the following 1D equation

$$f(x) = 0, \; x \in \mathbb{R}$$

where f is a real-valued function with one parameter.

Before using a numerical method, it is better (if possible)

- to check that the equation has at least one solution
- to determine the number of roots
- to localize the roots i.e. to determine some intervals $[a_i, b_i]$ in which the considered equation has one and only one solution

Intermediate value theorem

To this end, we have

Intermediate value theorem - existence of roots

Let I be an interval in $\mathbb{R}$, f an application from I into $\mathbb{R}$, continuous on I. If there exist two elements a and b in I such that $a < b$ and $f(a)f(b) \leq 0$, then there exists $r \in [a, b]$ such that $f(r) = 0$.

Intermediate value theorem - root unicity

Let a and b two real numbers such that $a < b$ and f an application from $[a, b]$ into $\mathbb{R}$, continuous and strictly monotone on $[a, b]$. If $f(a)f(b) \leq 0$, then there exists a unique value $r \in [a, b]$ such that $f(r) = 0$.

Example

Solve on $\mathbb{R}$

$$x - 0.2\sin(x) - 0.5 = 0$$

Let f be the function defined on $\mathbb{R}$ by $f(x) = x - 0.2\sin(x) - 0.5$

The function f is continuous and differentiable on $\mathbb{R}$ and since we have for any x

$$f'(x) = 1 - 0.2\cos(x) > 0$$

this function is also strictly increasing on $\mathbb{R}$. In addition, since

$$\lim_{x\to-\infty} f(x) = -\infty \quad \text{and} \lim_{x\to+\infty} f(x) = +\infty$$

we deduce that f admits a unique root on $\mathbb{R}$.

More precisely, $f(0) = -0.5 < 0 \quad \text{and} \quad f(\pi) = \pi - 0.5 > 0$, f admits a unique root in $\mathbb{R}$ located between 0 and π.

Example

Solve in $\mathbb{R}$

$$\cos(x) = e^{-x}$$

Let f be the function defined on $\mathbb{R}$ by $f(x) = \cos(x) - e^{-x}$

The function f is continuous and differentiable on $\mathbb{R}$ and since we have for any x

$$f'(x) = -\sin(x) + e^{-x}.$$

Here it is difficult to study the sign of f' and deduce the variations of f, since we find a "similar" problem.

Example

Solve in $\mathbb{R}$

$$\cos(x) = e^{-x}$$

Let us now consider the function g defined on $\mathbb{R}$ by $g(x) = e^x \cos(x) - 1 = e^x f(x)$

The function g is continuous and differentiable in $\mathbb{R}$ and since we have for any x

$$g'(x) = e^x(\cos(x) - \sin(x)) = \sqrt{2}e^x \cos\left(x + \frac{\pi}{4}\right).$$

this function is also strictly monotone on the intervals $[\frac{\pi}{4} + k\pi, \frac{5\pi}{4} + k\pi]$, $k \in \mathbb{Z}$.

The study of the successive signs of $g(\frac{\pi}{4} + k\pi)$ allows to localize its roots.

Outline

Outline

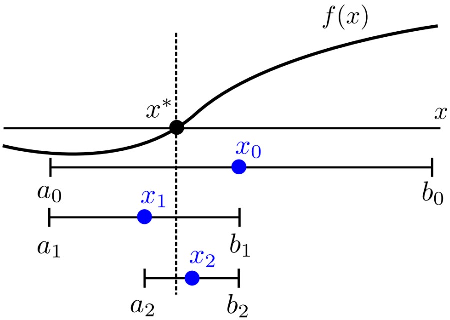
Bisection method

Principle of the bisection method

we start with an initial interval that contains a root and we build a sequence of intervals such that

- the root lies inside all the intervals
- the length of the intervals tends towards 0

One gets a converging process for localizing the roots by subdividing.



Bisection method

An interval $[a, b]$ is defined by a and b. To define the sequence of intervals, it is equivalent to fix the sequences (a_n) and (b_n) through a and b. Let $f : [a, b] \to \mathbb{R}$ be a continuous function such that $f(a) f(b) \leq 0$.

