II. Nonlinear Equations

Goals: - solve nonlinear (systems of) equations numerically
- understand that this can be hard...

Why? - Generally, the nonlinear equations that appear in practice are not solvable by analytical means $x = \ldots$

- Implicit methods for ODEs ... Very important for stiff problems (... very common in practice)
- Steady state CSTR example

MATLAB: fsolve

II.1 Single nonlinear equations

Problem: Solve $f(x) = 0$ for $f: D=[a,b] \to \mathbb{R}$

$\rightsquigarrow$ Iterative techniques

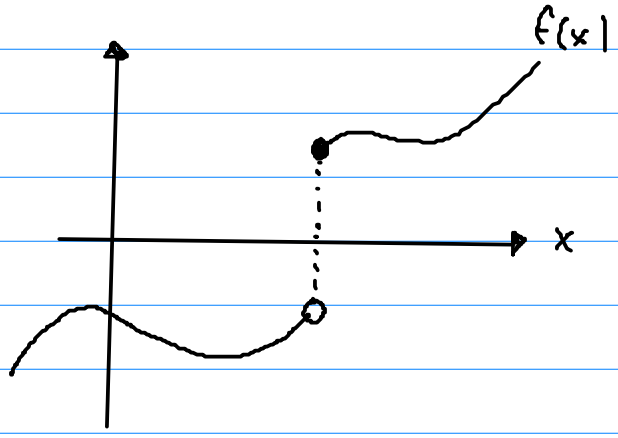
$$x^{(0)} \longrightarrow x^{(1)} \longrightarrow x^{(2)} \longrightarrow \ldots \longrightarrow x^* = \lim_{k \to \infty} x^{(k)}$$

$x^{(0)}$: starting value or initial guess

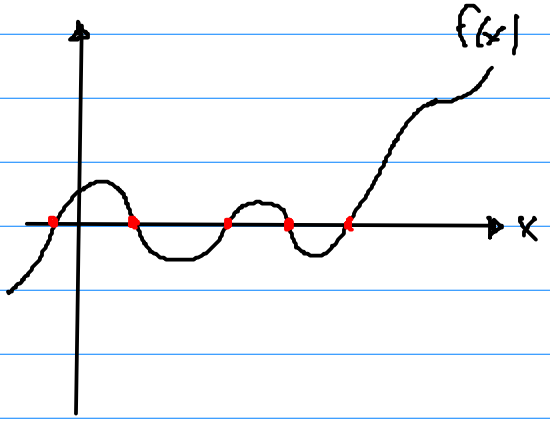
$x^{(1)}$: some algorithm

$f(x^*) = 0$

Remark: In general no existence nor uniqueness



f not continuous



several solutions

Assume f continuous in the following...

II.1.1 Fixed-point iterations and a few generalities

Our problem $f(x) = 0$ can be written as

$$x = \phi(x)$$

for some $\phi(x)$. This is known as a fixed-point equation (FPE) and x^* satisfying

$$x^* = \phi(x^*)$$

as a fixed-point (FP).

Def.: A fixed-point iteration (FPI) is defined as

$$x^{(k+1)} = \phi(x^{(k)}), \quad k = 0, 1, 2, \ldots$$

Def.: A fixed-point equation is said to be consistent with the original equation if

$$f(x^*) = 0 \iff x^* = \phi(x^*)$$

Ex.: (1) Solve $f(x) = x \cdot e^x - 1 = 0$ for $x \in [0,1]$

(i) $x \cdot e^x - 1 = 0$

$$x \cdot e^x = 1$$

$$x = e^{-x} = \phi_1(x)$$ fixed-point eq. (consistent ✓)

(ii) $x = \phi_2(x)$ with $\phi_2(x) = \dfrac{x^2 \cdot e^x + 1}{e^x (1 + x)}$

(iii) $x = \phi_3(x)$ with $\phi_3(x) = x + 1 - x \cdot e^x$

⇝ 5 slides

... Exercise: show consistency!

Rem.: (i) Fixed-point iterations not unique

(ii) Fixed-point iterations may not converge

(iii) If they converge, they may do that with different speeds

Def.: A sequence $x^{(k)}$ with limit x^* converges with order $p \geq 1$, if there exists a constant $C > 0$ such that

$$|x^{(k+1)} - x^*| \leq C |x^{(k)} - x^*|^p$$

for all sufficiently large k.

For $p = 1$, it must $0 < C < 1$.
The constant C is called the rate of convergence.

In particular, convergence with order $\left\{ \begin{array}{l} p = 1 \\ p = 2 \end{array} \right\}$

is called $\left\{ \begin{array}{l} \text{linear} \\ \text{quadratic} \end{array} \right\}$.

It is often helpful, e.g. for code verification, to measure C and p in numerical experiments.

