Felanalys

Approximationsfel

Datorer representerar nästan alla tal approximativt.

Notation Låt $x \in \mathbb{R}$. En APPROXIMATION av x betecknas $\hat{x}$.

Ex. $x = 3.141592\ldots \qquad \hat{x} = 3.14$

$$f(x) = e^x \qquad \hat{f}(x) = 1 + x + \frac{1}{2}x^2$$

Två sätt att beskriva felet

Absoluta felet: $\delta x := \hat{x} - x$

Relativa felet: $\dfrac{\delta x}{x} = \dfrac{\hat{x} - x}{x} = \dfrac{\text{"absoluta felet"}}{\text{"exakta värdet"}}$

Ex. $x = \pi \quad \hat{x} = 3.14$ (*)

$\delta x \approx \underset{\uparrow}{-}\,0.15927 \cdot 10^{-2} \qquad \dfrac{\delta x}{x} \approx -0.508 \cdot 10^{-3}$

OBS tecken

Om man inte känner x kan relativa felet approximeras $\dfrac{\delta x}{x} \approx \dfrac{\delta x}{\hat{x}}$ (om $\hat{x} \neq 0$)

Felgränser

Konstanter c_1, c_2 är FELGRÄNSER för respektive fel om:

$$|\delta x| \le c_1, \qquad \left|\frac{\delta x}{x}\right| \le c_2$$

I (*) är $c_1 = 0.2 \cdot 10^{-2}$ & $c_2 = 0.6 \cdot 10^{-3}$

Om det absoluta felet uppfyller $|\delta x| \leq 0.5 \cdot 10^{-n}$ sägs att approximationen $\hat{x}$ har (minst) n KORREKTA DECIMALER

OBS: n kan vara negativ!

OBS! Ej som i boken

Om det relativa felet uppfyller $|\delta x / x| < 0.5 \cdot 10^{-n}$ säger vi att approximationen $\hat{x}$ har (minst) n SIGNIFIKANTA SIFFROR.

Ex $x = \pi \quad \hat{x} = 3.14$

$\Rightarrow$ 2 korrekta decimaler, 2 signifikanta siffror

Felfortplantning

Antag att vi har en funktion /algoritm $f : I \to \mathbb{R}$, $I \subset \mathbb{R}$ och att vi vill beräkna $f(x)$ för $x \in I$.

Felfortplantning studerar hur fel i indata $\hat{x} - x$ fortplantar sig till fel i utdata $f(\hat{x}) - f(x)$.

Om vi antar att $f \in C^1(I)$ ger MV-satsen:

$$\delta f(x) = f(\hat{x}) - f(x) = f'(x + \theta(\hat{x} - x))(\hat{x} - x) =$$
$$= f'(x + \theta \delta x)\delta x \quad \text{där } \theta \in [0, 1]$$

Obs: $|\text{utdatafel}| > |\text{indatafel}|$ om $|f'| < 1$ och vice versa.

Om inte θ eller x är bekanta används FÖRSTA ORDNINGENS APPROXIMATIVA FELFORTPLANTNINGSFORMEL

$$\delta f(x) \approx f'(\hat{x})\delta x$$

Med $\delta f(x) \approx f'(\hat{x})\delta x$ blir felgränser:

$$|\delta f(x)| \lessapprox |f'(\hat{x})||\delta x| \quad \underline{o} \quad \left|\frac{\delta f(x)}{f(\hat{x})}\right| \lessapprox \left|\frac{f'(\hat{x})}{f(\hat{x})}\right| |\delta x|$$

<u>Exempel</u> Volymen av ett klot m. radie r.
Antag att vi mäter $\hat{r}$ med 1% relativfel
Ange felgräns för <u>relativfelet för</u> volymbestämningen

$$V(r) = \frac{4}{3}\pi r^3 \Rightarrow V'(r) = 4\pi r^2$$

Relativfelet ger

$$\Rightarrow |\delta r| \leq 0.01|r| \lessapprox 0.01|\hat{r}| \quad \text{(känner ej } r\text{)}$$

Approximativa felfortplantningsformeln ger:

$$\left|\frac{\delta V(r)}{\delta V(\hat{r})}\right| \lessapprox \left|\frac{V'(\hat{r})}{V(\hat{r})}\right| |\delta r| \lessapprox \left|\frac{4\pi\hat{r}^2}{(4/3)\pi\hat{r}^3}\right| \cdot |\hat{r}| = 0.03$$

<u>Svar</u>: 1% relativfel i indata ger 3% relativfel i utdata.

<u>Högre ordnings felapproximation</u>

$\delta f(x) = f'(\hat{x})\delta x$

Om $f'(\hat{x}) \approx 0$ är approx optimistisk

Om $f \in C^2(I)$ kan man härleda:

$$\delta f(x) = f'(x)\delta x + f''(x + \theta\delta x)\delta x^2 \cdot \frac{1}{2} \quad \theta \in [0,1]$$

Nu får vi $\delta f(x) = \frac{1}{2}f''(x + \theta\delta x)\delta x^2$ om $f(\hat{x}) \approx 0$

<u>Ex</u> $f(x) = x^2 + 2x + 1 \quad \hat{x} = -1 \approx x \quad \underline{o} \quad \left|\frac{\hat{x} - x}{x}\right| \leq 0.01$

$f'(x) = 2x + 2 \Rightarrow f'(\hat{x}) = 0$

Använd $\delta f(x) \approx f''(x)\delta x^2/2$

$$\Rightarrow \left|\frac{\delta f(x)}{f(\hat{x})}\right| \lessapprox \left|\frac{f''(\hat{x})}{f(\hat{x})}\right| \delta x^2/2 \leq 0{,}5 \cdot 10^{-4}$$

Felfortplantning i flera variabler

Låt $\mathbf{x} \in \mathbb{R}^n$ & $\hat{\mathbf{x}}$ vara en approx av $\mathbf{x}$.

Generalisering:

Absolut fel: $\delta \mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$

Relativ fel: $\dfrac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|}$

Felgränser: $\|\delta \mathbf{x}\| \leq c_1 \qquad \dfrac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq c_2$

Om $f \in C^1(I)$ där $I \subset \mathbb{R}^n$ kan felfortplantningen från indata $\hat{\mathbf{x}} = \mathbf{x} + \delta \mathbf{x}$ till fel i utdata $f(\hat{\mathbf{x}}) = f(\mathbf{x}) + \delta f(\mathbf{x})$ skattas:

$$\delta f(\mathbf{x}) = f(\hat{\mathbf{x}}) - f(\mathbf{x}) = \int_0^1 \frac{d}{ds} f(x + s\delta x)\, ds =$$

$$= \sum_{k=1}^{m} \frac{\partial f}{\partial x_k}(x + \theta \delta x)\, \delta x_k \qquad \theta \in [0,1]$$

Approximation om x är obekant:

$$\delta f(\mathbf{x}) \approx \nabla f(\hat{\mathbf{x}}) \cdot \delta \mathbf{x}.$$

Ex Härled formel för relativa felet för multiplikation

$f(x_1, x_2) = x_1 x_2$

$$|f(\hat{\mathbf{x}}) - f(\mathbf{x})| = |(x_1 + \delta x_1)(x_2 + \delta x_2) - x_1 x_2| =$$

$$= |x_1 \delta x_2 + x_2 \delta x_1 + \underbrace{\cancel{\delta x_1 \delta x_2}}_{\text{försumbar}}| \lessapprox |x_1 \delta x_2 + x_2 \delta x_1|$$

$$\Rightarrow \left| \frac{\delta f(x)}{f(x)} \right| \lessapprox \left| \frac{x_1 \delta x_2 + x_2 \delta x_1}{x_1 x_2} \right| \leq \left| \frac{\delta x_1}{x_1} \right| + \left| \frac{\delta x_2}{x_2} \right|$$

Konditionstal

Def (Relativa) KONDITIONSTALET definieras som:

kappa

$$\kappa(x) = \lim_{\delta \to 0^+} \max_{\|\delta x\| < \delta} \left(\frac{|\delta f(x)|}{|f(x)|} \Big/ \frac{\|\delta x\|}{\|x\|} \right) \approx$$

$$\approx \frac{\|\text{Relativfel utdata}\|}{\|\text{Relativfel indata}\|}$$ (största felet men för litet steg)

Tolkning

Relativt indatafel α kan ge relativ utdatafel $\kappa\alpha$

Ett problem med litet konditionstal sägs vara STABILT eller VÄLKONDITIONERAD. Tvärtom, stort $\kappa \to$ INSTABILT eller ILLAKONDITIONERAD.

Litet betyder ungefär $\kappa \in (0, 10)$, stort $\kappa \approx 10^8$

$$|\delta f(x)| \lesssim \|\nabla f(x)\| \, \|\delta x\|$$

$$\Rightarrow \kappa(x) \lesssim \|x\| \frac{\|\nabla f(x)\|}{|f(x)|}$$

Ex $f(x) = cx$ $\quad c$ konstant

$$\Rightarrow \kappa(x) \lesssim |x| \frac{|c|}{|cx|} = 1$$

d.v.s. multiplikation m konstant ger relativt indatafel = relativt utdatafel (OBS! absolut blir annorlunda)

Ex $f(x_1, x_2) = x_1 - x_2$ $\mathbf{x} = (1 \,;\, 1+10^{-100})$

Vill beräkna $f(\mathbf{x})$

$$\kappa(x) \approx \|x\| \frac{\|\nabla f(x)\|}{|f(x)|} \approx \sqrt{2}\, \frac{\|(1,-1)\|}{|1-(1+10^{-100})|} \approx 2 \cdot 10^{100}$$

STORT!

OBS: $x_1 - x_2$ är en instabil operation då $x_1 \approx x_2$

Senare i kursen: fungerar även för matriser! (matrisnormer)

Approximativa algoritmer

Den exakta funktionen f kan behöva ersättas med en approximativ funktion $\hat{f}$.

Def FRAMÅTFELET för $(f, \hat{f}, x)$ ges av $\hat{f}(x) - f(x)$, som beskriver approximativa algoritmens fel i utdata.