Bisection pseudo-code

$a_0 = a,\ b_0 = b$;
forall *n from* 0 *to* N **do**
  $m := \frac{(a_n + b_n)}{2}$;
  if $f(a) f(m) \leq 0$ **then**
    $a_{n+1} := a_n,\ b_{n+1} := m$;
  else
    $a_{n+1} := m,\ b_{n+1} := b_n$;
  end
end

Bisection algorithm

- The two sequences (a_n) and (b_n) satisfy by construction
  - $\forall\ n \in \mathbb{N}, \quad a_n \leq a_{n+1} \leq b_{n+1} \leq b_n$
  - $\forall\ n \in \mathbb{N}, \quad |a_n - b_n| = \frac{|a-b|}{2^n}$
  - $\forall\ n \in \mathbb{N}, \quad f(a_n)f(a) \geq 0, \quad f(b_n)f(a) \leq 0$
- consequently, the two sequences (a_n) and (b_n) are adjacent and they converge towards the same limit $r \in [a, b]$.
- since f is continuous on $[a, b]$, the sequences $(f(a_n))$ and $(f(b_n))$ converge towards $f(r)$.
- according to the sign of $f(a)$, they moreover satisfy, for any $n \in \mathbb{N}$

$$(f(a_n) \leq 0 \text{ and } f(b_n) \geq 0) \quad \text{or} \quad (f(a_n) \geq 0 \text{ and } f(b_n) \leq 0)\,.$$

- in both cases, one gets at the limit that $f(r) \leq 0$ and $f(r) \geq 0$, which implies that $f(r) = 0$.

Bisection method

Remarks

- When f is continuous on $[a, b]$ and $f(a)f(b) \leq 0$, this method converges.
- Only one evaluation per iteration of the function f is required
- Since we have

$$a_n \leq r \leq b_n, \quad \forall n \geq 0$$

  we can choose indifferently a_N or b_N as the approximation of the root, a_N being then a lower approximate value and b_N an upper approximate estimate
- we then have the following accuracy

$$|a_N - r| \leq |a_N - b_N| = \frac{|a - b|}{2^N}$$

Bisection algorithm

Remarks

- according to the expected precision ϵ , we can a priori determine the stopping index N such that

$$|a_N - b_N| = \frac{|a-b|}{2^N} < \epsilon$$

$$N \geq \text{floor}\left(\frac{\ln(|a-b|) - \ln(\epsilon)}{\ln(2)}\right) + 1$$

- This method converges even if the function f has a few roots in the initial interval.

Outline

Fixed-point methods

Principle

Searching for a solution to the equation $f(x) = 0$ can be seen as searching the solution to

$$g(x) = x$$

for example by setting

- $g(x) = x - f(x)$
- $g(x) = x - \dfrac{f(x)}{\alpha}$, with $\alpha \neq 0$
- $g(x) = x - \dfrac{f(x)}{\alpha(x)}$, with $\forall\, x \in I,\ \alpha(x) \neq 0$

Therefore, the root-finding for f amounts to searching for a fixed-point of g.

Fixed-point methods

We can then use the following algorithm

x_0 given;
forall *n from* 0 *to* ... **do**
| $x_{n+1} = g(x_n)$
end

Indeed, let us recall the following analysis result

Theorem

Let I be a closed and stable interval by g, $\xi \in I$ and (x_n) the sequence defined by the relations $x_0 = \xi$ and $x_{n+1} = g(x_n), \forall n \in \mathbb{N}$.
In addition, we assume that f is continuous on I.
If the sequence (x_n) converges, its limit is a fixed-point of g in I

A fixed-point of g is hence a root of f.

Fixed-point methods

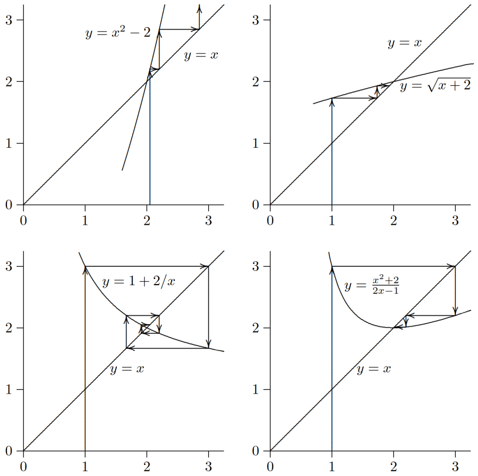


Figure: Fixed-point iterations for some nonlinear functions. From *M. T. Heath, Scientific computing: an introductory survey (2018)*

Fixed-point theorem

In particular, we have the

Fixed-point theorem

Let us assume that I is closed, g is a contraction mapping on I with ratio $k \in [0, 1)$ and I is stable by g. Then

