Numerical methods for nonlinear systems of equations

find x^*: $f(x^*) = 0$, e.g. $f(x) = x^2 + 2x + 1 \Rightarrow x^* = -1$

$f(x) = x e^x - 1 \Rightarrow x^* = ?$

minimize Kost / Energy: $E(x)$; $\min E(x)$

$\Downarrow$

$DE(x^*) = 0$

$f(x) = DE(x) \qquad x \in \mathbb{R}^d$

In general: $F: \mathbb{R}^n \to \mathbb{R}^n$, find $x^* \in \mathbb{R}^n$ s.t. $F(x^*) = 0$

$\nearrow$

system of n equations

Note: can not get x^* exactly !

Hence, we just approximate x^*.

In practice: $x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ s.t. $x^{(k)} \to x^*$

or iterative method = algorithm producing $x^{(k)} \to x^*$

Convergence speed is then essential !

Def $x^{(k)}$ linearly convergent to x^* if.

there is $L < 1$ such that convergence rate

$$\underbrace{\| x^{(k+1)} - x^* \|}_{e^{(k+1)}} \leq L \underbrace{\| x^{(k)} - x^* \|}_{e^{(k)}}$$

Def Convergence of order p if

there is $C > 0$ such that

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

with $C < 1$ for $p = 1$.

Ex 1) $\phi(x) = x + \frac{1 + \cos x}{\sin x}$, $x^{(k+1)} = \phi(x^{(k)})$, lin. conv.

2) $\phi(x) = \frac{1}{2}\left(x + \frac{a}{x} \right) \Rightarrow p = 2$ quadratic convergence
$\sqrt{a}$

When to stop a convergent iteration?

want: $\| x^{(k)} - x^* \| \leq \tau =$ tolerance.

→ after k_0 steps :(

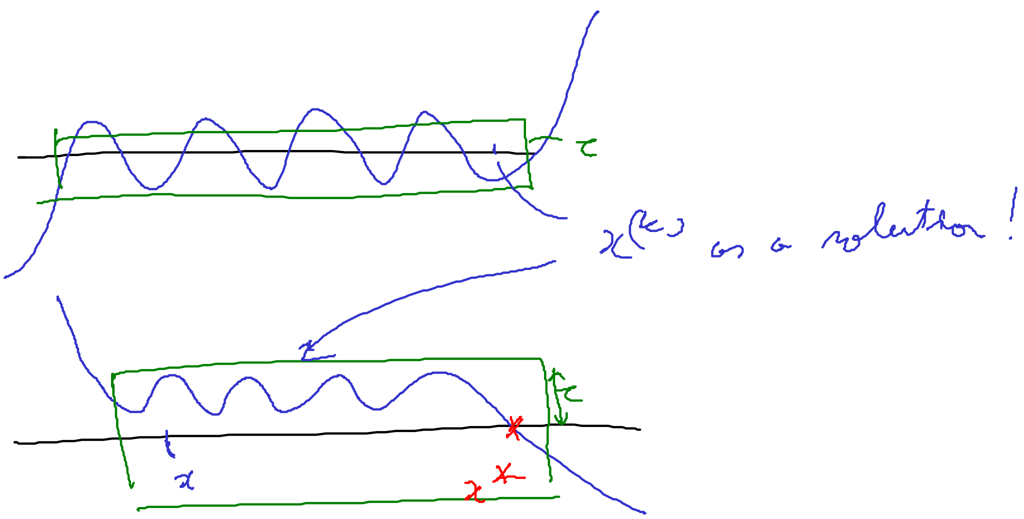
→ use the computed $x^{(k-1)}, x^{(k-2)}, \dots$

"a posteriori" criteria

$\| x^{(k)} - x^* \| \leq \tau$, but we don't know x^*

→ $\| x^{(k)} - x^{(k+1)} \| \approx 0 \Rightarrow$ stop the iteration

→ $F(x^*) = 0 \Rightarrow$ stop if $\| F(x^{(k)}) \| \approx 0$

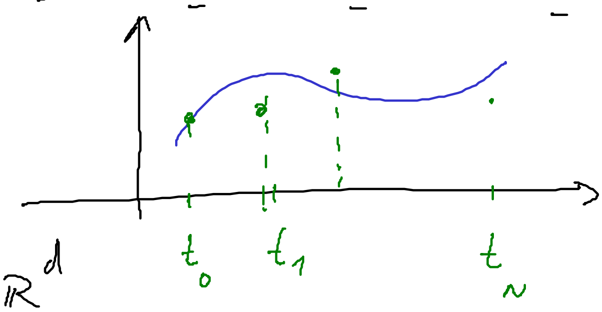


Note: $\dot{x}^{(k)}$ may be quite far away from x^*,
but in some applications it can be conceivable,
e.g. $\|F(x^{(k)})\|$ close to zero is important