Antag att f är inverterbar nära x. Då finns ett indatavärde $\hat{x} = f^{-1}(\hat{f}(x))$ så att

$$f(\hat{x}) = f(f^{-1}(\hat{f}(x))) = \hat{f}(x)$$

Def BAKÅTFELET för $(f, \hat{f}, x)$ ges av

$$\hat{x} - x = f^{-1}(\hat{f}(x)) - x$$

och motsvarar det absoluta fel δx som ger felfortplantningen $f(\hat{x}) - f(x) = \hat{f}(x) - f(x)$

Ex $f(x) = e^x$ $\hat{f}(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3$ $x = 1$

$\hat{f}(1) = 8/3$ $f(1) = e^1$

Framåtfelet:

$$\hat{f}(1) - f(1) = \frac{8}{3} - e^1 = -0{,}051615\ldots$$

Bakåtfelet $f^{-1}(x) = \ln(x)$

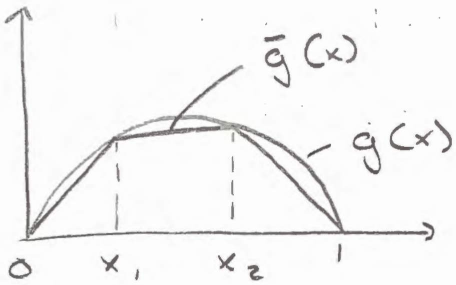
$\Rightarrow \hat{x} = f^{-1}(\hat{f}(1)) = \ln(8/3)$

Så bakåtfelet är

$$\hat{x} - x = \ln(8/3) - 1 = -0{,}019171\ldots$$

Numerisk algoritm & felfortplantning

Problem: Beräkna $I = \int_0^1 g(x)\,dx$



Förenklat problem:

$$\hat{I} = \int_0^1 \bar{g}(x)\,dx$$

Numerisk algoritm:

$$\hat{I} = \sum_{i=0}^{n} \left(g(x_i) + g(x_{i+1})\right) \frac{(x_{i+1} - x_i)}{2} \quad \leftarrow \text{approx}$$

Vi kan använda konditionstal även för algoritmer.

$$K(x) = \frac{\|\text{rel. utdatafel}\|}{\|\text{rel. indatafel}\|}$$

Def En numerisk algoritm $\hat{f} \approx f$ sägs vara STABIL för $x \neq 0$ om relativa bakåtfelets storlek

$$\frac{|f^{-1}(\hat{f}(x)) - x|}{\|x\|}$$

är litet.

Vi använder bakåtfel för att det ofta är enklare än framåtfel.

Betrakta $f(b) = A^{-1}b$ & $\hat{f}(b) = (A^{-1} + \varepsilon I)b$

Framåtanalys ger framåtstabiliteten:

$$\frac{\|\hat{f}(b) - f(b)\|}{\|f(b)\|}$$

Svårt om vi ej vet A^{-1}!

I bakåtanalysen använder vi $f^{-1}(y) = Ay$ vilket är lättare.

Datoraritmetik

Olika talformat:

- ◦ signed int32 (eller 64)
  - Alla heltal $[-2^{31}, 2^{31}-1]$ kan lagras
- ◦ single precision floating point (32 bitar)
  (eller doble precis..., 64 bitar)

I matlab är default double 64. Man kan skriva t.ex. `y = int64(pi);`

Flyttal

Ett flyttalssystem definieras av $F = (\beta, t, L, U)$

- ◦ $\beta \in N \setminus \{1\}$ är basen till systemet
- ◦ t är antalet siffror
- ◦ L, U är lägsta resp. högste värdet hos exponenten.

Varje flyttal x i F representeras:

$$x = \pm(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + \dots + d_{t-1}\beta^{1-t})\beta^e$$

där $1 \le d_0 \le \beta - 1$, $0 \le d_i \le \beta - 1$ för $1 \le i \le t-1$ & $L \le e \le U$

Den första faktorn kallas MANTISSA

Notation: Med mantisson $m = d_0.d_1 \dots d_{t-1}$ kan talet ovan skrivas $x = \pm m \cdot \beta^e$

Ex Flyttalssystemet IEEE-754 beskrivs av

$(\beta = 2,\ t = 53,\ L = -1022,\ U = 1023)$

t.ex. $x = 2.75 = (1 + 0 \cdot 2^{-1} + 2^{-2} + 2^{-3}) \cdot 2^1$ skrivs

$1.0110\dots0_2 \cdot 2^1$

Hur det lagras

0 1000... 01100000...

tecken

exponent (11 bitar)

mantissa uten $d_0 = 1$ (52 bitar)

d_0 alltid 1

Underflow, Overflow & Heltalsavrundning

$\min(\mathcal{F} \cap \mathbb{R}_+) = 1.0000 \cdot \beta^L := UFL$ ~ underflow level

motsvarande för OFL

För att konvertera/avrunda reella tal till flyttal!

närmsta tal i $\mathcal{F}$

$$fl(x) := \begin{cases} \arg\min_{y \in \mathcal{F}} (y-x) & \text{Om } UFL \leq |x| \leq OFL \\ \text{Inf} & \text{om } |x| > OFL \end{cases}$$

(förenklat)

Tal $|x| < UFL$ (t.ex. 0) ingår formellt ej i $\mathcal{F}$

I IEEE släpps normaliseringskravet, d_0 får vara 0.

Vi får tal under UFL genom att gradvis flytta bak 1 i mantissan, och 0 genom att ha 0 överallt.

För alla $UFL < |x| < OFL$:

$$\left| \frac{fl(x) - x}{x} \right| \leq 0.5 \cdot \beta^{1-t}$$

(För $|x| < UFL$ tappar vi denna precision!)

d.v.s. $fl(x)$ approximerar x m. $t-1$ signifikanta siffror.

Flyttalsaritmetik görs i ett utökat register för att bevara precision.

OBS generellt gäller inte att

$$fl(fl(a+b)+c) = fl(a+fl(b+c))$$

Utskiftning innebär att vi tappar noggrannhet då tal med olika storlek adderas.

Strategi för att undvika utskiftning: börja addera de minsta talen.

Kancellation innebär noggrannhetsförlust vid subtraktion av två nästan lika stora tal.

Strategi: skriv om problemet.

Ex: $\sqrt{x^2+1} - x = \ldots = \dfrac{1}{\sqrt{x^2+1} + x}$

risk för kancellation — bättre!

Framåtanalys av flyttalssystem

$f(x) = x^2 \qquad \hat{f} = fl(fl(x)^2)$

Låt oss verifiera $\left| \dfrac{\hat{f}(x) - f(x)}{f(x)} \right| = \mathcal{O}(\mu)$

$fl(x) = x(1+\delta)$ där $\delta \in [-\mu, \mu]$. där μ är maskintalet

$$\Rightarrow \hat{f}(x) = x^2(1+\delta_1)^2(1+\delta_2)$$

$$\Rightarrow \left| \frac{\hat{f}(x) - f(x)}{f(x)} \right| = \left| (1+\delta_1)^2(1+\delta_2) - 1 \right| = \mathcal{O}(\mu)$$

Bakåtanalys

Rep: $\hat{f}$ av f är stabil om bakåtfelet är litet.

d.v.s. $\left| \dfrac{f^{-1}(\hat{f}(x)) - x}{x} \right| = \mathcal{O}(\mu)$

Låt oss verifiera att $\hat{f}(x) = fl(fl(x)^2)$ är stabil.

$$f^{-1}(\hat{f}(x)) = x\left((1+\delta_1)\sqrt{(1+\delta_2)}\right) = x\left(1 + \delta_1 + \frac{\delta_2}{2} + \mathcal{O}(\delta^2)\right)$$

$$\Rightarrow \left| \frac{f^{-1}(\hat{f}(x)) - x}{x} \right| \lesssim \left| \frac{x(\delta_1 + \delta_2/2)}{x} \right| = \mathcal{O}(\mu)$$

OK!

Lana 200324

Numerisk lösning av ekvationer

Idag: Icke-linjära ekvationer i 1 variabel

$f: \mathbb{R} \to \mathbb{R}, \quad f(x) = 0$

Endast i undantagsfall har vi explicita lösningar. Vi behöver numeriska metoder.

Ex: $\frac{\pi}{2} - \arcsin x - x\sqrt{1-x^2} - 1.24 = 0$

Har en lösning $x^* = 0.16617..$

Def En lösning eller ROT x^* till $f(x) = 0$ kallas ENKELROT om $f'(x) \neq 0$, MULTIPELROT med MULTIPLICITET m om $f'(x^*) = f''(x^*) = \ldots = f^{(m-1)}(x^*) = 0$ men $f^{(m)}(x^*) \neq 0$

Ex: $f(x) = x^4 - 6x^3 + 2x^2 - 10x + 3 = (x-1)^3(x-3)$

har rot $x^* = 1$ med multiplicitet 3 och en enkelrot $x^* = 3$.

Iterationsmetoder

Första approximation x_0

Beräkna en talföljd $\{x_k\}_{k=1}^{\infty}$

Metoden konvergerar om $\lim_{k \to \infty} x_k = x^*$ där $f(x^*) = 0$

Konvergensordning q är största $q > 0$ s.a.

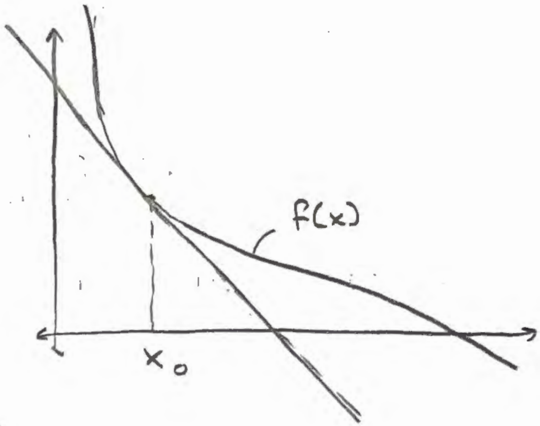
$$\lim_{k \to \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^q} = c < \infty$$

c kallas asymptotiska felkonstanten

Olika fall av konvergens

- $q = 1$: Linjär konvergens
- $q > 0$: Superlinjär konvergens
- $q = 2$: Kvadratisk konvergens

Newtons metod



Antag x_0 approximativ rot.

Taylorutveckla kring x_0

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

Har linjäriserat kring x_0

=> Lokalt linjär modell:

$$f(x_0) + f'(x_0)(x - x_0) = 0$$

$$\Rightarrow \boxed{x = x_0 - \frac{f(x_0)}{f'(x_0)}}$$

troligtvis en bättre approximation!

Upprepa med x_1 istället för x_0 ...

$$\Rightarrow \boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad k = 0, 1, \ldots}$$

Detta kallas Newtons metod.

Konvergens

Om x_0 är långt från x^* kan metoden divergera.

Antag att $f(x^*) = 0$, $f'(x^*) \neq 0$, $x_k \to x^*$

Taylorutveckla kring x_k:

$$0 = f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + \frac{1}{2} f''(\xi_k)(x^* - x_k)^2 \quad (*)$$

Newtoniterationen ger: $x_{k+1} = x_k - \dfrac{f(x_k)}{f'(x_k)}$

$$\Leftrightarrow \quad 0 = f'(x_k)(x_{k+1} - x_k) + f(x_k) \quad (**)$$

$\longrightarrow$

$\longrightarrow$ (*) − (**) ger:

$$0 = f'(x_k)(x^* - x_{k+1}) + \frac{1}{2} f''(\xi_k)(x^* - x_k)^2$$

$$\Rightarrow |x_{k+1} - x^*| = \frac{|f''(\xi_k)|}{2|f'(x_k)|} |x_k - x^*|^2$$

$$\Rightarrow \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} \xrightarrow{k \to \infty} c \quad \text{där}$$

$$c = \lim_{k \to \infty} \frac{|f''(\xi_k)|}{2|f'(x_k)|} = \frac{|f''(x^*)|}{2|f'(x^*)|} < \infty \quad \text{enl. ant.}$$

Alltså: Om metoden konvergerar mot en enkelrot så konvergerar den kvadratiskt.