- g admits on I a unique fixed-point $r \in I$.
- for any initial guess $\xi \in I$, the sequence (x_n), defined by $x_0 = \xi$ and the recursive relation $x_{n+1} = g(x_n)$, converges towards the fixed-point r.
- we have the following estimates

$$\forall\ n \in \mathbb{N}, \quad |x_n - r| \leq \frac{k^n}{1-k}|x_1 - x_0|$$

$$\forall\ n \in \mathbb{N}, \quad |x_n - r| \leq \frac{k}{1-k}|x_n - x_{n-1}|$$

Fixed-point methods

Example

Let f be the function defined on $I = [0, +\infty[$ by

$$f(x) = x - e^{-(1+x)}$$

This function has a unique root r between 0 and 1

Fixed-point methods

Example

Let f be the function defined on $I = [0, +\infty[$ by

$$f(x) = x - e^{-(1+x)}$$

Algorithm 1: Let g be the function defined on I by

$$g(x) = e^{-(1+x)}$$

We easily check that I is stable by g, that g is a contraction mapping on I and that I is closed

Therefore, for any $\xi \in I$, the sequence (x_n) defined by $x_0 = \xi$ and $x_{n+1} = g(x_n)$, converges towards the unique fixed-point of g which is also the root of f

Fixed-point methods

Example

Let f be the function defined on $I = [0, +\infty[$ by

$$f(x) = x - e^{-(1+x)}$$

Algorithm 2: Let us now consider h as the function defined on I by

$$h(x) = x^2 e^{(1+x)}$$

The function h admits two fixed points: 0 and r in I

Let us study, for any $\xi \in I$, the asymptotic behavior of the sequence (x_n) defined by $x_0 = \xi$ and $x_{n+1} = h(x_n)$

Fixed-point methods

Example

Let f be the function on $I = [0, +\infty[$ defined by

$$f(x) = x - e^{-(1+x)}$$

For the fixed-point function h, we show that:

- if $\xi \in [0, r[$ then the sequence (x_n) converges towards 0
- if $\xi = r$ then the sequence (x_n) is constant
- if $\xi \in]r, +\infty[$ the sequence (x_n) diverges towards $+\infty$

This second algorithm is then unadapted to get an approximation of the root r !

Fixed-point methods

Stability of the fixed-point

Let g be a map from I into I that admits a fixed-point $r \in I$.

- We say that r is an attractive or stable fixed-point if there exists $\eta > 0$ such that any sequence (x_n) defined by $x_0 \in]r - \eta, r + \eta[\cap I$, the recursive relation $x_{n+1} = g(x_n)$ converges towards r.
- We say that r is a repulsive or unstable fixed-point when for any sequence (x_n) defined by the recursive relation $x_{n+1} = g(x_n)$, there exists $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$, the sequence (x_n) moves away from r.

Fixed-point methods

Theorem

Let g be a map from I into I that admits a fixed-point $r \in I$. We suppose that g is differentiable at r.

- if $|g'(r)| < 1$, then r is an attractive fixed-point.
- if $|g'(r)| > 1$, then r is a repulsive fixed-point.
- if $|g'(r)| = 1$, then both cases can arise

Remark

in the previous example, we have $|h'(r)| > 1$. The fixed point is repulsive and cannot be attained.

Outline

Convergence speed

Let (x_n) be a sequence that converges towards a number r.

- we say that the convergence speed is linear, if there exists C, $0 < C < 1$ such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - r|}{|x_n - r|} = C. \tag{1}$$

- the number C is called the convergence speed.
- we say that the convergence is at least linear, if there exists C, $0 < C < 1$ such that

$$|x_{n+1} - r| \leq C\,|x_n - r| \quad \forall n \geq 0$$

Convergence speed

- we say that the convergence is of order q, if there exists $q > 1$, $C > 0$ such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - r|}{|x_n - r|^q} = C$$

- we say that the convergence is at least of order q, if there exists $q > 1$, $C > 0$ such that

$$|x_{n+1} - r| \leq C\,|x_n - r|^q \quad \forall n \geq 0$$

- a second-order convergence is also called quadratic and a convergence of order 3 is said to be cubic.

Convergence speed

Practical meaning

Let us set for any $n \in \mathbb{N}$, $e_n = |x_n - r|$. The number e_n represents the error when we approximate r by x_n.