ODEs: unknown $\underline{u}: [0,T] \to \mathbb{R}^d$

$t_0 = 0$

$\underline{u}(t_0), \underline{u}(t_1), \underline{u}(t_2), \dots, \underline{u}(t_N)$; $t_N = T$



Now: find $\underline{x}^* \in \mathbb{R}^d$
s.t. $F(\underline{x}^*) = 0$

Argue: impossible to get $\underline{x}^*$ so search approximation
$\underline{x}^{(0)}, \underline{x}^{(1)}, \underline{x}^{(2)}, \dots, \underline{x}^{(k)} \simeq x^{(*)}$

Iteration $k \neq$ time-stepping for ODEs

Speed of convergence

here: linear convergence with rate L

$\|x^{(k+1)} - x^*\| \le L \|x^{(k)} - x^*\|$ with $0 < L < 1$

order p: $\|x^{(k+1)} - x^*\| \le \|x^{(k)} - x^*\|^p$

$\neq$ convergence order at ODEs

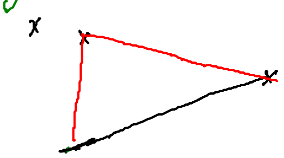
$O(h^p)$: $\|u_N - u(t_N)\| \le C \cdot h^p$

$x^{(0)}, x^{(1)}, \dots, x^{(k)}$

Ex If we know, we have a lin. convergence with rate L:

$\|x^{(k+1)} - x^*\| \le L \|x^{(k)} \pm x^{(k+1)} - x^*\| \le L\left(\|x^{(k)} - x^{(k+1)}\| + \|x^{(k+1)} - x^*\|\right)$

triangle inequality



$\Rightarrow \underbrace{(1-L)}_{>0} \|x^{(k+1)} - x^*\| \le L \|x^{(k)} - x^{(k+1)}\| \Rightarrow$

$$\Rightarrow \boxed{\| x^{(k+1)} - x^* \| \le \frac{L}{1-L} \| x^{(k-1)} - x^{(k)} \|}$$

Suppose we want $\| x^{(k+1)} - x^* \| < \tau$ tolerance

$$\Rightarrow \text{enough } \boxed{\frac{L}{1-L} \| x^{(k-1)} - x^{(k)} \|} < \tau$$

$\Rightarrow$ enough to look at $\| x^{(k-1)} - x^{(k)} \| \cancel{<} \frac{1-L}{L}$

if $\| x^{(k-1)} - x^{(k)} \| \le \tau \cdot \frac{1-L}{L}$, then

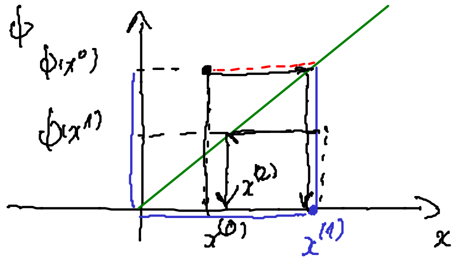
$\| x^{(k+1)} - x^* \| \le \tau \Rightarrow$ hence stop at step $(k+1)$.

Fixed point iterations

Idea: relate $F(x) = 0$ "find a zero of F" with $\phi(x) = x$ "find a fixed point of ϕ"

$\Downarrow$

iteration: $x^{(k+1)} = \phi(x^{(k)})$



Example $F(x) = x e^x - 1$

$F(x) = 0 \Leftrightarrow -x e^x + 1 = 0 \Leftrightarrow \boxed{-x e^x + x + 1 = x} \Leftrightarrow$ ϕ_3

$x(e^x + 1) = x + 1 \Leftrightarrow$

$\boxed{x = \frac{x+1}{e^x+1}}$ ϕ_2

$x e^x - 1 = 0 \Leftrightarrow x e^x = 1 \Leftrightarrow \boxed{x = e^{-x}}$ ϕ_1

start by $x^{(0)}$; then we have 3 possibilities to create an iteration:

1) $x^{(1)} = \phi_1(x^{(0)})$, $x^{(2)} = \phi_1(x^{(1)})$, ..., $x^{(k+1)} = \phi_1(x^{(k)})$

2) ϕ_2 ϕ_2 ϕ_2

3) ϕ_3 ϕ_3 ϕ_3

Def $\phi : U \subset \mathbb{R}^d \to U \subset \mathbb{R}^d$ is called contraction:

if there is a constant $0 \le L < 1$ such that

$$\| \phi(x) - \phi(y) \| \le L \cdot \| x - y \| \quad \text{for all } x, y \in U$$

T// ϕ contraction $\Rightarrow$ $x^{(k+1)} = \phi(x^{(k)})$ convergent iteration!

T// $\phi : U \subset \mathbb{R}^d \to U$, $\phi(x^*) = x^*$, ϕ differentiable in x^*, $\| D\phi(x^*) \| < 1$. Then the iteration $x^{(k+1)} = \phi(x^{(k)})$ converges $\boxed{\text{locally}}$ linearly and $L = \| D\phi(x^*) \|$.

we have to start at $x^{(0)}$ close to $x^{(*)}$

T// $\phi : U \subset \mathbb{R} \to \mathbb{R}$, U = interval, ϕ $(m+1)$-times differentiable

assume $\phi(x^*) = x^*$ and $\phi^{(l)}(x^*) = 0$ for $l = 1, 2, \dots, m$.

with $m \ge 1$. Then the iteration $x^{(k+1)} = \phi(x^{(k)})$ converges to x^* locally with order $p \ge m+1$.

Example 1) $\phi_1(x) = e^{-x}$, $\phi_1'(x^*) = -e^{-x^*}$, look for $x^* \in]0,1]$ $\Big\} \Rightarrow |\phi_1'(x^*)| < 1$

$\Downarrow$

it converges at least linearly

2) $\phi_2(x) = \dfrac{x+1}{e^x+1} \Rightarrow \phi_2'(x) = \dfrac{e^x + 1 - (x+1)e^x}{(e^x+1)^2} = \dfrac{-xe^x + 1}{(e^x+1)^2} =$

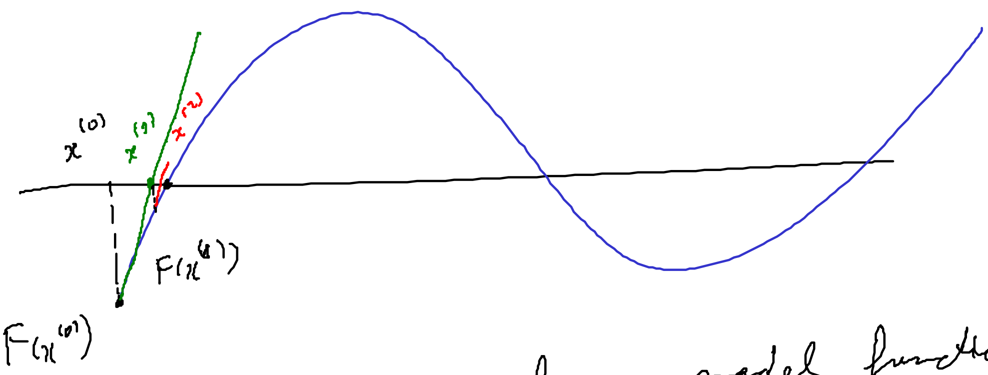
$$= \frac{F(x)}{(e^x+1)^2} \Rightarrow \phi_2'(x^*) = \frac{F(x^*)}{(e^{x^*}+1)^2} = 0$$

$\Rightarrow$ it converges at least quadratically!