Detta innebär att den ungefär fördubblar antalet korrekta decimaler per iteration.

Linjär konvergens vid multipelrot.

<u>Alternativ om x^* har multiplicitet m:</u>

$$x_{k+1} = x_k - m \frac{f(x_k)}{f'(x_k)}$$ ger kvadratisk konvergens

Nackdel: behöver känna till multiplicitet

Sekantmetoden

Newtons metod kräver uttryck för derivatan. Om uttryck saknas, approximera med differenskvot:

$$f'(x_k) \approx \frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}$$

Detta ger sekantmetoden:

$$\boxed{x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k)-f(x_{k-1})} \qquad k = 1, 2, \text{---}}$$

Notera att två startpunkter x_0 & x_1 krävs.

Konvergens vid enkelrot ger: $q = 1,618\ldots$ (superlinjär)

Sekantmetoden är alltså långsammare än Newton.

Fixpunktsiterationer

Skriv om $f(x)=0$ på formen $x=g(x)$ (addera t.ex. x på båda sidor). Vissa sätt är bättre än andra.

Iterera: $x_{k+1} = g(x_k)$ $k = 0, 1, \ldots$ med x_0 startapprox.

Om vi har konvergens $x_k \to x^* \Rightarrow x^* = g(x^*)$ d.v.s. x^* är fixpkt till g.

Lokal konvergens om $|g'(x^*)| < 1$. Mindre $|g'(x^*)|$ ger snabbare konvergens.

Ex: $g(x) = x - \frac{f(x)}{f'(x)}$ ger newtons metod.

$$g'(x) = 1-(f'(x)f'(x) - f(x)f''(x))/(f'(x))^2 =$$

$$= \frac{f(x)f''(x)}{(f'(x))^2} \Rightarrow g'(x^*) = \frac{\overset{0}{\cancel{f(x^*)}}\,f''(x^*)}{(f'(x^*))^2} = 0$$

Newtons metod "optimal", $|g'(x^*)| = 0$

Ex $x^3+x-1=0$ kan skrivas $x = \underbrace{1-x^3}_{g(x)}$

$\Rightarrow g'(x) = -3x^2$

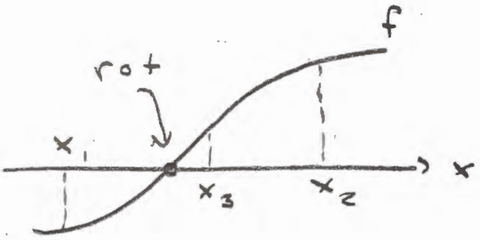
Beräkna...

$x^* = 0.6823 \Rightarrow |g'(x^*)| \approx 1,40 > 0$ ingen konvergens!

Men $x = \sqrt[3]{1-x}$ fungerar,

$g'(x^*) \approx 0,716 < 1$

Intervallhalveringsmetoden



Om $f(x_1)<0,\ f(x_2)>0$

Satsen om mellanliggande värde

$\Rightarrow \exists x^* \in (x_1,x_2)$ s.a. $f(x^*)=0$ (om f kont)

$$\left.\begin{array}{l} \text{Bilda } x_3 = \frac{1}{2}(x_1+x_2). \\ f(x_3)<0 \Rightarrow x^* \in (x_3,x_2) \\ f(x_3)>0 \Rightarrow x^* \in (x_1,x_3) \\ f(x_3)=0 \Rightarrow x^* = x_3 \end{array}\right\} \quad \text{Upprepa!}$$

Ger alltid konvergens men är långsam. Kan kombineras med andra metoder. Ger s.k. hybridmetoder.

Lösningsnoggrannhet

Ändligt antal iterationer och avrundningsfel ger att vi måste acceptera $f(\hat{x}) \neq 0$. Felet $\delta x = \hat{x} - x^*$ kan vi estimera m.h.a. Taylorutvecklingar:

$$f(\hat{x}) = f(x^*+\delta x) = \overset{0}{\cancel{f(x^*)}} + f'(x^*)\delta x + O(\delta x^2)$$

Små $\delta x \Rightarrow f(\hat{x}) = f'(\hat{x})\delta x \Rightarrow \delta x \approx \dfrac{f(\hat{x})}{f'(\hat{x})}$

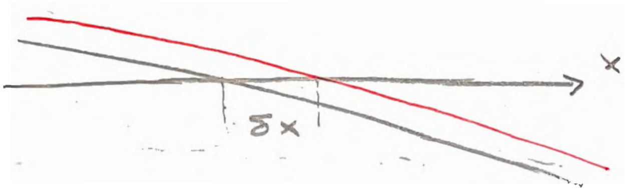
Detta är en metodoberoende feluppskattning:

$$|\hat{x} - x^*| \approx \left|\frac{f(\hat{x})}{f'(\hat{x})}\right|$$

⟶

$$\longrightarrow \qquad |\hat{x} - x^*| \approx \left| \frac{f(\hat{x})}{f'(\hat{x})} \right|$$

Vi ser att det är illa konditionerat om $|f'(\hat{x})|$ är litet.



flack kurva
=> liten höjdledsförändring ger stor sidledsförändring.

Om x^* rot med multiplicitet m:

Längre taylorutveckling i $\delta x^m = m! \dfrac{f(\hat{x})}{f^{(m)}(\hat{x})}$ istället

Lana 200325

Linjära rum

Kallas även vektorrum

Generalisering av $\mathbb{R}^n$

Vad elementen är spelar ingen roll, bara hur de samverkar.

Def

Ett LINJÄRT RUM eller VEKTORRUM är en mängd V, tillsamans med tal (skalärer) K (oftast $K = \mathbb{R}$, ibland $K = \mathbb{C}$) om följande axiom gäller:

① $\forall u, v \in V \quad \exists!\, u \oplus v \in V$

② $\forall u \in V \ \underline{o}\ \alpha \in K \quad \exists!\, \alpha \odot u \in V$

och följande räknelagar gäller:

③ $u \oplus v = v \oplus u \quad \forall u, v \in V$

④ $(u \oplus v) \oplus w = u \oplus (v \oplus w) \quad \forall u, v, w \in V$

⑤ $\exists 0 \in V : 0 \oplus u = u \oplus 0 = u \quad \forall u \in V$

⑥ $\forall u \in V \quad \exists -u \in V : u \oplus (-u) = (-u) \oplus u = 0$

⑦ $\alpha \odot (\beta \odot u) = (\alpha\beta) \odot u \quad \forall \alpha, \beta \in K \quad \forall u \in V$

⑧ $\alpha \odot (u \oplus v) = \alpha \odot u \oplus \alpha \odot v \quad \forall \alpha \in K \quad \forall u, v \in V$

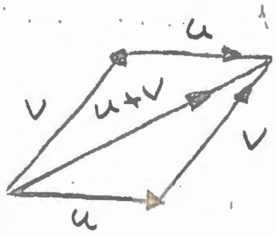
⑨ $(\alpha + \beta) \odot u = \alpha \odot u \oplus \beta \odot u \quad \forall \alpha, \beta \in K \quad \forall u \in V$

⑩ $\exists 1 \in K : 1 \odot u = u \quad \forall u \in V$

Notation / begrepp

Element i V kallas VEKTORER (även om de är t.ex. funktioner). Om $K = \mathbb{C}$ kallas V ett KOMPLEXT LINJÄRT RUM.

Ex: ∘ Geometriska vektorer



etc.

$u \oplus v = v \oplus u$ $\exists -u$ $\exists 0$

- $V = \mathbb{R}^n$ (euklidiska rummet)
- $\mathbf{x} \oplus \mathbf{y} = (x_1 + y_1, \dots, x_n + y_n)$
- $\alpha \odot \mathbf{x} = (\alpha x_1, \dots, \alpha x_n)$

etc

- m×n matriser

- Mängden av alla reellvärda funktioner

$(f \oplus g)(t) = f(t) + g(t)$

$(\alpha \cdot f)(t) = \alpha f(t)$

$\mathbb{0}(t) \equiv 0$

$(-f)(t) = -f(t)$

etc.

Räknelagarna följer här av de från $\mathbb{R}$ (se ovan)

- Andra ordningens differentialoperatorer verkande på C^2

- $V = \mathbb{R}^+$, men ha "gånger" som $\oplus$ o exponent som $\odot$
  (mer på storgruppsövningen)

Def

En delmängd $M \subseteq V$ kallas ett UNDERRUM av ett linjärt rum V om M är ett linjärt rum med avseende på samma operationer.

Sats

En icke-tom delmängd M av ett linjärt rum V är ett underrum av V omm

(*) $u, v \in M \Rightarrow u \oplus v \in M$, $u \in M, \alpha \in \mathbb{R} \Rightarrow \alpha \odot u \in M$

eller ekvivalent:

$u, v \in M \quad \alpha, \beta \in \mathbb{R} \Rightarrow \alpha \odot u \oplus \beta \odot v \in M$

Bevis

$\boxed{\Rightarrow}$ Vill visa: M underrum till $V \Rightarrow$ (*) gäller

Det följer direkt av axiom ① & ②

$\boxed{\Leftarrow}$ Antag att M uppfyller (*).

① & ② uppfylls direkt.

Låt $u \in M$ (M icke-tom) och $\alpha = 0$.

V är ett linjärt rum, så $\alpha \odot 0 = \mathbf{0}$

(*) ger därför: $\mathbf{0} \in M$

Tag $\alpha = -1$ och $u \in M$ godtyckligt.

V är ett linjärt rum, så $-u = (-1) \odot u$

(*) ger därför $-u \in M$

③ - ⑩ gäller för alla element i V, och således också för alla element i M.

Alla axiom är verifierade.

▨

Exempel på underrum

Underrum till 3 dim. åskådliga rummet ($\mathbb{R}^3$ utan koordinatsystem typ)

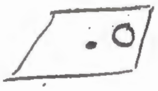
Identifiera punkter med ortsvektorer.

Underrum:

a) Origo (dim 0)

b) Räta linjer genom origo (dim 1)

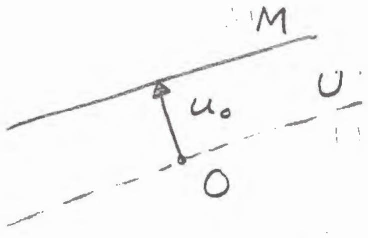
c) Plan genom origo (dim 2)



d) Hela åskådliga rummet (dim 3)

OBS: linjen/planet måste gå genom origo!