- if the convergence speed is linear then there exists $0 < C < 1$ such that $e_{n+1} \sim Ce_n$
- this means that asymptotically the error is reduced by a factor C at each iteration.
- the smaller will be the ratio C, the faster will be the convergence of the sequence

Convergence speed

Practical meaning

- if the convergence is of order $q > 1$, then there exists $C > 0$ such that $e_{n+1} \sim C e_n^q$.
- let us then set for all $n \in \mathbb{N}$, $\lambda_n = -\log_{10} e_n$.
- the number λ_n is a "measure" of the number of exact decimals of x_n.
- indeed if $e_n = 10^{-5}$ then $\lambda_n = 5$, if $e_n = 10^{-10}$ then $\lambda_n = 10$, etc...
- we have

$$\lambda_{n+1} \sim q\lambda_n.$$

  which means that asymptotically the number x_{n+1} has q times more "exact decimals" than x_n.
- the larger will be the convergence order, the faster will be the convergence of the sequence

Application to the fixed-point method

Order of convergence of a fixed-point method

Let (x_n) be a sequence defined by the recursive relation $x_{n+1} = g(x_n)$ and let r be a fixed-point of g.
If g is a three times differentiable function in I, then from the Taylor-Young formula, we have for any $n \in \mathbb{N}$

$$\begin{aligned} x_{n+1} - r &= g(x_n) - g(r) \\ &= g'(r)(x_n - r) + \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(r)}{6}(x_n - r)^3 + o((x_n - r)^3) \end{aligned}$$

that is

$$e_{n+1} = g'(r)e_n + \frac{g''(r)}{2}e_n^2 + \frac{g'''(r)}{6}e_n^3 + o(e_n^3)$$

Application to the fixed-point method

Order of convergence of a fixed-point method

$$e_{n+1} = g'(r)e_n + \frac{g''(r)}{2}e_n^2 + \frac{g'''(r)}{6}e_n^3 + o(e_n^3)$$

Several cases then appear

- if $g'(r) \neq 0$ and $|g'(r)| < 1$, then $e_{n+1} \sim Ce_n$ with $C = |g'(r)|$.
  The sequence (x_n) converges linearly to r.
- if $g'(r) = 0$ and $g''(r) \neq 0$, then $e_{n+1} \sim Ce_n^2$ with $C = \frac{|g''(r)|}{2}$.
  The sequence (x_n) is convergent of order 2.
- if $g'(r) = g''(r) = 0$ and $g'''(r) \neq 0$, then $e_{n+1} \sim Ce_n^3$ with $C = \frac{|g'''(r)|}{6}$.
  The sequence (x_n) is converging of order 3.
- and so on, if we assume more smoothness on g.

Outline

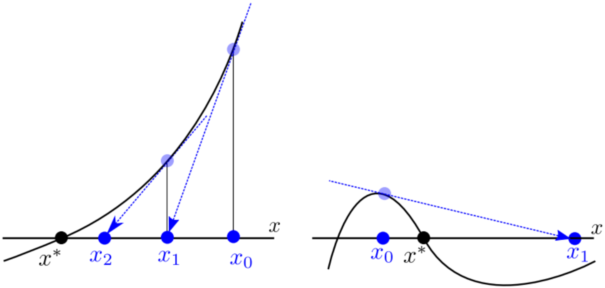
The Newton method

Description of the method

- if f is an affine function

$$f(x) = ax + b \quad (a \neq 0)$$

  the root is $r = -b/a$.
- the idea is to substitute f by an affine approximation $\rightarrow$ we can use its tangent.



The Newton method

Description of the method

Let us assume that f is a function defined on an interval I, differentiable on I and such that it has a root r in I

- let x_0 be a point I close enough to the root r
- we then have

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + o(x - x_0) \\ &= f_{x_0}(x) + o(x - x_0) \end{aligned}$$

with $f_{x_0}(x) = f'(x_0)(x - x_0) + f(x_0)$

The Newton method

Description of the method

- the affine function f_{x_0} admits a root x_1 if and only if $f'(x_0) \neq 0$, and in this case

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

- we can expect then that x_1 will be closer to the root r than x_0 i.e. that x_1 is a better estimate of r
- we can then iterate with x_1 instead of x_0 and so on...
- we expect to improve the approximation of the root r through successive iterations.