3) $\phi_3(x) = -\dfrac{1}{x}$, $x^* \in]0,1] \Rightarrow \phi_3'(x^*) = -\dfrac{1}{x^*}$

$|\phi_3'(x^*)| > 1$

Newton Method



Idea: replace $F(x)$ by a model function $\tilde{F}(x)$ and approximate $x^* \approx$ solution of $\tilde{F}(x) = 0$

example: approximate $F(x)$ by its linearization near $x^{(0)}$:

$$F(x) \approx F(x^{(0)}) + \cancel{(x - x^{(0)})\, DF(x^{(0)})}\ DF(x^{(0)})(x - x^{(0)}) =: \tilde{F}(x)$$

Newton !

look at
$$F(x^{(0)}) + \cancel{(x - x^{(0)})\, DF(x^{(0)})}\ DF(x^{(0)})\,(x - x^{(0)}) = 0 \iff$$

$$(x - x^{(0)})\, DF(x^{(0)}) = -F(x^{(0)})$$

$$DF(x^{(0)})\,(x - x^{(0)}) = -F(x^{(0)})$$

$\hookrightarrow$ in general, this is a linear system of equations, to be solved

$$DF(x^{(0)})^{-1} \Big| \quad \Rightarrow \quad x - x^{(0)} = -DF(x^{(0)})^{-1} F(x^{(0)})$$

$$\Rightarrow \quad \tilde{x}^* := x^{(0)} - DF(x^{(0)})^{-1} F(x^{(0)})$$

$$x^{(1)} := \tilde{x}^* = x^{(0)} - DF(x^{(0)})^{-1} F(x^{(0)})$$

Newton-Iteration: start by an initial guess $x^{(0)}$ close to x^*

$$\boxed{x^{(k+1)} = x^{(k)} - DF(x^{(k)})^{-1} F(x^{(k)})}$$

Note Never compute the inverse of a matrix.

Instead, solve linear systems !

MATLAB: $x^{new} = x^{old} - DF(x^{old}) \backslash F(x^{(k)})$

$x^{new} = x^{old} - \text{solve}(\cancel{D}F'(x^{old}), F(x^{(k)}))$

Remarks

1) sometimes is better to think:

$$F(x) = \frac{1}{(x+1)^2} + \frac{1}{(x+0.1)^2} - 1 \quad \text{for } x > 0$$

$$F(x) + 1 \simeq 2x^{-2} \quad \text{for } x \gg 1$$

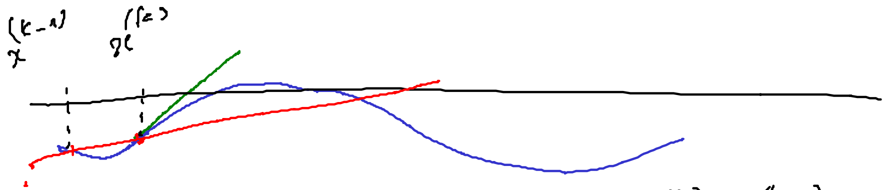
idea: $g(x) = \dfrac{1}{\sqrt{1+F(x)}} \simeq$ linear for $x \gg 1$

Hence rewrite $F(x) = 0 \iff g(x) = 1$ and apply Newton for $g(x) - 1 = 0$.

Whenever you can: plot!

2) $F: \mathbb{R} \to \mathbb{R}$ but F is abstract so do not know what is F'!

Idea: approximate $F'(x^{(k)}) \simeq \dfrac{F(x^{(k)}) - F(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$



method of secant

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

Note: replacing the derivative by the finite differences costs in lowering the convergences order:

from order 2 for Newton

to order $p = 1.62 < 2$ for secant method.

Note Newton method is locally convergent of order 2.

$x^{(k+1)} = x^{(k)} - DF(x^{(k)})^{-1} F(x^{(k)})$ is a fixed point iteration

$$\phi(x) = x - DF(x)^{-1} F(x) \qquad \frac{F(x)}{F'(x)}$$

$F: \mathbb{R} \to \mathbb{R}$ $\qquad \phi'(x) = 1 - \dfrac{F'(x)F'(x) - F(x)F''(x)}{(F'(x))^2} =$

$$= 1 - \left(1 - \frac{F(x)F''(x)}{(F'(x))^2}\right) = \frac{F(x)F''(x)}{(F'(x))^2}$$

$F(x^*) = 0 \Rightarrow \phi'(x^*) = 0 \Rightarrow$ at least second order convergence

provide $F'(x^*) \neq 0$

if $F'(x^*) = 0$ $\quad (F(x) = (x-1)^2)$ not clear!

Newton works only if $DF(x^{(k)})$ invertible!