En godtycklig linje är en translation av en linje genom origo.



Def

En delmängd $M \subseteq V$ kallas AFFIN om det finns en vektor $u_0 \in V$ och ett underrum U av V så att $M = u_0 \oplus U = \{u_0 \oplus u : u \in U\}$

Fler exempel på underrum

Låt $V = \mathbb{R}^{n \times n}$ ($n \times n$ matriser). linjärt rum

Låt M vara alla anti-symmetriska $n \times n$ matriser (d.v.s. $A \in M \Rightarrow A^T = -A$)

$M \subseteq V$ icke-tom ty $O \in M$. (nollmatrisen)

$A, B \in M \Rightarrow (A \oplus B)^T = A^T \oplus B^T = (-A) \oplus (-B) = -(A \oplus B)$

Så $A, B \in M \Rightarrow A \oplus B \in M$, och uppenbart $\alpha \odot A \in M$

$\Rightarrow M$ är ett underrum

Ex $V = F([a,b])$ reellvärda funktioner på $[a,b]$

Har många intressanta underrum (funktionsrum)

a) $C([a,b])$, kontinuerliga funktioner på $[a,b]$

b) $C^k([a,b])$, k ggr kontinuerligt deriverbara funktioner på $[a,b]$.

c) P_n, polynom av grad <u>högst</u> $\underbrace{n}$ Annars ej slutet under + & .

d) $L^2((a,b))$, f s.a. $\exists \int_a^b f^2(t)\,dt < \infty$

För a)-c) är det enkelt att kontrollera

$$f, g \in M,\ \alpha \in \mathbb{R} \Rightarrow f+g \in M,\ \alpha f \in M$$

<u>Bevis för d)</u>

$0 \leq (a-b)^2 = a^2 - 2ab + b^2$ ger $ab \leq \frac{1}{2}(a^2+b^2)$

$\Rightarrow \underline{(a+b)^2} = a^2 + 2ab + b^2 \leq \underline{2a^2 + 2b^2}$

Om $f, g \in L^2((c,d))$

$$\int_c^d (f(t)+g(t))^2\,dt \leq 2\int_c^d f^2(t)\,dt + 2\int_c^d g^2(t)\,dt < \infty$$

$\Rightarrow f+g \in L^2((c,d))$

Om $\alpha \in \mathbb{R}$, $f \in L^2((c,d))$

$$\int_c^d (\alpha f(t))^2\,dt = \alpha^2 \int_c^d f^2(t)\,dt < \infty$$

$\Rightarrow \alpha f \in L^2((c,d))$

Så $L^2((c,d))$ underrum till $F((c,d))$

▨

Ex

Lösningar till homogena differentialekvationer kan också vara underrum.

Låt V vara funktioner $f(t,x)$ med kont. andraderivator för $t \in (0,1)$, $x \in (0,1)$. Då är V ett linjärt rum.

Låt $M = \left\{ f : \frac{\partial^2 f}{\partial t^2} - \frac{\partial^2 f}{\partial x^2} = 0 \right\}$

M är ett underrum ty:

f=0 är en lösning, så M är icke-tom.

Om $f,g \in M$ & $\alpha,\beta \in \mathbb{R}$ Låt $\alpha f + \beta g =: h$

$$\frac{\partial^2 h}{\partial t^2} - \frac{\partial^2 h}{\partial x^2} = \alpha \frac{\partial^2 f}{\partial t^2} + \beta \frac{\partial^2 f}{\partial x^2} + \alpha \frac{\partial^2 f}{\partial x^2} - \beta \frac{\partial^2 f}{\partial x^2} = 0$$

$\Rightarrow h \in M$

Ex

Låt $V \subset C([0,\infty))$ sådana att $f \in V$ om

$\int_0^\infty f^2(x) e^{-x} dx < \infty$ (viktat L^2 rum)

V underrum ty:

Om $f,g \in V$, $\alpha \in \mathbb{R}$

$$\int_0^\infty (f(x)+g(x))^2 e^{-x} dx = \int_0^\infty 2(f^2(x)+g^2(x)) e^{-x} dx =$$

$$= 2\int_0^\infty f^2(x) e^{-x} dx + 2\int_0^\infty g^2(x) e^{-x} dx < \infty$$

$$\int_0^\infty (\alpha f(x))^2 dx = \alpha^2 \int_0^\infty f^2(x) dx < \infty$$

$\Rightarrow f+g \in V$, $\alpha f \in V$.

Linjär algebra och numerisk analys   200320

Linjära avbildningar

Skriver fr.o.m. nu $u+v$ & αu istället för $u \oplus v$ och $\alpha \odot u$

Rep: Sats [1.1.]

∘ $\emptyset \neq M \subset V$ V linj. rum

M underrum omm $u, v \in M$, $\alpha, \beta \in \mathbb{R} \Rightarrow \alpha u + \beta v \in M$

Linjära avbildningar

Def: Låt U, V vara reella linjära rum.

En avbildning (funktion/transformation) $F: U \to V$ kallas LINJÄR om

$F(u+v) = F(u) + F(v)$ $\forall u, v \in U$ och

$F(\alpha u) = \alpha F(u)$ $\forall \alpha \in \mathbb{R}$, $u \in U$

eller ekvivalent

$F(\alpha u + \beta v) = \alpha F(u) + \beta F(v)$ $\forall \alpha, \beta \in \mathbb{R}$, $u, v \in U$

Om $U = V$ säger vi att F är en linjär avbildning på V.

Om u & v är i $\mathbb{R}^n$ är F linjär om den är på formen $F(x) = Ax$ där A är en $n \times m$-matris.

OBS: $\alpha = 0$ ger $F(0) = 0$ (bra kontroll)

Exempel på linjära avbildningar av geo. vek.
spegling, vridning, skalning, projektion

Ex Låt $U = \mathbb{R}^n$, $V = \mathbb{R}^m$, A $m \times n$ matris
$\Rightarrow F(x) = Ax$ linjär avbildning

Ex Avbildningar på funktionsrum kallas operatorer
Låt $U = C'([0,1])$, $V = C([0,1])$ och $F(f) = Df = f'$
$F: U \to V$ är linjär avbildning ty

$$F(\alpha f + \beta g) = D(\alpha f + \beta g) = \alpha Df + \beta Dg = \alpha F(f) + \beta F(g)$$
$$\forall f, g \in U, \ \alpha, \beta \in \mathbb{R}$$

Om $p(t)$ är ett polynom av grad n så är
$F(f) = p(D)f = (a_n D^n + \dots + a_0 D^0)f = a_n f^n + \dots + a_0 f$
en linjär avbildning $F: C^n(\mathbb{R}) \to C(\mathbb{R})$. kallas
linjär differentialoperator. Koefficienterna
kan bero på x men inte f.

Ex Integraloperator:
$F(f) = \int_a^x f(t)dt$ ger $F: C^0([a,b]) \to C'([a,b])$,
en linjär avbildning. (Kontroll som övning)

Def: Låt $F: U \to V$ vara linjär avbildning.
NOLLRUMMET för F är $N(F) = \{u \in U \mid F(u) = 0\}$
VÄRDERUMMET för F är $V(F) = \{F(u) \mid u \in U\}$

$N(F)$ är allt som mappar på 0
$V(F)$ är allt F mappar på.

Kontroll av att $N(F)$ är linjärt:

$F(0)=0 \Rightarrow N(F) \neq \emptyset$

Låt $u_1, u_2 \in N(F)$, $\alpha, \beta \in \mathbb{R}$

$$F(\alpha u_1 + \beta u_2) \underset{\substack{\uparrow \\ F \text{ linjärt}}}{=} \alpha F(u_1) + \beta F(u_2) \underset{\substack{\uparrow \\ u_1, u_2 \in N(F)}}{=} 0$$

$\Rightarrow \alpha u_1 + \beta u_2 \in N(F)$ OK!

$N(F)$ är därmed ett underrum till U.

Kontroll av att $V(F)$ är linjärt.

$F(0)=0$ ty F linjär $\Rightarrow V(F) \neq \emptyset$

Låt $v_1, v_2 \in V(F)$, $\alpha, \beta \in \mathbb{R}$

$\Rightarrow \exists u_1, u_2 \in U$ s.a. $F(u_1)=v_1$, $F(u_2)=v_2$

$$\alpha v_1 + \beta v_1 = \alpha F(u_1) + \beta F(u_2) \underset{\substack{\uparrow \\ F \text{ linj}}}{=} F(\underbrace{\alpha u_1 + \beta u_2}_{\in U})$$

Så $\alpha v_1 + \beta v_1 \in V(F)$ OK!

$V(F)$ är därmed ett underrum till V

Ett linjärt ekvationssystem, en linjär differentialekvation eller system av sådana m. fl. kan uttryckas $\boxed{F(u)=v}^{(*)}$ där $F: U \to U$ är en linjär avbildning, u är sökt och v känd.

Sats [1.4.] Antag att u_p är en känd lösning till $(*)$ (partikulärlösning). Då är $u \in U$ en lösning till $(*)$ omm $u = u_p + u_h$ där $u_h \in N(F)$

(Lösningsrummet är ett affint rum)

Bevis Då $F(u_p)=v$ gäller

$$F(u)=v=F(u_p) \underset{\substack{\uparrow \\ F \text{ linj}}}{\Leftrightarrow} F(u-u_p)=0 \Leftrightarrow \underbrace{u-u_p}_{=: u_h} \in N(F) \quad ▨$$

Lösningarna till $F(u)=v$ är det affina rummet $u_p + N(F)$

Linjärt (o)beroende bas och dimension

Def Låt $v_1, v_2, \ldots, v_p \in V$, V linjärt rum. $\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_p v_p$ med $\alpha_j \in \mathbb{R}$ kallas en LINJÄRKOMBINATION av $v_1, \ldots, v_p$.
$\text{Span}\{v_1, \ldots, v_p\} = \text{Span}\{v_j\}_{j=1}^{p}$ är mängden av alla linjärkombinationer av $v_1, \ldots, v_p$.
Kallas LINJÄRA HÖLJET av $v_1, \ldots, v_p$.

Övning: Visa att $\text{span}\{v_j\}_{j=1}^{p}$ är ett underrum av V.

Def Låt $u_1, \ldots, u_n \in V$, V linjärt rum. $u_1, \ldots, u_n$ sägs SPÄNNA UPP V om $V = \text{Span}\{u_i\}_{i=1}^{n}$ (varje element i V är en linjärkombination av $u_1, \ldots, u_n$.)