The Newton method

Newton's algorithm

x_0 given;
forall n *from* 0 *to* ... **do**

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

end

Remarks

- to have a well-defined sequence (x_n), we must have $f'(x_n) \neq 0$, $\forall n \in \mathbb{N}$.
- at each iteration, we have to evaluate two functions: computation of $f(x_n)$ and computation of $f'(x_n)$
- the Newton method is a fixed-point method with $g(x) = x - \frac{f(x)}{f'(x)}$

Convergence of the Newton method

Theorem

Let f be an application from I into I and $r \in I$ a root of the function f. We assume that f is twice differentiable in a neighborhood of r and that $f'(r) \neq 0$.
Then, there exists $\eta > 0$ such that for any $x_0 \in]r - \eta, r + \eta[\cap I$ the Newton method generates a well-defined sequence (x_n) which converges at least quadratically towards r.

Indeed

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

and so $g'(r) = 0$

Convergence of the Newton method

Remarks

- this result indicates that if x_0 is close enough to r (and if $f'(r) \neq 0$) then the method converges
- when there is convergence, it is fast (at least of order 2)
- if x_0 is not close enough to r, then divergence may occur
- in practice, there is generally no way to know if x_0 is close enough to r
- if the derivative does not exist or is discontinuous at the root, Newton's method may fail

Newton method

Example

Example : $x^2 = a$ Let $a > 0$

- we search for an approximation of $\sqrt{a}$
- here $f(x) = x^2 - a$ and the Newton algorithm writes

$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$$

- it can be easily shown that for any $x_0 > 0$ this sequence converges towards $\sqrt{a}$

Newton method

Example

Example : $x^2 = a$ for $a = 2$ and $x_0 = 1$ ones gets

$$\begin{aligned} x_0 &= 1 \\ x_1 &= \frac{3}{2} = 1.5 \\ x_2 &= \frac{17}{12} = 1.41666666666666\ldots \\ x_3 &= \frac{577}{408} = 1.41421568627450\ldots \\ x_4 &= \frac{665857}{470832} = 1.41421356237468\ldots \end{aligned}$$

Newton method

Example : $x^2 = a$

for $a = 2$ and $x_0 = 1$ one gets

$$\begin{aligned} x_0 &= 1 \\ x_1 &= \frac{3}{2} = 1.5 \\ x_2 &= \frac{17}{12} = 1.41666666666666\ldots \\ x_3 &= \frac{577}{408} = 1.41421568627450\ldots \\ x_4 &= \frac{665857}{470832} = 1.41421356237468\ldots \end{aligned}$$

let us remind us that

$$\sqrt{2} = 1.414213562373095\ldots$$

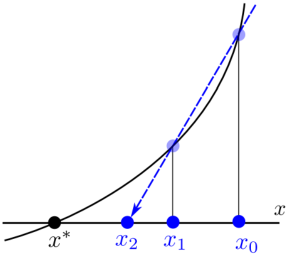
Outline

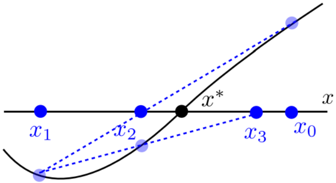
Another remark on Newton method

One Newton iteration $x_{n+1} = x_n - f(x_n)/f'(x_n)$ requires the evaluation of two functions: $f(x_n)$ and $f'(x_n)$.

- the derivative f must be known and we must be able to implement its evaluation f'
- we could also use

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





The secant method

A new method

Hence, one gets a close form method for evaluating f' :

x_0 given;
forall n *from* 0 *to* ... **do**

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

end

Remarks

- this method is well-defined if at each iteration $f(x_n + h_n) - f(x_n) \neq 0$
- the numerical step h_n can be different at each iteration
- at each step, we always have two evaluations: computation of $f(x_n)$ and $f(x_n + h_n)$

The secant method

to avoid this double evaluation, one can set

$$h_n = x_{n-1} - x_n \quad \forall n \geq 0$$

Indeed, if (x_n) converges, then (h_n) converges towards 0 and at each iteration we only have one evaluation: computation of $f(x_n)$ (if the algorithm is correctly written!)

forall *n from* 0 *to* ... **do**

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

end

the resulting algorithm is called the secant method

The secant method

Analysis of the method

Let us assume that f is a function defined on an interval I and that it has a root r in I