For this we define the error at the k-th iteration as

$$\varepsilon^{(k)} = |x^{(k)} - x^*|$$

Then we can write

$$\varepsilon^{(k)} = C \cdot \left(\varepsilon^{(k-1)}\right)^p$$

$$\varepsilon^{(k+1)} = C \cdot \left(\varepsilon^{(k)}\right)^p$$

Taking the log on both sides gives

$$\log\left(\varepsilon^{(k)}\right) = \log(C) + p \cdot \log\left(\varepsilon^{(k-1)}\right)$$

$$\log\left(\varepsilon^{(k+1)}\right) = \log(C) + p \cdot \log\left(\varepsilon^{(k)}\right)$$

This can be solved for C and p:

$$p = \frac{\log\left(\varepsilon^{(k+1)}\right) - \log\left(\varepsilon^{(k)}\right)}{\log\left(\varepsilon^{(k)}\right) - \log\left(\varepsilon^{(k-1)}\right)}$$

$$C = \frac{\varepsilon^{(k+1)}}{\left(\varepsilon^{(k)}\right)^p} = \frac{\varepsilon^{(k)}}{\left(\varepsilon^{(k-1)}\right)^p}$$

Ex.: (2) Determine C & p for the fixed-point iterations from Ex. (1) ~~> Slides

Catch: How to compute $\varepsilon^{(k)} = |x^{(k)} - x^*|$?

$= ?$

This brings us to the practical question:

When to stop iterating?

Stopping criteria (SC):

(SC1) $|x^{(k)} - x^{(k-1)}| \leq atol$ (absolute-)

(SC2) $|x^{(k)} - x^{(k-1)}| \leq rtol \cdot |x^{(k)}|$ (relative-)

(SC3) $|x^{(k)} - x^{(k-1)}| \leq tol \cdot (1 + |x^{(k)}|)$ (hybrid-)

(SC4) $|f(x^{(k)})| \leq ftol$ (function-)

tolerance

II.1.2 Bisection method

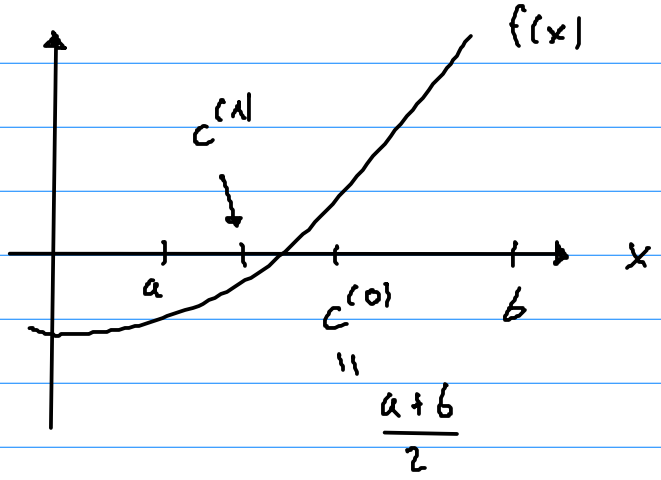
Assume: we know a, b with

$$f(a) < 0 \quad \text{and} \quad f(b) > 0$$

$\rightsquigarrow$ root(s) bracketed

Idea: divide / bisect interval and keep subinterval fulfilling above assumption

...



Rem.: (i) Very easy and robust (need only f)

(ii) A priori error estimate $\varepsilon^{(k)} \leq \frac{b-a}{2^{k+1}}$

(iii) Slow convergence (linear)

(iv) Not easy to generalize to systems

II.1.3 Newton's method

Idea: linearize f

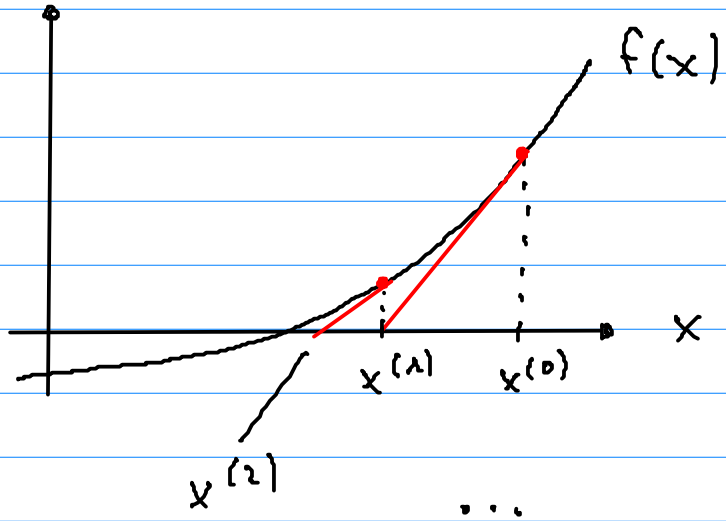
Taylor at $x^{(k)}$:

$$f(x) = \underbrace{f(x^{(k)}) + f'(x^{(k)})\cdot(x - x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x-x^{(k)}) + \dots}_{\tilde{f}(x)}$$

$$\tilde{f}(x) \overset{!}{=} 0 \rightsquigarrow x^{(k+1)}$$

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

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



Rem.: (i) Needs f'