Def $u_1, \ldots, u_n \in V$ sägs vara LINJÄRT BEROENDE om

$$(*) \quad \lambda_1 u_1 + \ldots + \lambda_n u_n = 0$$

har en icke-trivial lösning. (Något $\lambda_i \neq 0$).
Om $(*)$ endast har trivial lösning sägs $u_1, \ldots, u_n$ vara LINJÄRT OBEROENDE.

OBS: Om någon $u_i = 0$ är $u_1, \ldots, u_n$ alltid linjärt beroende (sätt $\lambda_i = 1$ & resterande $\lambda_j = 0$ så är $(*)$ uppfylld).

Lemma [1.1]

Om ingen av $u_1, \dots, u_n$ är 0 så är följande påståenden ekvivalenta:

(i) De är linjärt beroende

(ii) Någon vektor u_i med $i \geq 2$ är en linjärkombination av de föregående.

(iii) Någon vektor u_i är lin. komb av de övriga.

Bevisidé: (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i)

(Bevis finns i boken)

Lemma [1.2]

Om $u_1, \dots, u_m \in V$ är linjärt oberoende & $v \in V$ men $v \notin \mathrm{Span}\{u_i\}_{i=1}^{m}$ så är $u_1, \dots, u_m, v$ linjärt oberoende.

Bevis Låt $U = \mathrm{Span}\{u_1, \dots, u_m\}$.
$0 \in U$ och $u_1, \dots, u_m$ är ober. så ingen av $u_1, \dots, u_m, v$ är 0. Eftersom $v \notin U$ gäller Lemma 1.1. *inte* för något i. Alltså är $u_1, \dots, u_m, v$ linjärt oberoende.

▨

Def

En uppsättning vektorer $u_1, \dots, u_n \in V$ som är linjärt oberoende och spänner upp V kallas en BAS för V.

Def

Det maximala antalet n av linjärt oberoende vektorer i ett linjärt rum V kallas DIMENSIONEN av V, skrivs

$\dim V = n$.

Om det inte finns ett maximalt antal sägs V vara OÄNDLIGDIMENSIONELLT.

Om V bara innehåller nollelementet är $\dim V = 0$. (Enda 0-dim vektorrummet).

Sats [1.5]

Antag $\dim V = \overset{\neq \infty}{n} > 0$. Då finns en uppsättning av n linj. ober. vektorer i V. Varje sådan uppsättning är en bas för V.

Bevis Låt $u_1, \ldots, u_n$ vara godt. uppsättning n linj. ober. vektorer i V.

Definitionen av dim ger existensen av minst en sådan uppsättning.

Låt $v \in V$ vara en godt. vektor.

Fall 1: $v \notin \text{span}\{u_1, \ldots, u_n\}$

[Lma 1.2] $\Rightarrow$ $u_1, \ldots, u_n, v$ linj. ober ← $n+1$ st

Detta motsäger att n är maximala antalet linj. ober vektorer i V.

Så **Fall 2**: $v \in \text{span}\{u_1, \ldots, u_n\}$ måste gälla. Detta för $\forall v \in V$, ty v var godt.

$\Rightarrow V = \text{span}\{u_1, \ldots, u_n\} \Rightarrow u_1, \ldots, u_n$ bas för V. $\boxtimes$

Sats [1.6]

Antag $V = \text{span}\{u_1, \dots, u_n\}$. Då är varje uppsättning med fler än n vektorer linjärt beroende.

(Se bevis från första linalgkursen.)

Sats [1.7]

Antag V ändligdimensionellt.

Alla baser har då $n = \dim V$ element.

Bevisidé:

Antag $V = \text{span}\{u_1, \dots u_m\}$ Om u_i är linjärkomb av de andra kan vi ta bort den. $u_1, \dots, u_{i-1}, u_{i+1}, \dots u_m$ spänner fortfarande upp V. Fortsätt till man har en linj. ober mängd. Vi har nu en bas med $n = \dim V$ ingående vektorer.

Sats [1.9]

Antag $\dim V = n > 0$ och $u_1, \dots, u_m \in V$ med $m < n$ är linjärt oberoende. Då finns vektorer $u_{m+1}, \dots, u_n$ så att $u_1, \dots, u_n$ är en bas för V.

Bevis: Låt $U_m = \text{Span}\{u_1, \dots, u_m\}$. $\dim U_m < n$. Så U_m är inte hela V. Då finns $u_{m+1} \in V$ så att $u_{m+1} \notin U_m$. Lemma 1.2 ger $u_1, \dots, u_m, u_{m+1}$ är linj. ober. Upprepa tills vi har n vektorer. De utgör en bas enl. Sats 1.5. ▨

Lana 200331

Dimension för underrum, direkt summa, rang

Sats [1.10]

Antag att V är ändlig dimensionellt och U är ett underrum till V. Då är $\dim U \leq \dim V$. Om $\dim U = \dim V$ är $U = V$.

Bevis

Låt $n = \dim V$. U kan inte ha fler än n linjärt oberoende vektorer, ty V kan inte ha det. Alltså $\dim U \leq \dim V$

Om $\dim U = n$ så finns en bas med n vektorer Detta är även en bas för V, ty varje uppsättning med n linj. ober vektorer utgör en bas enl. Sats 1.5. Så $U = V = \text{span}\{\text{bas}\}$. ▨

Direkt summa

Def. Låt U_1 & U_2 vara två underrum till V. då är $U_1 + U_2 := \{u_1 + u_2 \mid u_1 \in U_1, u_2 \in U_2\}$. Summan är en DIREKT SUMMA om $u \in U_1 + U_2$ har en entydig uppdelning $u = u_1 + u_2$ med $u_1 \in U_1$ och $u_2 \in U_2$. Vi skriver $U_1 \oplus U_2$

Lemma [1.4]

$U_1 + U_2$ är en direkt summa omm

$U_1 \cap U_2 = \{0\}$

Bevisidé

Om $v \in U_1 \cap U_2$ ger $\underset{\in U_1}{u_1} + \underset{\in U_2}{u_2} = \underbrace{(u_1+v)}_{\in U_1} + \underbrace{(u_2-v)}_{\in U_2}$

Två olika uppdelningar

Fullständigt bevis i boken.

Sats [1.11]

Om V ändligdim och U underrum till V så finns ett underrum W så att $V = U \oplus W$.

$\dim W = \dim V - \dim U$

W kallas komplementet till U i V.

Bevisidé: Använd sats 1.9 med $u_1, \ldots, u_m$ bas för U och $u_{m+1}, \ldots, u_n$ bas för W.

Dimension i samband med matriser

Sats [1.14] Dimensionssatsen

Om A är en $m \times n$-matris så gäller

$$\dim N(A) + \dim V(A) = n$$

Bevis

Antag $0 < p = \dim N(A) < n$ och $e_1, \dots, e_p$ är en bas för $N(A) \subseteq \mathbb{R}^n$.

Sats 1.9 ger vektorer $e_{p+1}, \dots, e_n$ så att $e_1, \dots, e_n$ är en bas för $\mathbb{R}^n$.

Ett godt. $x \in \mathbb{R}^n$ kan skrivas

$$x = \sum_{j=1}^{n} \lambda_j e_j \Rightarrow Ax = \sum_{j=1}^{n} \lambda_j A e_j =$$

$$= \sum_{j=1}^{p} \lambda_j \underbrace{A e_j}_{=0} + \sum_{j=p+1}^{n} \lambda_j A e_j$$

Så $V(A)$ spänns upp av $Ae_{p+1}, \dots, Ae_j$

Är dessa linj. ober? Antag

$$(*) \quad \sum_{j=p+1}^{n} \alpha_j A e_j = 0 \Leftrightarrow \underbrace{Ay = 0}_{\Rightarrow y \in N(A)} \quad \text{med} \quad y = \sum_{j=p+1}^{n} \alpha_j e_j$$

$\Rightarrow y$ kan skrivas $\sum_{j=1}^{p} \beta_j e_j - \sum_{j=1}^{p} \alpha_j e_j$ (Låt $\beta_j = -\alpha_j$)

Så $0 = y - y = \sum_{j=p+1}^{n} \alpha_j e_j - \sum_{j=1}^{p} -\alpha_j e_j = \sum_{j=1}^{n} \alpha_j e_j$

men $e_1, \dots, e_n$ är linj. ober! $\Rightarrow \alpha_j = 0 \ \forall j$

Så $(*)$ har bara trivial lösning.

$\Rightarrow Ae_{p+1}, \dots, Ae_n$ är linj ober $\Rightarrow$ bas för $V(A)$

$\longrightarrow$

$\dim V(A) = n - p \Rightarrow \dim V(A) + \dim N(A) = n$

Om $p = 0$ blir $N(A) = \{0\}$, samma argument ger $\dim V(A) = n$

Om $p = n$ blir $N(A) = \mathbb{R}^n$ och $V(A) = \{0\}$

$\Rightarrow \dim V(A) = 0$

☒

Dimensioner och matrisoperationer

$F : \mathbb{R}^n \to \mathbb{R}^m \quad F(\mathbf{x}) = A\mathbf{x} \quad A \in \mathbb{R}^{m \times n}$

Kolonnrangen är dim värderummet =

= kolonnrummet, d.v.s. rummet som spänns upp av kolonnerna

Radrangen är dim radrummet

Sats [1.15] Rangsatsen

För godt. $m \times n$-matris A är

kolonnrang = radrang

Värdet kallas rang(A), rangen av A.

Bevis

Fullständig Gausselimination transformerar A m.h.a. elementära radoperationer till någon matris A'. Låt $j_1, \ldots, j_r$ vara pivotkolonnerna (r st) T.ex.

$$A' = \underbrace{\begin{pmatrix} 0 & 1 & * & 0 & \cdots & 0 & * \\ \vdots & 0 & 0 & 1 & & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & * \\ \vdots & \vdots & \vdots & \vdots & & \vdots & 0 \\ 0 & 0 & 0 & 0 & & 0 & 0 \end{pmatrix}}_{n} \begin{array}{l} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} r \\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\} m - r \end{array}$$

$\longrightarrow$

$Ax = 0 \iff A'x = 0$ så $N(A) = N(A')$

Radrum oförändrat under elem. radop.

$\Rightarrow$ radrang(A) = radrang$(A') \underset{\substack{\uparrow \\ r \text{ linj. ober rader}}}{=} r$

Kolonnrang$(A) \overset{\substack{\text{dim. satsen} \\ \downarrow}}{=} \dim V(A) = n - \dim N(A) =$

$= n - \dim N(A') \underset{\substack{\uparrow \\ \text{dimsats}}}{=} \dim V(A') \underset{\uparrow}{=} r$

ty kolonnerna med nr $j_1, \dots, j_r$ är linj. ober

$\Rightarrow$ Kolonnrang$(A) = r =$ Radrang(A) ▨

<u>OBS</u>: $Ax = 0 \iff A'x = 0$

Är samma sak som (*)

$x_1 a_1 + \dots + x_n a_n = 0 \iff x_1 a_1' + \dots + x_n a_n' = 0$

där a_i är kolonner i A & a_i' kolonner i A'

Kolonnerna med nr $j_1, \dots, j_r$ i A' är uppenbart linj. ober $\Rightarrow x_j = 0 \ \forall j$ - enda lösning till (*)

$\Rightarrow$ kolonnerna med nr $j_1, \dots, j_r$ i A är linj. ober $\Rightarrow$ dessa kolonner utgör en bas för $V(A)$.