- let x_0 and x_1 be two points in I close enough to the root r
- we substitute in a neighborhood of x_1 the function f by the line passing through the points $(x_1, f(x_1))$ and $(x_0, f(x_0))$ of equation

$$f_{x_1}(x) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x - x_1) + f(x_1)$$

The secant method

Analysis of the method

- the affine function f_{x_1} admits a root x_2 if and only if $f(x_1) - f(x_0) \neq 0$, and in this case

$$x_2 = x_1 - f(x_1)\left(\frac{x_1 - x_0}{f(x_1) - f(x_0)}\right)$$

- we can expect that x_2 is closer to the root r than both x_0 and x_1
- we can then iterate with x_2 and x_1 and so on...
- we expect to improve the approximation of the root r by successive approximations.

The secant method

The secant method algorithm

x_0 x_1 given;
forall *n from* 0 *to* ... **do**

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

end

Remarks

- to get a well-defined sequence (x_n), we must have that $f(x_n) \neq f(x_{n-1})$ for all $n \in \mathbb{N}$.
- at each iteration, we have one evaluation: computation of $f(x_n)$
- the convergence analysis is similar to Newton's method,

Convergence of the secant method

Theorem

Let f be a map from I in I and $r \in I$ a root of the function f. We assume that f is twice continuously differentiable in a neighborhood of r and that $f'(r) \neq 0$.
Then, if (x_0, x_1) are sufficiently close to r the secant method generates a sequence (x_n) which is well-defined and converging towards r.
The error satisfies

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

In this case, the convergence is at least of order $\frac{1+\sqrt{5}}{2} = 1.618...$

Outline

Comparison between the algorithms

Bisection

- the method is converging
- only one evaluation at each iteration
- convergence speed is linear and therefore slow

Newton

- fast convergence when it converges
- not to sensitive to round-off errors if $f'(r)$ is not too small
- can diverge if the initial guess is not correctly chosen
- requires the evaluation of the derivative
- two evaluations at each iteration

Secant

- relatively fast convergence when the convergence occurs
- requires one evaluation of the function at each iteration
- can diverge if the initial guess is not correctly calibrated

Example

Resolution of $x - 0.2 \sin x - 0.5 = 0$ with the four algorithms

	Bisection $x_{-1} = 0.5$ $x_0 = 1.0$	Secant $x_{-1} = 0.5$ $x_0 = 1.0$	Newton $x_0 = 1$	Fixed-point $x_0 = 1$ $x = 0.2 \sin x + 0.5$
1	$0,75$	$0,5$	$0,5$	$0,50$
2	$0,625$	$0,61212248$	$0,61629718$	$0,595885$
3	$0,5625$	$0,61549349$	$0,61546820$	$0,612248$
4	$0,59375$	$0,61546816$	$0,61546816$	$0,614941$
5	$0,609375$			$0,61538219$
6	$0,6171875$			$0,61545412$
7	$0,6132812$			$0,61546587$
8	$0,6152343$			$0,61546779$
9	$0,6162109$			$0,61546810$
10	$0,6157226$			$0,61546815$
11	$0,6154785$			
12	$0,6153564$			
13	$0,6154174$			
14	$0,6154479$			
15	$0,6154532$			
16	$0,61547088$			
17	$0,61546707$			
18	$0,61546897$			
19	$0,615468025$			
20	$0,615468502$			

Outline

Convergence acceleration

Principle

Being given a sequence (x_n) that converges to r, accelerating the convergence consists in replacing the initial sequence by a sequence (y_n) that converges faster than (x_n) towards r, i.e; satisfying

$$\lim_{n \to +\infty} \frac{y_n - r}{x_n - r} = 0.$$

Example

if (x_n) converges linearly then (y_n) will converge faster or to a higher order.

Relaxation method

Let us consider a fixed-point method

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

that slowly converges or diverges

The equation that we are looking for $x = g(x)$ can also be written for any $\alpha \neq -1$

$$x + \alpha x = g(x) + \alpha x$$

or

$$x = \frac{g(x) + \alpha x}{1 + \alpha} = G(x)$$

We can then think of using a fixed-point

$$y_{n+1} = G(y_n)$$

Relaxation method

From the previous results, this method will converge as soon as y_0 is close to the fixed-point r and when

$$|G'(r)| = \left| \frac{g'(r) + \alpha}{\alpha + 1} \right| < 1$$

The convergence will be better when $|G'(r)|$ is small

Since we are free to choose the relaxation parameter α, the idea is to take it as close as possible to $-g'(r)$!