<u>Remark</u> Newton:

$$\underline{\underline{DF}}(\underline{x}^{(k)})\, \underline{\Delta} = -\underline{F}(\underline{x}^{(k)})$$

$\left.\begin{array}{l}\text{if } x \in \mathbb{R}^d,\ d \text{ large} \\ \text{or } DF \text{ expensive}\end{array}\right\} \Rightarrow$ computationally intensive

huge amount of comput. time!

one cheap way to lower the comp. task is to

1) reuse $\underline{\underline{DF}}(x^{(k)})$ for several steps

"simplified Newton"

2) use decomposition of $\underline{\underline{DF}}(x^{(k)}) = LU$

$\underline{\underline{A}}\, \underline{x} = \underline{b}$ Gauss elimination

$$\underline{\underline{A}} = \underline{\underline{L}}\, \underline{\underline{U}}, \qquad \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} \begin{bmatrix} * & * \\ 0 & * \end{bmatrix}$$

$L \underbrace{\boxed{Ux}}_{y} = b$ $\quad$ solve for $Ly = b$ cheap; $\quad$ solve for $Ux = y$ cheap

$\Rightarrow$ reuse the factors L and U for several steps!

Price to pay: only linear convergence.

find x^*: $F(x^*)=0$: Newton-method: $x^{(0)}$ = initial guess near x^*

iteration:

solve $DF(x^{(k)})\, s = F(x^{(k)})$

Newton correction

define $x^{(k+1)} := x^{(k)} - s$

Stoping criterium:

$$\| x^{(k)} - x^* \| \simeq \| x^{(k)} - x^{(k+1)} \| = \|s\| = \underbrace{\| DF(x^{(k)})^{-1} F(x^{(k)}) \|}_{\le \tau \text{ tolerance}}$$

Newton-method is quadratically convergent $\Rightarrow$ stop the iteration

$F : \mathbb{R}^d \to \mathbb{R}^d$, d = large $\Rightarrow$ large system $\Rightarrow$

solving the linear system might be expensive.

possible aproach: reuse the old derivative / inv.

$DF(x^{(k-1)})$ instead of $DF(x^{(k)})$

$\Rightarrow$ stopping before computing $x^{(k+1)}$:

(*) have $x^{(k)}$ and $DF(x^{(k-1)})^{-1}$ e.g. as factors L, U of $DF(x^{(k-1)})$

(*) if $\underbrace{\| DF(x^{(k-1)})^{-1} F(x^{(k)}) }_{\text{simplified Newton correction at step } k \mapsto k+1}\| \le \tau$ then stop!

Damped Newton Method

Numerical examples (arctan(ax)) show strong dependence of the convergence of Newton-method on initial guess, mainly due to overshooting = = too large steps!

$\longrightarrow$ damp the Newton corrections!

Idea: choose damping factor $\lambda^{(k)}$ s.t.

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} DF(x^{(k)})^{-1} F(x_k)$$

$$0 < \lambda^{(k)} < 1$$

choose $\lambda^{(k)}$ as the largest λ s.t.

$$\Delta\bar{x}(\lambda) := DF(x^{(k)})^{-1} F(x^{(k)} + \lambda \Delta x^{(k)})$$

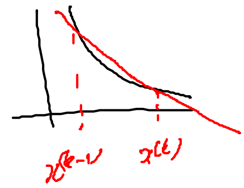
$$\|\Delta\bar{x}(\lambda)\| \le \left(1-\frac{\lambda}{2}\right)\|\Delta x^{(k)}\|$$

Quasi-Newton Method

What is if we do not have $DF(x^{(k)})$?

$F: \mathbb{R}\to\mathbb{R}$ replace $F'(x^{(k)})$ by $\dfrac{F(x^{(k)})-F(x^{(k-1)})}{x^{(k)}-x^{(k-1)}}$

$\Rightarrow$ secant method.



$F:\mathbb{R}^d\to\mathbb{R}^d$

need: $\underline{\underline{J}}^{(k)} \approx \underline{\underline{DF}}(x^{(k)})$ such that:

$$\boxed{\underline{\underline{J}}_k (x^{(k)} - x^{(k-1)}) = F(x^{(k)}) - F(x^{(k-1)})}$$

many possible solutions for $\underline{\underline{J}}_k$

Wish: $\underline{\underline{J}}^{(k)}$ via small modifications of $\underline{\underline{J}}^{(k-1)}$

optimal way: Broyden method

optimal implementation Shermann-formula

↪ used inside fsolve or others professional solvers.