Sats [1.18]

Låt A vara en $n \times n$-matris. Då är följande påståenden ekvivalenta:

(i) $A\mathbb{x} = \mathbb{0}$ har endast lösningen $\mathbb{x} = \mathbb{0}$

$(N(A) = \{0\},\ \dim N(A) = 0)$

(ii) $A\mathbb{x} = \mathbb{b}$ lösbar för alla $\mathbb{b} \in \mathbb{R}^n$

$(V(A) = \mathbb{R}^n,\ \dim V(A) = n)$

(iii) $A\mathbb{x} = \mathbb{b}$ entydigt lösbar $\forall \mathbb{b} \in \mathbb{R}^n$

(iv) rang $A = n$

(v) Kolonnerna i A är linjärt oberoende

(vi) A är inverterbar

(vii) $\det A \neq 0$

Koordinater och basbyten

Låt $\mathbb{e} = \{e_1, \dots, e_n\}$ vara en bas för V.

(ordningen spelar roll)

Varje vektor $u \in V$ kan skrivas på formen

$$u = x_1 e_1 + \dots + x_n e_n = \sum_{j=1}^{n} x_j e_j$$

Talen $x_1, \dots, x_n$ är entydiga ty om också

$$u = x_1' e_1 + \dots + x_n' e_n$$

$$\Rightarrow 0 = u - u = (x_1 - x_1')e_1 + \dots + (x_n - x_n')e_n$$

Eftersom $\{e_1, \dots, e_n\}$ är linj ober är $x_i = x_i'$ $\forall i$

Varje vektor $u \in V$ motsvarar vektorn

$$\mathbb{x} = [u]_{\mathbb{e}} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

$\mathbb{x}$ kallas koordinatvektor och talen $x_1, \dots, x_n$ kallas koordinater för u i basen $\mathbb{e}$

Linjär kombinationer i V motsvarar linjärkombinationer i $\mathbb{R}^n$. ty:

Om $u, v \in V$, $\mathbb{x} = [u]_{\mathbb{e}}$, $\mathbb{y} = [v]_{\mathbb{e}}$

$$u + v = \sum_{i=1}^{n} x_i e_i + \sum_{i=1}^{n} y_i e_i = \sum_{i=1}^{n} (x_i + y_i) e_i$$

$$\Rightarrow \quad [u+v]_{\mathbb{e}} = x + y = [u]_{\mathbb{e}} + [v]_{\mathbb{e}}$$

$$\alpha u = \alpha \sum_{i=1}^{n} x_i e_i = \sum_{i=1}^{n} (\alpha x) e_i$$

$$\Rightarrow \quad [\alpha u]_{\mathbb{e}} = \alpha x = \alpha [u]_{\mathbb{e}}$$

Generellt:

$$\boxed{(*) \quad [\alpha_1 u_1 + \dots + \alpha_k u_k]_{\mathbb{e}} = \alpha_1 [u_1]_{\mathbb{e}} + \dots + \alpha_k [u_k]_{\mathbb{e}}}$$

Basbyten

Låt $\mathbb{e} = \{e_1, \dots, e_n\}$ och $\mathbb{e}' = \{e_1', \dots, e_n'\}$ vara två baser för V.

$$x = [u]_{\mathbb{e}} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad x' = [u]_{\mathbb{e}'} = \begin{pmatrix} x_1' \\ \vdots \\ x_n' \end{pmatrix}$$

$$\Rightarrow \quad u = \sum_{i=1}^{n} x_i e_i = \sum_{i=1}^{n} x_i' e_i'$$

$$(*) \Rightarrow x = [u]_{\mathbb{e}} = \sum_{i=1}^{n} x_i' [e_i']_{\mathbb{e}} = \underbrace{\begin{pmatrix} | & & | \\ [e_1']_{\mathbb{e}} & \cdots & [e_n']_{\mathbb{e}} \\ | & & | \end{pmatrix}}_{T,\ n \times n \text{ matris}} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\Rightarrow \quad x = T x'$$

⟶

$x = Tx'$ där kolonnerna i matrisen T är koordinatvektorerna för $e_1' \ldots e_n'$ i basen $\mathbf{e}$. T kallas transformations-matrisen. Om $T = (t_{ij})$ så är

$e_j' = \sum_{i=1}^{n} t_{ij} e_i$. $e_1', \ldots, e_n'$ linj ober i V

=> kolonner i T linj. ober i $\mathbb{R}^n$.

=> T är inverterbar.

$$x = Tx' \Leftrightarrow x' = T^{-1}x$$

Byt roll på $\mathbf{e}$ & $\mathbf{e}'$ => $T^{-1} = \begin{pmatrix} | & & | \\ [e_1]_{\mathbf{e}'} & \cdots & [e_n]_{\mathbf{e}'} \\ | & & | \end{pmatrix}$

Tydligare skrivsätt:

$$x = T_{\mathbf{e} \leftarrow \mathbf{e}'} x' \Leftrightarrow x' = T_{\mathbf{e}' \leftarrow \mathbf{e}} x$$

$$\Rightarrow \quad (T_{\mathbf{e} \leftarrow \mathbf{e}'})^{-1} = T_{\mathbf{e}' \leftarrow \mathbf{e}}$$

Lana 200403

Basbyten (Fortsättning)

$$[\alpha u_1 + \dots + \alpha_k u_k]_e = \alpha_1 [u_1]_e + \dots + \alpha_k [u_k]_e \quad (*)$$

$$T = \begin{pmatrix} | & & | \\ [e'_1]_e & \cdots & [e'_n]_e \\ | & & | \end{pmatrix}, \quad T = (t_{ij}),$$

$$\Rightarrow e'_j = \sum_{i=1}^{n} t_{ij} e_i \quad (**)$$

e o e' baser för vektorrum V.

Om V är ett underrum till U.

Låt f vara någon bas för U, och vi har e_i uttryckt i f. Hur uttrycker vi e'_i i f?

$$[e'_i]_f \overset{(**),(*)}{=} \sum_{i=1}^{n} t_{ij}[e_i]_f = \begin{pmatrix} [e_1]_f & \cdots & [e_n]_f \\ | & & | \end{pmatrix} \overbrace{\begin{pmatrix} t_{1j} \\ \vdots \\ t_{nj} \end{pmatrix}}^{[e'_j]_e}$$

$$\Rightarrow \boxed{\left([e'_1]_f \cdots [e'_n]_f\right) = \left([e_1]_f \cdots [e_n]_f\right) T} \quad (***)$$

Ex: [1.21] Låt

$$e_1 = \begin{pmatrix} 3 \\ 0 \\ -1 \end{pmatrix} \quad e_2 = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \quad e'_1 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \quad e'_2 = \begin{pmatrix} 0 \\ 3 \\ 2 \end{pmatrix}$$

vara två baser för planet $x - 2y + 3z = 0$ i $\mathbb{R}^3$ (uttryckta i standardbasen för $\mathbb{R}^3$). Vad är T?

$$(***) \Rightarrow \underset{[e'_i]_f}{\begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 0 & 2 \end{pmatrix}} = \underset{[e_i]_f}{\begin{pmatrix} 3 & 1 \\ 0 & -1 \\ -1 & -1 \end{pmatrix}} T \qquad \text{Obs: } T \ 2 \times 2$$

Stryk t.ex. första raden $(2 \cdot (2{:}a) - 3 \cdot (3{:}e))$

$$\Rightarrow \begin{pmatrix} 1 & 3 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix} T \iff T = \begin{pmatrix} 1 & 1 \\ -1 & 3 \end{pmatrix}$$

Skalärprodukt

Def:

En SKALÄRPRODUKT eller INRE PRODUKT i ett reellt linjärt rum V är en reellvärd funktion $\langle u,v\rangle$ av $u,v\in V$ med egenskaperna:

(i) $\langle u,v\rangle = \langle v,u\rangle \quad \forall u,v\in V$

(ii) $\langle \alpha u,v\rangle = \alpha\langle u,v\rangle \quad \forall u,v\in V,\ \alpha\in\mathbb{R}$

(iii) $\langle u_1+u_2,v\rangle = \langle u_1,v\rangle + \langle u_2,v\rangle \quad \forall u_1,u_2,v\in V$

(iv) $\langle u,u\rangle \geq 0 \quad \forall u\in V$ och $\langle u,u\rangle = 0 \Leftrightarrow u = \mathbb{0}$

Ex: Standardskalärprodukten i $\mathbb{R}^n$

$$\langle x,y\rangle = x\cdot y = x_1y_1+\dots+x_ny_n = x^Ty$$

(iv): $\langle x,x\rangle = x_1^2+\dots+x_n^2 \geq 0$

$$\langle x,x\rangle = 0 \Leftrightarrow x_1 = \dots = x_n = 0$$

Ex: Alternativ skalärprodukt i $\mathbb{R}^2$, $x,y\in\mathbb{R}^2$

$$\langle x,y\rangle = x_1y_1+x_1y_2+x_2y_1+2x_2y_2$$

(i), (ii) & (iii) är enkla att kontrollera

(iv): $\langle x,x\rangle = x_1^2+2x_1x_2+2x_2^2 = (x_1+x_2)^2+x_2^2$

Alltid ≥ 0; likhet om $x_1+x_2=0$ & $x_2=0$

$\Leftrightarrow x_1=x_2=0$ OK!

Ex: Låt A symmetrisk $n \times n$ matris som är positivt definit, d.v.s. $x^TAx \ge 0$ $\forall x \in \mathbb{R}^n$, med likhet omm $x = 0$. Låt $\langle x, y \rangle = x^TAy$.

Detta är en inre produkt ty:

(i) $\langle x, y \rangle = x^TAy = (x^TAy)^T = y^TA^Tx =$

$$= y^TAx = \langle y, x \rangle$$

egenskap (ii) & (iii) som övning

(iv) $\langle x, x \rangle = x^TAx \ge 0$ med likhet omm $x = 0$

$\Leftrightarrow$ A positivt definit.

OBS: förra exemplet är specialfallet $A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$

Ex: Om $V = C([a,b])$. Låt $\langle f, g \rangle = \int_a^b f(t)g(t)dt$

Detta är en skalärprodukt ty:

(i), (ii), (iii) följer av räkneregler för integraler.