Aitken acceleration method

Hypothesis

Let (x_n) be a sequence converging to r and such that

$$x_{n+1} - r = k\,(x_n - r) \text{ where } 0 < k < 1$$

We have

$$\begin{aligned} x_{n+1} - r &= k\,(x_n - r)\,, \\ x_{n+2} - r &= k\,(x_{n+1} - r)\,. \end{aligned}$$

and by difference one gets

$$x_{n+2} - x_{n+1} = k\,(x_{n+1} - x_n)\,.$$

Aitken acceleration method

The principle

Let (x_n) be a sequence converging to r and such that

$$x_{n+1} - r = k\,(x_n - r) \text{ where } 0 < k < 1$$

By reporting then in the first equation written as

$$r = x_n + \frac{x_{n+1} - x_n}{1 - k}$$

we have

$$r = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} + x_n - 2x_{n+1}}.$$

three consecutive terms of the sequence are then enough to get r !

Aitken acceleration method

The principle

But the hypothesis is very strong and unrealistic.

The idea of Aitken is to generalize this remark to sequences that converge linearly that is

$$x_{n+1} - r = k_n \,(x_n - r) \quad \text{with} \lim_{n\to\infty} k_n = k \in [0, 1[\,.$$

For n large, k_n is almost constant, and so the number

$$y_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} + x_n - 2x_{n+1}}$$

should be close to r.

Aitken acceleration method

Theorem

Let (x_n) be a sequence that converges linearly to r.Then the sequence (y_n) defined by

$$y_n := x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} + x_n - 2x_{n+1}}$$

satisfies

$$\lim_{n\to\infty} \frac{y_n - r}{x_n - r} = 0.$$

The Aitken process allows to accelerate the convergence of a sequence which is linearly converging.

Aitken acceleration method

Remark:

For the computations, we will use the equivalent expression

$$y_n = x_{n+1} + \frac{1}{\frac{1}{x_{n+2}-x_{n+1}} - \frac{1}{x_{n+1}-x_n}}$$

this one having a better behavior regarding the round-off errors due to the use of a computer.

Steffensen algorithm

The principle

Let (x_n) be defined by:

$$x_{n+1} = g(x_n) \ \forall n \geq 0$$

and let us assume that (x_n) converges at least linearly to r.

This convergence can be improved bu using the Aitken method.

The idea is to use y_n (which is *a priori* closer to the limit r than x_n) instead of x_n in the Aitken algorithm to expect a double acceleration...

Steffensen algorithm

One gets the algorithm

x_0 given;
forall *n from* 0 *to* ... **do**

$$y_n := g(x_n);$$
$$z_n := g(y_n);$$
$$x_{n+1} := x_n - \frac{(y_n - x_n)^2}{z_n - 2y_n + x_n}$$

end

Steffensen algorithm

Remarks

- this algorithm is a new fixed-point algorithm

$$x_{n+1} = G\left(x_n\right) \text{ with } G\left(x\right) = x - \frac{\left(g\left(x\right) - x\right)^2}{g\left(g\left(x\right)\right) - 2g\left(x\right) + x}.$$

- we show that if $g'\left(r\right) \neq 0$, then $G'(r) = 0$. The algorithm associated with G converges then quadratically.
- hence, compared with the algorithm associated to g
  - we accelerate the convergence when it is converging
  - we have a converging process, even if $\left|g'\left(r\right)\right| \geq 1$.
- it must be noticed that if the algorithm for G converges faster that the one for g, each iteration needs two function evaluations: we have to pay the price !

Outline

Problem description

Multivariate function

Let us consider the equation

$$F(X) = 0$$

where $F : \mathbb{R}^N \mapsto \mathbb{R}^N$ or in terms of scalar equations

$$\begin{cases} f_1(x_1, x_2, \dots, x_N) & = 0 \\ f_2(x_1, x_2, \dots, x_N) & = 0 \\ \vdots & \vdots \\ f_N(x_1, x_2, \dots, x_N) & = 0 \end{cases}$$

