Numerical roots finding

(Lectures 4 and 5 of Numerical methods)

Engineering Mathematics 1

Oscar Benjamin and Lucia Marucci
Department of Engineering Mathematics
University of BRISTOL

Numerical analysis

- Most engineering problems can be posed mathematically, but:
  - they do not have an explicit solution (in closed form),
  - are solved *approximately* on computers,
  - continuous quantities are *discretised*: graphs $\rightarrow$ numbers.
- Numerical analysis:
  - theory of how computers solve maths problems
  - the theory of discretisation and algorithms
  - Many algorithms are iterative (see next lecture):
  - get sequences of better and better answers
  - but do these answers converge to the true answer?

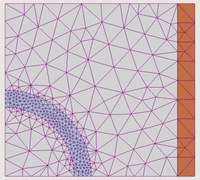
Engineering Hotspot: Computational engineering

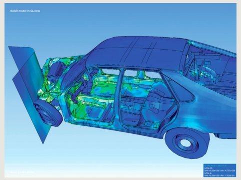
Most engineering software - CFD, FEA, Matlab/Simulink

- creates sequence of approximate solutions (vectors of numbers)
- but do they converge, and What happens if they don't?!

Finite Element Analysis (FEA)

FE breaks a structure into thousands/millions of small elements:





Do we converge to reality as we increase the number of elements?

The rate of convergence is also important.

Root-finding

Root-finding refers to finding the *roots* of functions i.e. to find x such that

$$f(x) = 0$$

This is equivalent so solving an arbitrary equation for x since we can rearrange any equation into this form by subtracting the right-hand-side. e.g.

$$\mathrm{e}^x = \frac{2}{1+x} \quad \rightarrow \quad \mathrm{e}^x - \frac{2}{1+x} = 0$$

So for this problem we have

$$f(x) = \mathrm{e}^x - \frac{2}{1+x}$$

and the solutions of the equation are the roots of f.

Fixed-point iteration method

Let's find approximate solutions of the equation

$$f(x) = 0$$

If $f(x)$ is, for example, a polynomial, we can find exact roots, but this is not always possible! We want to use numerical methods to find approximate solutions.

Fixed-point iteration method

Firstly, we rearrange

$$f(x) = 0 \tag{1}$$

as

$$x = g(x) \tag{2}$$

so that any solution of the latter, which is a fixed point of g, is a solution of the original equation.
We can make a sequence of values $x_0, x_1, x_2 \ldots$ using

$$x_{n+1} = g(x_n) \tag{3}$$

if this sequence converges then it converges to a root of f.

Fixed point iteration

- Example: Apply iteration to solve:

$$\cos x = x.$$

- Procedure *(keep pressing "cos" on your calculator)*
  - Choose initial guess $x_0 = 0$ for root
  - Let $x_1 = \cos x_0$
  - Let $x_2 = \cos x_1$
  - Let $x_3 = \cos x_2$
  - And so on and so on and so on etc.
    (should give better and better approximations to root)
- i.e. form a sequence $x_{n+1} = \cos(x_n)$

First 30 iterates

0
1.00000000000000
0.54030230586814
0.85755321584639
0.65428979049778
0.79348035874257
0.70136877362276
0.76395968290065
0.72210242502671
0.75041776176376
0.73140404242251
0.74423735490056
0.73560474043635
0.74142508661011
0.73750689051324
0.74014733556788
0.73836920412232
0.73956720221226
0.73876031987421
0.73930389239691
0.73893775671534
0.73918439977149
0.73901826242741
0.73913017652967
0.73905479074692
0.73910557192654
0.73907136529894
0.73909440737909
0.73907888599499
0.73908934140339

Rate of converge

- Answers converge towards $L = 0.739085\ldots$
- Some natural questions:
  - What would be the error after 80 steps?
    $\Rightarrow$ *rate of convergence*
  - Does it work for other x_0 or functions other than $\cos$?
    $\Rightarrow$ *convergence and stability analysis*
- Exercise: compute the ratio of differences
  $\dfrac{x_{n+1} - x_n}{x_n - x_{n-1}}$ for the last few iterates
- answer is approx $-0.6736\ldots \approx -\sin(L)$.
- Why? ... (note $-\sin(x) = \frac{d}{dx}\cos(x)$)