(iv) $\langle f, f \rangle = \int_a^b \underbrace{f^2(t)}_{\ge 0} dt \ge 0$

$\int_a^b f^2(t)dt = 0 \Leftrightarrow f(t) \equiv 0$ på $[a,b]$ ty f kont.

* * *

Idé bakom $\langle f, g \rangle = \int_a^b f(t)g(t)dt$

Låt $a = 0$ & $b = 1$

Vi vill ha något som liknar $\mathbb{R}^n$ skalärprodukt:

$$x_1y_1 + x_2y_2 + \dots \approx \sum_{i=1}^{n} f\left(\tfrac{i}{n}\right) g\left(\tfrac{i}{n}\right)$$

Men vi måste normalisera för n om vi ska ta många värden!

$$\frac{1}{n} \sum_{i=1}^{n} f\left(\tfrac{i}{n}\right) g\left(\tfrac{i}{n}\right) \quad \text{(Riemannsumma!)}$$

Låt $n \to \infty$: $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$

Def

Om V har en skalärprodukt $\langle\ ,\ \rangle$ så definierar vi LÄNGDEN eller NORMEN av en $u \in V$ som

$$\|u\| = \sqrt{\langle u,u\rangle} \qquad \text{(ok p.g.a. (iv))}$$

AVSTÅNDET mellan u och $v \in V$ är $\|u-v\|$.

OBS: Man kan ha normer som inte kommer från en skalärprodukt, till exempel $\|x\| = |x_1| + |x_2|$ i $\mathbb{R}^2$.

Egenskaper (För normer från skalärprodukt)

$$\|u+v\|^2 = \langle u+v, u+v\rangle = \langle u, u+v\rangle + \langle v, u+v\rangle =$$

$$= \langle u,u\rangle + \langle u,v\rangle + \langle v,u\rangle + \langle v,v\rangle =$$

$$= \|u\|^2 + 2\langle u,v\rangle + \|v\|^2$$

$$\Rightarrow \boxed{\langle u,v\rangle = \frac{\|u+v\|^2 - \|u\|^2 - \|v\|^2}{2}}$$

$$\|\alpha u\| = \sqrt{\langle \alpha u, \alpha u\rangle} = |\alpha|\sqrt{\langle u,u\rangle} = |\alpha|\cdot\|u\|$$

$$\boxed{\|\alpha u\| = |\alpha|\cdot\|u\|}$$

Sats [2.1] Cauchy-Schwarz olikhet
$|\langle u,v\rangle| \leq \|u\|\cdot\|v\| \quad \forall u,v \in V$ med likhet omm u & v är linjärt beroende.

Bevis

Om $v=0$ gäller olikheten och u,v linj. ber.

Om $v \neq 0$: Tag godt. $\alpha \in \mathbb{R}$.

$$0 \leq \|u-\alpha v\|^2 = \|u\|^2 + 2\langle u,-\alpha v\rangle + \|-\alpha v\|^2 =$$
$$= \|u\|^2 - 2\alpha\langle u,v\rangle + \alpha^2\|v\|^2$$

Låt $\alpha = \dfrac{\langle u,v\rangle}{\|v\|^2}$ (ok ty $v \neq 0$).

$$0 \leq \|u\|^2 - 2\frac{\langle u,v\rangle^2}{\|v\|^2} + \frac{\langle u,v\rangle^2}{\|v\|^4}\|v\|^2 = \|u\|^2 - \frac{\langle u,v\rangle^2}{\|v\|^2}$$

Multiplicera med: $\|v\|^2$:

$$0 \leq \|u\|^2\|v\|^2 - \langle u,v\rangle^2 \iff \langle u,v\rangle^2 \leq \|u\|^2\|v\|^2$$

$\Rightarrow \quad |\langle u,v\rangle| \leq \|u\|\cdot\|v\|$ (den sökta olikheten)

Vi har likhet omm $\|u-\alpha v\| = 0$

$\iff u-\alpha v = 0 \iff u = \alpha v$ d.v.s. u,v. linj ber.

▨

Triangelolikheten och vinklar

Def

Om $u,v \neq 0$ är VINKELN mellan u och v

$$\theta = \arccos\left(\frac{\langle u,v\rangle}{\|u\|\cdot\|v\|}\right)$$

OBS: Cauchy-Schwarz $\Rightarrow -1 \leq \dfrac{\langle u,v\rangle}{\|u\|\cdot\|v\|} \leq 1$

Så θ är definierad $\forall u,v$ (i linjärt rum med skalärprodukt).

Sats [2.2] Triangelolikheten

$\|u+v\| \le \|u\| + \|v\| \quad \forall u,v \in V$

likhet omm $v=0$ eller $u=\alpha v$ med $\alpha \ge 0$

Bevis

$$\|u+v\|^2 = \|u\|^2 + 2\langle u,v\rangle + \|v\|^2 \le$$

likhet om $\langle u,v\rangle \ge 0$

$$\le \|u\|^2 + 2|\langle u,v\rangle| + \|v\|^2 \le$$

CS, likhet om $v=0$ eller $u=\alpha v$ där $\alpha = \frac{\langle u,v\rangle}{\|v\|^2}$

$$\le \|u\|^2 + 2\|u\|\cdot\|v\| + \|v\|^2 =$$

$$= (\|u\|+\|v\|)^2$$

$\Leftrightarrow \quad \|u+v\| \le \|u\| + \|v\|$

Likhet omm $v=0$ eller $u=\alpha v$, $\alpha \ge 0$ ▨

Ex: Vinkel mellan två polynom

Bestäm vinkeln mellan $f=t$ & $g=t^3$ i $C([0,1])$ med skalärprodukt $\langle f,g\rangle = \int_0^1 f(t)g(t)\,dt$

$$\|f\|^2 = \int_0^1 t^2\,dt = 1/3$$

$$\|g\|^2 = \int_0^1 (t^3)^2\,dt = 1/7$$

$$\langle f,g\rangle = \int_0^1 t \cdot t^3\,dt = 1/5$$

$$\theta = \arccos\left(\frac{\langle f,g\rangle}{\|f\|\,\|g\|}\right) = \arccos\left(\frac{1/5}{1/\sqrt{3}\cdot 1/\sqrt{7}}\right) =$$

$$= \arccos\frac{\sqrt{21}}{5} = 0{,}4115\ldots \text{ rad}$$

Lana 200403 Föreläsning 2

Ortogonalitet

Om u,v har vinkel θ mellan sig

$$\cos\theta = \frac{\langle u,v\rangle}{\|u\|\cdot\|v\|}$$

Vinkelräta: $\theta = \frac{\pi}{2} \Rightarrow \langle u,v\rangle = 0$

Def

$u,v \in V$ är ORTOGONALA om $\langle u,v\rangle = 0$

En mängd är ORTOGONAL MÄNGD om dess vektorer är parvis ortogonala.

$u \in V$ är NORMERAD om $\|u\| = 1$

En ortogonal mängd med normerade vektorer kallas ORTONORMERAD eller ON-MÄNGD.

En ORTOGONALBAS är en bas och ortogonal mängd.

En ON-BAS är en bas och ON-mängd.

OBS: Nollvektorn är ortogonal mot alla vektorer.

Ex: $f=1$ & $g=2t-1$ är ortogonala i $C([0,1])$ med $\langle f,g\rangle = \int_0^1 f(t)g(t)dt$ ty:

$$\langle f,g\rangle = \int_0^1 1\cdot(2t-1)dt = [t^2-t]_0^1 = 0$$

$\|f\|^2 = \int_0^1 1^2 dt = [t]_0^1 = 1$ så f är normerad.

$\|g\|^2 = \int_0^1 (2t-1)^2 dt = \ldots = 1/3$ så g ej normerad.

men $\frac{g}{\|g\|} = \sqrt{3}g$ är normerad

$\Rightarrow \{f, \sqrt{3}g\}$ är en ON-mängd.

Sats [2.3]

a) En ON-mängd är linj. ober

b) Om $e_1, \dots, e_n$ är en ON-bas

$$u = \langle u, e_1 \rangle e_1 + \dots + \langle u, e_n \rangle e_n \quad \forall u \in V$$

OBS: $\langle u, e_i \rangle$ är koordinat i för u i $\mathbf{e}$

Bevis för b)

$$(*) \quad u = \sum_{i=1}^{n} \lambda_i e_i \Rightarrow \langle u, e_j \rangle = \sum_{i=1}^{n} \lambda_i \langle e_i, e_j \rangle$$

$$\langle e_i, e_j \rangle = \begin{cases} 0 & \text{om } i \neq j \\ 1 & \text{om } i = j \end{cases} \quad \text{ty } e_1 \dots e_n \text{ ON-bas}$$

$$\Rightarrow \langle u, e_i \rangle = \lambda_i$$

Sätt in i (*)

$$u = \langle u, e_1 \rangle e_1 + \dots + \langle u, e_n \rangle e_n \qquad ▨$$

Om $e_1, \dots, e_n$ bara är ortogonalbas, ersätt e_i med $\frac{e_i}{\|e_i\|}$ d.v.s. $u = \sum_{i=1}^{n} \frac{\langle u, e_i \rangle}{\|e_i\|^2} e_i$

Gram-Schmidts ortogonaliseringsprocess

Ett systematiskt sätt att konstruera en ON-bas.

Ex: Låt $v_1 = \begin{pmatrix}1\\1\\1\end{pmatrix}$ $v_2 = \begin{pmatrix}0\\1\\1\end{pmatrix}$ $v_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}$, $V = \text{span}\{v_i\}_{i=1}^{3}$

Bestäm en ON-bas $\{e_1, e_2, e_3\}$.

Steg 1: Låt $e_1' = v_1$

Steg 2: Bilda $e_2' = v_2 - \alpha e_1'$.

Vi vill att $\langle e_1', e_2' \rangle = 0$

$$\Rightarrow \langle e_1', v_2 - \alpha e_1' \rangle = \langle v_2, e_1' \rangle - \alpha \langle e_1', e_1' \rangle = 0$$

$$\Rightarrow \alpha = \frac{\langle v_2, e_1' \rangle}{\langle e_1', e_1' \rangle} \quad \text{d.v.s.} \quad e_2' = v_2 - \frac{\langle v_2, e_1' \rangle}{\langle e_1', e_1' \rangle} e_1'$$

I vårt exempel:

$$e_2' = \begin{pmatrix}0\\1\\1\end{pmatrix} - \frac{3}{4}\begin{pmatrix}1\\1\\1\end{pmatrix} = \frac{1}{4}\begin{pmatrix}-3\\1\\1\end{pmatrix}$$

$v_2 \in \text{span}\{e_1\} \Rightarrow e_2' \neq 0$ o $\text{span}\{e_1', e_2'\} = \text{span}\{v_1, v_2\}$

Steg 3:

$$\text{Låt} \quad e_3' = v_3 - \underbrace{\frac{\langle v_3, e_1' \rangle}{\langle e_1', e_1' \rangle} e_1'}_{\text{proj. på } e_1'} - \underbrace{\frac{\langle v_3, e_2' \rangle}{\langle e_2', e_2' \rangle} e_2'}_{\text{proj. på } e_2'}$$

lika

$$\text{Vi får} \quad \langle e_3', e_1' \rangle = \langle v_3, e_1' \rangle - \frac{\langle v_3, e_1' \rangle}{\cancel{\langle e_1', e_1' \rangle}} \cancel{\langle e_1', e_1' \rangle}$$

$$- \frac{\langle v_3, e_2' \rangle}{\langle e_2', e_2' \rangle} \cancel{\langle e_2', e_1' \rangle}^{0} =$$

$$= 0.$$

Och p.s.s. $\langle e_3', e_2' \rangle = 0$

⟶

→

Så $\{e_1', e_2', e_3'\}$ är parvis ortagonala.

