Lecture 3: Single nonlinear equations

COMP5930M Scientific Computation

Today

Standard notation

Given a scalar function $F(x) : \mathbb{R} \to \mathbb{R}$, find a point $x^* \in \mathbb{R}$ s.t.

$$F(x^*) \ = \ 0.$$

Although not all our problems are immediately viewed in this form we can always rewrite them in this way.

Commonly termed the *zero-finding problem*

The basic approach

- Since we do not have a direct method of solution we use an iterative method
- A solution algorithm generates a sequence of values x_n

$$x_0, x_1, x_2, \ldots$$

  from a given initial point x_0

- Further details of the process are specific to each algorithm

Convergence criteria

Possible conditions to satisfy:

- $|x_{n+1} - x^*| < |x_n - x^*|$
  ie. we are getting closer to the root at each step
- $|F(x_{n+1})| < |F(x_n)|$
  ie. the function $F(x)$ is reduced at each step

These criteria are distinct and one does not imply the other.

Different algorithms may satisfy one of these,
rarely both, and often neither

Convergence rate

Assume the sequence $x_0, x_1, \ldots, x_n$ converges to x^*. We say the sequence **converges linearly**, if there exists $0 < \alpha < 1$ and

$$\lim_{n\to\infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|} = \alpha.$$

Here α is the **rate of convergence**, i.e. the error is (eventually) reduced by a constant factor of α after each iteration.
If $\alpha = 1$ the sequence converges **sublinearly**.

Convergence rate

Assume the sequence $x_0, x_1, \ldots, x_n$ converges to x^*. We say the sequence **converges linearly**, if there exists $0 < \alpha < 1$ and

$$\lim_{n\to\infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|} = \alpha.$$

Here α is the **rate of convergence**, i.e. the error is (eventually) reduced by a constant factor of α after each iteration.

If $\alpha = 1$ the sequence converges **sublinearly**.

Sequence **converges superlinearly**, if for some $q > 1$ and $\alpha > 0$

$$\lim_{n\to\infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|^q} = \alpha.$$

If $q = 2$, we say it **converges quadratically**.

Convergence rate

Assume the sequence $x_0, x_1, \ldots, x_n$ converges to x^*. We say the sequence **converges linearly**, if there exists $0 < \alpha < 1$ and

$$\lim_{n \to \infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|} = \alpha.$$

Here α is the **rate of convergence**, i.e. the error is (eventually) reduced by a constant factor of α after each iteration.
If $\alpha = 1$ the sequence converges **sublinearly**.

Sequence **converges superlinearly**, if for some $q > 1$ and $\alpha > 0$

$$\lim_{n \to \infty} \frac{|x^* - x_{n+1}|}{|x^* - x_n|^q} = \alpha.$$

If $q = 2$, we say it **converges quadratically**.

Examples of orders and rates of convergence

Define $\varepsilon_n = |x^* - x_n|$ as the error of the nth iteration step.

Ex.1: If $\varepsilon_n = 1/n$, then

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^q} = \frac{n^q}{n+1} \to \begin{cases} 0, \text{ if } q < 1 \\ 1, \text{ if } q = 1 \\ \infty, \text{ if } q > 1 \end{cases}$$

so that the convergence is **sublinear** ($q = 1$, $\alpha = 1$)

Examples of orders and rates of convergence

Define $\varepsilon_n = |x^* - x_n|$ as the error of the nth iteration step.

Ex.2: If $\varepsilon_n = 1/2^n$, then

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^q} = \frac{2^{nq}}{2^{n+1}} = \frac{(2^n)^{q-1}}{2} \to \begin{cases} 0, \text{ if } q < 1 \\ 1/2, \text{ if } q = 1 \\ \infty, \text{ if } q > 1 \end{cases}$$

so that the convergence is **linear** ($q = 1$, $\alpha = 1/2$)

Examples of orders and rates of convergers

Define $\varepsilon_n = |x^* - x_n|$ as the error of the nth iteration step.

Ex.3: If $\varepsilon_n = 1/2^{2^n}$, then

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^q} = \frac{(2^{2^n})^q}{2^{2^{n+1}}} = \frac{(2^{2^n})^q}{(2^{2^n})^2} \to \begin{cases} 0, \text{ if } q < 2 \\ 1, \text{ if } q = 2 \\ \infty, \text{ if } q > 2 \end{cases}$$

so that the convergence is **quadratic** $(q = 2)$

The Bisection Method

- Assume the function $F(x)$ is continuous
- Assume we know two points x_L and x_R, such that

$$F(x_L)\ F(x_R)\ \leq\ 0$$

called the bracket condition for the bracket $[x_L, x_R]$

This implies that there is a solution $x^* \in [x_L, x_R]$, since the function changes sign over that interval
(due to the Intermediate Value Theorem).

The algorithm of bisection method

At iteration n:

- Consider the point $x_C^n = (x_L^n + x_R^n)/2$ and find $F(x_C^n)$.
  - If $F(x_L^n)\,F(x_C^n) \leq 0$ then $x^* \in [x_L^n, x_C^n]$.
  - If $F(x_C^n)\,F(x_R^n) < 0$ then $x^* \in [x_C^n, x_R^n]$.
- Select the new interval (also termed the **bracket**) $[x_L^{n+1}, x_R^{n+1}]$ as the subinterval containing x^*.
  - Note that in one step the bracket containing x^* has been halved.
- Repeat this process until $x_R^n - x_L^n < TOL$, where TOL is a user-supplied value,

  i.e. repeat until the bracket is sufficiently small.