What can go wrong

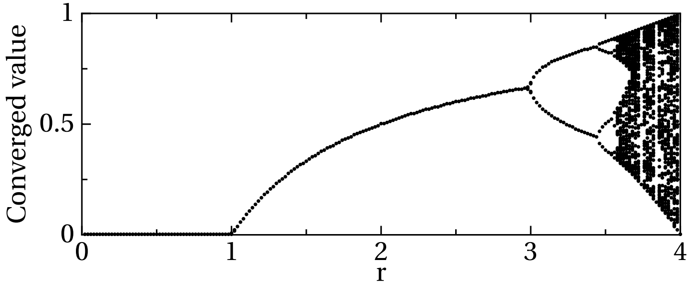
1. Divergence:
   Example: x_{n+1}=$g(x_n)$=$x_n^3 - 1$:
   - there is a solution (draw a graph!)
   - but, using $x_0 = 0$:
     $x_1 = -1$,
     $x_2 = -2$
     $x_3 = -9$,
     $x_4 = -730$,
     $x_5 = -3.8 \times 10^8\ldots$
   - using $x_0 = 2$:
     $x_1 = 7$,
     $x_2 = 342$,
     $x_3 = 4 \times 10^7\ldots$
2. Cycles ... or Chaos ...

Engineering Hotspot: Chaotic dynamics

- Models of population dynamics can lead to recursive sequences, e.g:

$$x_{n+1} = r x_n (1 - x_n) \qquad \text{“The logistic equation”}$$

- For $r < 1$ this converges to $L = 0$
- Limit behaviour for different r:



Theory of fixed point iteration I

- Consider fixed point iteration $x_{n+1} = g(x_n)$
- Let root be L, i.e. $L = g(L)$
- Let $E_n := x_n - L$ denote error of n^{th} iterate
- using *Taylor series* (see next term) for small E_n

$$L + E_{n+1} = g(x_n) = g(L + E_n) \approx g(L) + g'(L)E_n = L + g'(L)E_n$$

where $g'(x) = \frac{dg(x)}{dx}$. Cancelling L from both sides:

$$E_{n+1} \approx g'(L)E_n$$

- “error at step $n+1$ is $r = g'(L)$ times error at step n”

Theory of fixed point iteration II

- Let $r = g'(L)$ be (theoretical) rate of convergence
- For x close enough to L, error grows approximately by factor r each step
- We want this error to decrease:

Necessary condition for convergence

Given a fixed point iteration $x_{n+1} = g(x_n)$, a necessary condition for convergence to a root L is

$$|g'(L)| < 1$$

- If $0 < g'(L) < 1$ convergence is monotonic
- If $-1 < g'(L) < 0$ convergence is oscillatory

Return to example $g(x) = x^3 - 1$

- We are trying to find $x = L$ that solves $x = g(x)$
- Clearly $L > 1$ (draw the graph).
- But $g'(x) = 3x^2 > 3$ for $x > 1$
- However, we can rearrange

$$x = x^3 - 1 \quad \Rightarrow x^3 = x + 1 \quad \Rightarrow x = (x+1)^{1/3}$$

- suggests another method

$$x_{n+1} = (x_n + 1)^{1/3}$$

- Show theoretically that this converges (for x_0 close to root)
- e.g. $x_1 = 1, \quad x_2 = \quad, \quad x_3 = \quad,$

Exercise

- Consider the equation

$$x = x^2 - 2$$

  which has a root at $x = 2$
- show theoretically that the fixed point iteration scheme

$$x_{n+1} = x_n^2 - 2$$

  to try to find this root must diverge
- find a rearranged iteration scheme that will converge to $x = 2$

Engineering Hotspot: large scale engineering computation

- In CFD, FEA, etc. often have to solve for $O(10^6)$ unknowns;

$$\text{i.e. solve} \qquad A\mathbf{x} = \mathbf{b}$$

  with A an $n \times n$ matrix with $n = O(10^6)$
- Gaussian (row) elimination takes $O(n^3) = O(10^{18})$ steps.
- Instead use fixed point iteration for matrices:

Iterative solvers of matrix equations

Let $A = L + D + U$ (Lower triangular + Diagonal + Upper triangular)