(ii) Quadratic convergence ($p=2$) when close enough

(iii) Can fail, e.g. $f'(x^{(k)}) = 0$

(iv) Generalizes to systems

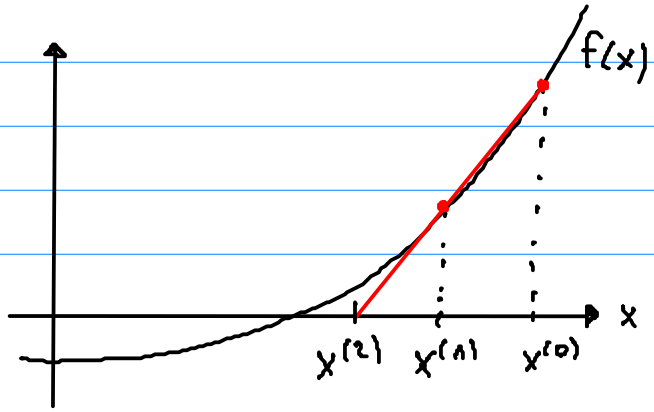
II.1.4 Secant method

Idea: If f' is not directly available or expensive to compute, approx. f' with finite differences

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

Then

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



Rem.: (i) No need for f'!

... so-called quasi-Newton methods

(ii) Converges with $p = \frac{1}{2}(1+\sqrt{5}) \approx 1.618$

... faster than bisection, but slower than Newton

(iii) Fails when $f(x^{(k)}) = f(x^{(k-1)})$

(iv) Generalizes to systems

II.2 Systems of nonlinear equations

Problem: Given an n-dimensional real and continuous function $\vec{f}: D \subset \mathbb{R}^n \to \mathbb{R}^n$, solve

$$\vec{f}(\vec{x}) = 0$$

Short notation for system of nonlinear equations:

$$\left.\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, \dots, x_n) &= 0 \end{aligned}\right\} \quad \vec{f}(\vec{x}) = 0$$

Ex.: (3) $n = 2$, $D = [0,2]^2 \subset \mathbb{R}^2$

$$f_1(x_1, x_2) = x_1^2 + x_2 - 2 = 0$$

$$f_2(x_1, x_2) = x_2 \cdot e^{x_1} - 2 = 0$$

$\rightsquigarrow$ contour / level curves (slides)

II.2.1 Newton's method

Idea: linearize $\vec{f}$

Taylor at $\vec{x}^{(k)}$

Jacobian Matrix

$$Df(\vec{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

$$\vec{f}(\vec{x}) \approx \vec{f}(\vec{x}^{(k)}) + D\vec{f}(\vec{x}^{(k)})(\vec{x} - \vec{x}^{(k)}) + \dots$$

$$\underset{\sim}{\vec{f}} = \vec{f}(\vec{x}^{(k)}) + D\vec{f}(\vec{x}^{(k)})(\vec{x} - \vec{x}^{(k)}) \overset{!}{\simeq} 0$$

$$\rightsquigarrow \quad \vec{x}^{(k+1)} = \vec{x}^{(k)} - D\vec{f}(\vec{x}^{(k)})^{-1} \vec{f}(\vec{x}^{(k)})$$

inverse of Jacobian

Ex.: (4) for example (3)

$$D\vec{f}(\vec{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} = 2x_1 & \frac{\partial f_1}{\partial x_2} = 1 \\ \frac{\partial f_2}{\partial x_1} = x_2 \cdot e^{x_1} & \frac{\partial f_2}{\partial x_2} = e^{x_1} \end{pmatrix} = \begin{pmatrix} 2x_1 & 1 \\ x_2 \cdot e^{x_1} & e^{x_1} \end{pmatrix}$$

$$D\vec{f}^{-1}(\vec{x}) = \frac{1}{(2x_1 - x_2) \cdot e^{x_1}} \begin{pmatrix} e^{x_1} & -1 \\ -x_2 \cdot e^{x_1} & 2x_1 \end{pmatrix}$$

$\rightsquigarrow$ slides

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$A^{-1} = \frac{1}{ad - cb} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

(5) CSTR

$\rightsquigarrow$ slides

Rem.: (i) Needs Jacobian

(ii) Quadratic convergence when close enough

(iii) Fails when Jacobian singular

i.e. $D\vec{f}$ not invertible

... $\sim f'(x) = 0$ ...

04.10.23