The Newton-Raphson method

- the Newton-Raphson method is a generalization to higher-dimensional problems of the one-dimensional Newton method

$$x_{n+1} = x_n - (f'(x_n))^{-1} f(x_n)$$

- it involves the Jacobian matrix of F:

$$F'(X_n) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix}$$

  all the derivatives being evaluated at point X_n.
  The Newton-Raphson method formally writes down

$$X_{n+1} = X_n - [F'(X_n)]^{-1} F(X_n)$$

Taylor Series for vector functions

Theorem

Let $X = (x_1, x_2, \ldots, x_n)^T$, $F = (f_1, f_2, \ldots, f_m)^T$, and assume that $F(X)$ has bounded derivatives up to order at least two. Then for a direction vector $P = (p_1, p_2, \ldots, p_n)^T$, the Taylor expansion for each function f_i in each coordinate x_j yields

$$F(X+P) = F(X) + F'(X)P + \mathcal{O}\left(\|P\|^2\right),$$

where $F'(X)$ is the Jacobian matrix of first derivatives of F at X. Thus we have

$$f_i(X+P) = f_i(X) + \sum_{i=1}^{n} \frac{\partial f_i}{\partial x_j} p_j + \mathcal{O}\left(\|P\|^2\right), \quad i = 1, \ldots, m$$

For a small $P = R - X_n$, we have $0 = F(X_n + P) \approx F(X_n) + F'(X_n)(R - X_n)$.
We then define δ_n such as $F(X_n) + F'(X_n)\delta_n = 0$

The Newton-Raphson method

In a practical computation, we do not explicitly compute the inverse of the Jacobian matrix which would be too expensive. We prefer to write the algorithm under the following form

X_0 given;
forall *n from* 0 *to* ... **do**
    Solve the linear system $F'(X_n)\delta_n = -F(X_n)$;
    $X_{n+1} = X_n + \delta_n$;
end

The Newton-Raphson method

Remarks :

- the choice of the initial guess is crucial and the risk that the algorithm diverges truly exists.
- the convergence is second-order and therefore is really fast (when it converges!)
- the Newton-Raphson method is expensive since at each iteration one must
  - evaluate $N^2 + N$ functions (the N^2 partial derivatives of the Jacobian matrix, plus the N coordinates functions)
  - solve $N \times N$ the linear system (with a dense matrix!)

The Broyden method

Principle

- in the Newton-Raphson method, the computation of the Jacobian matrix is highly expensive
- we will then only determine an approximate value B_n at each iteration
- we have seen that the secant method could be deduced from the Newton method by approximating

$$f'(x_n) \quad \text{by} \quad \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

- in higher dimension N, we will just force the sequence of matrices (B_n) to verify the same relation

$$B_n(X_n - X_{n-1}) = F(X_n) - F(X_{n-1})$$

The Broyden method

Remarks:

- this relation does not allow us to uniquely define the matrix B_n (n equations for n^2 unknowns)
- it only imposes its value in one direction
- Broyden proposed to update B_n to B_{n+1} by simply adding a rank-one matrix

$$B_{n+1} = B_n + \frac{(\delta F_n - B_n \delta X_n)(\delta X_n)^T}{\|\delta X_n\|^2}$$

  where we introduced $\delta X_n = X_{n+1} - X_n$ and $\delta F_n = F(X_{n+1}) - F(X_n)$.
- we immediately verify that the sequence of defined matrices (B_n) then satisfy the relation.

The Broyden method

The Broyden algorithm

X_0 and B_0 given;
forall *n from* 0 *to* . . . **do**

  Solve the linear system $B_n \delta_n = -F(X_n)$;
  $X_{n+1} = X_n + \delta_n$;
  $\delta F_n = F(X_{n+1}) - F(X_n)$;
  $$B_{n+1} = B_n + \frac{(\delta F_n - B_n \delta_n)(\delta_n)^T}{\|\delta_n\|^2};$$

end

The Broyden method

Remarks:

- we can take the initial matrix as $B_0 = Id$; after a certain time, the matrix becomes a suitable approximation of the Jacobian matrix.
- it can be proved that in general and as for the secant method, the convergence is superlinear.
- The sequence of matrices (B_n) does not necessarily converge towards the Jacobian of F.

Summary

We have seen a few methods to find roots of 1D non-linear equations.

- the fixed-point theory is a fundamental concept to develop algorithms,
- they are methods to accelerate convergence (Aitken-acceleration)
- built-in methods combine the bisection, Newton and secant methods

In two or more dimensions the situtation is more complicated

- Newton method can still be used, but is very costly
- Cheaper methods can be devised by approximating the Jacobian
- Root-finding is strongly linked to optimization: zeros of the Jacobian help to detect extrema of functions