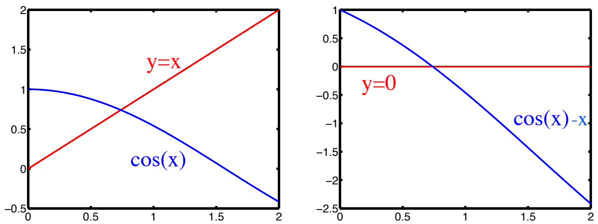
$$\text{Jacobi iteration:} \quad D\mathbf{x}_{n+1} = -(L+U)\mathbf{x}_n + \mathbf{b}$$

$$\text{or, Gauss-Seidel iteration:} \quad D\mathbf{x}_{n+1} = -U\mathbf{x}_n - L\mathbf{x}_{n+1} + \mathbf{b}$$

See James section 5.5.4

Numerical Root finding, other methods

- We saw how the iteration method can solve $x = g(x)$ provided $|g'(x)| < 1$.
- But is there a better way?
- More generally, how do computers find accurate solutions to $f(x) = 0$? e.g. $f(x) = \cos(x) - x$



Use of intermediate value theorem

- Idea, want to find a root (or zero) $x = x^*$ that solves $f(x) = 0$, for some function f.
- Sufficient condition for existence of a root:

Version of intermediate value theorem (IMVT)

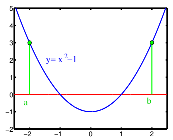
Suppose $f : \mathbb{R} \to \mathbb{R}$ is a continuous function, and $a < b$ such that $f(a)$ and $f(b)$ are nonzero and of opposite signs, then there exists x^* with $a < x^* < b$ such that $f(x^*) = 0$.

- Note, this is a sufficient condition, not a necessary condition . . .

Three lessons

1. $f(a)$, $f(b)$ must be opposite sign e.g

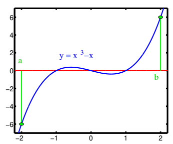
$$f(x) = x^2 - 1, \qquad a = -2, \quad b = +2$$



IMVT does not apply, but there are roots

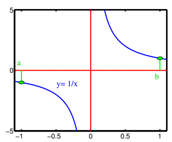
2. even if IMVT applies, can be multiple roots

$$f(x) = x^3 - x, \qquad a = -2, \quad b = +2$$



3. f must be continuous, e.g.

$$f(x) = \frac{1}{x},$$

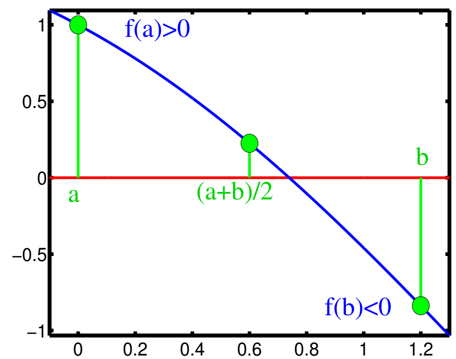


no roots between $a = -1$, $b = +1$ even though $f(a)f(b) < 0$

Bisection method

Basic graphical idea: use IMVT repeatedly:

- Let $f(x)$ be continuous with $f(a_1)f(b_1) < 0$
- then for $n = 1, 2, \ldots$
- Let $c_n := (a_n + b_n)/2$
- If $f(c_n) = 0$, stop algorithm
- If $f(c_n)f(a_n) < 0$, then let $a_{n+1} = a_n$, $b_{n+1} = c_n$
- If $f(c_n)f(b_n) < 0$, then let $a_{n+1} = c_n$, $b_{n+1} = b_n$



- Keep halving interval, use midpoint of interval as guess of root
- Stop when half interval length less than tolerance

Convergence of the bisection method

Exercise find root of $f(x) = \cos(x) - x$ with $a_1 = 0,\ b_1 = 1$ to 2DP.