Convergence rate of bisection method

The initial error is $\varepsilon_0 := |x^* - x_C^0| \leq (x_R^0 - x_L^0)/2$.

At each iteration, we half the interval so that the error is halved:

$$\varepsilon_n := |x^* - x_C^n| \leq \frac{x_R^n - x_L^n}{2} = \ldots = \frac{x_R^0 - x_L^0}{2^{n+1}}.$$

Convergence rate of bisection method

The initial error is $\varepsilon_0 := |x^* - x_C^0| \leq (x_R^0 - x_L^0)/2$.

At each iteration, we half the interval so that the error is halved:

$$\varepsilon_n := |x^* - x_C^n| \leq \frac{x_R^n - x_L^n}{2} = \ldots = \frac{x_R^0 - x_L^0}{2^{n+1}}.$$

The error of the method at step i can be bounded from above:

$$\varepsilon_n \leq \frac{1}{2^{n+1}}(x_R^0 - x_L^0).$$

Therefore the method converges linearly at rate $\alpha = 1/2$.

Convergence rate of bisection method

The initial error is $\varepsilon_0 := |x^* - x_C^0| \leq (x_R^0 - x_L^0)/2.$

At each iteration, we half the interval so that the error is halved:

$$\varepsilon_n := |x^* - x_C^n| \leq \frac{x_R^n - x_L^n}{2} = \ldots = \frac{x_R^0 - x_L^0}{2^{n+1}}.$$

The error of the method at step i can be bounded from above:

$$\varepsilon_n \leq \frac{1}{2^{n+1}}(x_R^0 - x_L^0).$$

Therefore the method converges linearly at rate $\alpha = 1/2$.

Convergence rate of bisection method

The error of the method at step n can be bounded from above:

$$\varepsilon_n \leq \frac{1}{2^{n+1}}(x_R^0 - x_L^0).$$

Therefore the method converges linearly at rate $\alpha = 1/2$.

Note: The convergence is **not monotone** in general, i.e. it can happen that for some steps n we have $|F(x_{n+1})| > |F(x_n)|$.

The upper bound above guarantees that eventually $\lim_{n\to\infty} x_C^n = x^*$ so that $\lim_{n\to\infty} F(x_C^n) = 0$.

Convergence rate of bisection method

The error of the method at step n can be bounded from above:

$$\varepsilon_n \leq \frac{1}{2^{n+1}}(x_R^0 - x_L^0).$$

Therefore the method converges linearly at rate $\alpha = 1/2$.

Note: The convergence is **not monotone** in general, i.e. it can happen that for some steps n we have $|F(x_{n+1})| > |F(x_n)|$.

The upper bound above guarantees that eventually $\lim_{n\to\infty} x_C^n = x^*$ so that $\lim_{n\to\infty} F(x_C^n) = 0$.

Pros and cons of bisection method

Performance:

- Guaranteed to converge to x^*
- Slow (linear convergence rate)

Other issues:

- We require an initial bracket (2 values), not just an initial guess (1 value)
- In practice we may have to search for a bracket given one point
- The initial bracket $[x_L, x_R]$ may contain more than one zero and it is not clear which it will compute

Newton's Method (recalling)

Assumption: $F(x)$ differentiable with derivative $F'(x)$, initial guess x_0 s.t. $F'(x_0) \neq 0$.

- Start from x_0, compute $F(x_0)$ and $F'(x_0)$
- Newton step n:

$$x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}, \quad n = 0, 1, 2, \ldots$$

- Iterate until $|F(x_n)| < \mathrm{TOL}$ or maximum number of iterations reached.

Generates a sequence of iterates x_n that converges to x^*.

Pros and cons of Newton's method

Performance

- Fast (quadratic convergence rate)
- Not robust

Other issues

- Requires the derivative function
- Requires a "*good*" initial guess

Convergence criteria

(1) The Bisection Method is usually stopped when

$$|b - a| \; < \; TOL_x$$

for a bracket $[a, b]$.

(2) Newton's Method is usually stopped when

$$|F(x)| \; < \; TOL_F$$

TOL_x and TOL_F are appropriately chosen tolerances

Problems?

- (1) does not necessarily imply (2);
- (2) does not necessarily imply (1).

In practice

We usually accept a solution x_n if

- $|x_{n+1} - x_n| < TOL_x$ (beware of stalled iteration!)
- or, $|F(x_{n+1})| < TOL_F$

or, accept we have failed

- if the number of function evaluations/iterations exceeds a user-specified number N_f

The last criteria handles failure *gracefully*

Summary

Two contrasting, classical approaches to the problem

- Bisection method
  - Derivative not needed
  - Robust
  - Slow (linear convergence)
- Newton's method
  - Derivative required
  - Not robust
  - Fast (quadratic convergence)

Next time...

Lecture

Extending Newton's method