The method is Robust, but it converges very slowly

- After n steps, length of interval $= (b_1 - a_1)2^{-n}$
- Using midpoint as guess of root:

$$\text{worst case error} = \frac{1}{2}(b_1 - a_1)2^{-n} \qquad \text{NB: } E_{n+1} \simeq \frac{1}{2}E_n$$

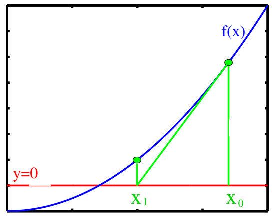
- Implies rate of convergence $r = 0.5$
- To get error less than ϵ, need $(b_1 - a_1)2^{-(n+1)} < \epsilon$, so need

$$n + 1 > \frac{\log(b_1 - a_1) - \log \epsilon}{\log 2}$$

$\Rightarrow$ need a quicker method

Newton's method (I)

Also known as the Newton-Raphson method



- Use tangent to generate sequence of approximations

$$\frac{f(x_0)}{x_0 - x_1} = f'(x_0), \qquad \text{so} \qquad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

- Note: requires not only that f is continuous, but also that its derivative $f' = \frac{df}{dx}$ exists and is continuous.

Newton's method (II)

Newton's method

Choose initial guess x_0 of root, compute sequence of better approximations x_1, x_2, x_3, etc. via

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

until $|x_{n+1} - x_n| <$ tolerance or $|f(x_n)| <$ tolerance

Exercise apply Newton's method to our favourite equation $x = \cos x$

- $f(x) = x - \cos x,\ f'(x) = 1 + \sin x$
- General iteration:

$$x_{n+1} = x_n - \left(\frac{x_n - \cos x_n}{1 + \sin x_n}\right)$$

- Take e.g. $x_0 = 0$. Then

$$x_1 = 0 - (0 - \cos(0))/(1 + \sin(0)) = 1,$$
$$x_2 = 1 - (1 - \cos(1))/(1 + \sin(1)) = 0.75036\ldots$$

Convergence of Newton's method

Compare convergence for solving $x = \cos(x)$:

Newton's method:

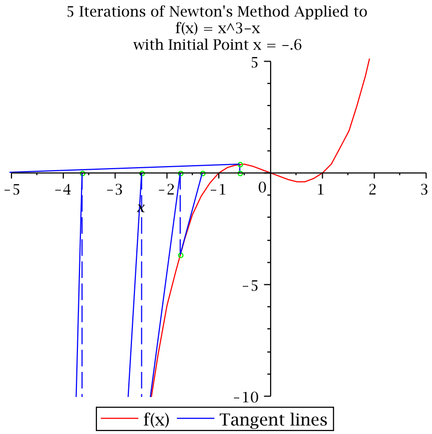
n	x_n
0	0.000000000000000
1	1.000000000000000
2	0.750363867840244
3	0.739112890911362
4	0.739085133385284
5	0.739085133215161
6	0.739085133215161
7	0.739085133215161

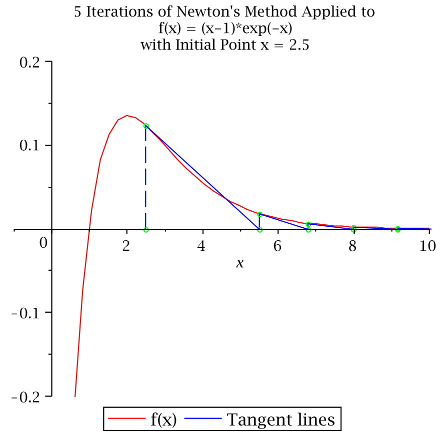
Standard fixed point method:

n	x_n
0	0
1	1.00000000000000
2	0.54030230586814
3	0.85755321584639
4	0.65428979049778
5	0.79348035874257
6	0.70136877362276
7	0.76395968290065

- Upshot: Newton's method converges much faster
- Actually, Newton has $r = 0$: Error$_{n+1} \propto (\text{Error}_n)^2$

What can go wrong with Newton's method





What can go wrong with Newton's method

1. May jump way outside of interval before converging, e.g.

$$f(x) = x^3 - x, \qquad x_0 = -0.6$$

2. Or may diverge, e.g.

$$f(x) = (x-1)e^{-x}, x_0 = 2.5$$

$\Rightarrow$ not as robust as bisection.

- *Read* James 4th edition, section 7.9.3
- *Do* exercises James 4th edition, 7.9.4: 64
- *Read* James 4th edition, section 9.4.8
- *Do* James 4th edition, exercises 9.4.10, 25 & 26