$v_3 \notin \text{span}\{v_1, v_2\} = \text{span}\{e_1', e_2'\}$

$\Rightarrow e_3' \neq 0$ och $\text{span}\{e_1', e_2', e_3'\} = \text{span}\{v_1, v_2, v_3\}$

I vårt exempel:

$$e_3' = \begin{pmatrix}0\\0\\1\\1\end{pmatrix} - \frac{2}{4}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} - \frac{2}{12}\begin{pmatrix}-3\\1\\1\\1\end{pmatrix} = \frac{1}{3}\begin{pmatrix}0\\-2\\1\\1\end{pmatrix}$$

Så e_1', e_2', e_3' utgör en ortagonalbas för V.

<u>Steg 4</u>: Normera:

$$e_i = \frac{e_i'}{\|e_i'\|} \quad \text{ger} \quad \{e_1, e_2, e_3\} = \left\{ \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}, \frac{1}{2\sqrt{3}}\begin{pmatrix}-3\\1\\1\\1\end{pmatrix}, \frac{1}{\sqrt{6}}\begin{pmatrix}0\\-2\\1\\1\end{pmatrix} \right\}$$

e_1, e_2, e_3 utgör en ON-bas för V.

<u>Sats</u> [2.4]

Låt V vara ett ändligdimensionellt linjärt rum med skalärprodukt definierad och $\dim V > 0$. Då har V en ON-bas.

<u>Bevis</u> Låt $n = \dim V$ och $v_1, \ldots, v_n$ bas för V.

Låt $e_1' = v_1$. <u>Induktion</u>:

Antag att vi har $e_1', \ldots, e_k'$ parvis ortagonala och $\text{span}\{e_1', \ldots, e_k'\} = \text{span}\{v_1, \ldots, v_k\}$.

Låt $$\boxed{e_{k+1}' = v_{k+1} - \sum_{j=1}^{k} \frac{\langle v_{k+1}, e_j' \rangle}{\langle e_j', e_j' \rangle} e_j'} \qquad (*)$$

$$\langle e_{k+1}', e_i' \rangle = \langle v_{k+1}, e_i' \rangle - \sum_{j=1}^{k} \frac{\langle v_{k+1}, e_j' \rangle}{\langle e_j', e_j' \rangle} \underbrace{\langle e_i', e_j' \rangle}_{0 \text{ om } i \neq j} =$$

$$= \langle v_{k+1}, e_i' \rangle - \langle v_{k+1}, e_i' \rangle = 0$$

→

⟶

Så e'_{k+1} enl. (*) ger att även

$\{e'_1, \ldots, e'_{k+1}\}$ parvis ortagonala.

Vi linj. ober $\rightarrow$ $v_{k+1} \notin \text{span}\{v_1, \ldots, v_k\} = \text{span}\{e'_1, \ldots, e'_k\}$

$\Rightarrow e'_{k+1} \neq 0$ och $\text{span}\{e'_1, \ldots, e'_{k+1}\} \underset{\uparrow}{=} \text{span}\{v_1, \ldots, v_{k+1}\}$

(ty enl. (*) är v_{k+1} linkomb av $e'_1, \ldots, e'_{k+1}$)

Induktion ger att vi kan bilda

$\{e'_1, \ldots, e'_n\}$ ortagonala så $V = \text{span}\{e'_1, \ldots, e'_n\}$

Vi kan normera:

$$e_i = \frac{e'_i}{\|e'_i\|} \quad \forall i = 1, \ldots, n$$

$\Rightarrow \{e_1, \ldots, e_n\}$ ON-bas för V ▨

Ortogonalprojektioner

Sats [2.5] Pythagoras sats

Om $\{u_1, \ldots, u_n\}$ är en ortagonal mängd i V

$$\left\| \sum_{i=1}^{n} u_i \right\|^2 = \sum_{i=1}^{n} \|u_i\|^2$$

Bevis:

$$\left\| \sum_{i=1}^{n} u_i \right\|^2 = \left\langle \sum_{i=1}^{n} u_i , \sum_{i=1}^{n} u_i \right\rangle \overset{\substack{\text{skalärprodukt} \\ \text{linjär} \\ \downarrow}}{=} \sum_{i=1}^{n} \sum_{j=1}^{n} \overbrace{\langle u_i, u_j \rangle}^{0 \text{ om } i \neq j} =$$

$$= \sum_{i=1}^{n} \langle u_i, u_i \rangle = \sum_{i=1}^{n} \|u_i\|^2$$

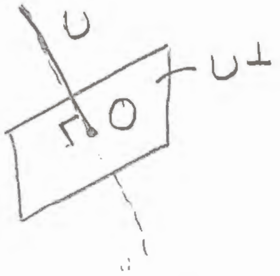
▨

Def

Låt U vara ett underrum av V.

Det ORTOGONALA KOMPLEMENTET är

$$U^\perp = \{ v \in V \mid \langle v,u \rangle = 0 \quad \forall u \in U \}$$



"Alla vektorer i V som är ortogonala mot alla vektorer i U"

Sats [2.6]

$U^\perp$ är ett underrum av V

Bevis

$v_1, v_2 \in U^\perp$, $\alpha, \beta \in \mathbb{R}$, $u \in U$

$$\Rightarrow \langle \alpha v_1 + \beta v_2, u \rangle = \alpha \overset{0}{\cancel{\langle v_1, u \rangle}} + \beta \overset{0}{\cancel{\langle v_2, u \rangle}} = 0$$

$$\Rightarrow \alpha v_1 + \beta v_2 \in U^\perp$$ ▨

Def

Om en vektor $u \in V$ kan skrivas $u = u' + u''$ där $u' \in U$ och $u'' \in U^\perp$ kallas u' ORTOGONALPROJEKTIONEN av u på U.

OBS: Om U är oändligdim är det inte säkert att ortogonalprojektionen existerar.

Sats [2.1]

Om ortogonalprojektionen existerar är den entydig.

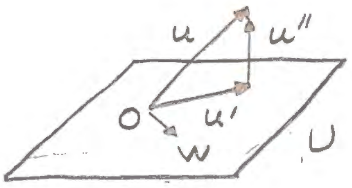
Lemma [2.2]

Om orto.proj. existerar $\forall u \in V$ så är $(U^\perp)^\perp = U$.

Lana 200415

Ortogonalitet & Tillämpningar

Om U underrum till V; $u = u' + u''$ med $u' \in U$, $u'' \in U^\perp$ är u' ortogonalprojektionen av u på U.



Sats [2.7]

Låt $u \in V$. För en vektor $u' \in U$:

$$\underbrace{u-u' \in U^\perp}_{(i)} \iff \underbrace{\|u-u'\| \le \|u-w\|}_{(ii)} \quad \forall w \in U$$

d.v.s. ortoganalprojektionen är närmaste vektorn i U för u.

Bevis

Antag $u-u' \in U^\perp$, $w \in U$ godt.

$$\|u-w\|^2 = \|\underbrace{(u-u')}_{\in U^\perp} + \underbrace{(u'-w)}_{\in U}\|^2 = \{\text{pythagoras sats}\} =$$

$$= \|u-u'\|^2 + \|u'-w\|^2 \ge \|u-u'\|^2$$

så $\|u-w\| \ge \|u-u'\| \quad \forall w \in U$

d.v.s. (i) $\Rightarrow$ (ii)

Antag $\|u-u'\| \le \|u-w\|$. Låt $u'' = u-u'$, w godt.

Vill visa att $u'' \in U^\perp$. För $t \in \mathbb{R}$:

$$\|u''\| \underset{(ii)}{\le} \|u-(u'-tw)\| = \|u''+tw\|$$

Bilda $\varphi(t) = \|u''+tw\|^2 = \|u''\|^2 + 2t\langle u'', w\rangle + t^2\|w\|^2$

$\varphi(t)$ har min i $t=0$ ty (*)

$\Rightarrow \quad 0 = \varphi'(0) = 2\langle u'', w\rangle$.

w godt så $u'' \in U^\perp$

▨

Sats [2.8]

Antag $\dim U = k$. För $u \in V$ existerar entydigt $u' \in U$ så $u - u' \in U^{\perp}$. Om $\{e_1, \dots, e_k\}$ är en ON-bas för U

$$u' = \langle u, e_1 \rangle e_1 + \dots + \langle u, e_k \rangle e_k \quad (u' = 0 \text{ om } k = 0)$$

Anm: om $\|e_i\| \neq 1$, använd $\frac{1}{\|e_i\|} e_i$

Sats [2.9]

Om $\dim V = n$, U underrum till V, så

$$\dim U + \dim U^{\perp} = n$$

Fundamentala underrum till matriser

Låt A $m \times n$-matris

$N(A)$ och $V(A^T)$ är underrum av $\mathbb{R}^n$

$V(A)$ och $N(A^T)$ är underrum av $\mathbb{R}^m$

Sats [2.11] De 4 fundamentala underrumen

För en godt. $m \times n$-matris A så gäller

- $N(A) = V(A^T)^{\perp}$
- $N(A)^{\perp} = V(A^T)$
- $N(A^T) = V(A)^{\perp}$
- $N(A^T)^{\perp} = V(A)$

Bevis

$$A = \begin{pmatrix} — & r_1 & — \\ & \vdots & \\ — & r_m & — \end{pmatrix} \qquad Ax = \begin{pmatrix} r_1^T \cdot x \\ \vdots \\ r_m^T \cdot x \end{pmatrix}$$

$x \in N(A) \iff Ax = 0 \iff x \perp$ mot $\forall r_i^T$

$\iff x \perp \forall$ kolonner i A^T

$\iff x \perp V(A^{\perp}) = \text{span}\{$kolonner i $A^T\}$

$\iff x \in V(A^T)^{\perp}$

x godt. i $N(A)$

$\Rightarrow \underline{N(A) = V(A^T)^{\perp}}$

Det följer nu att

$$(N(A))^{\perp} = (V(A^T)^{\perp})^{\perp}$$

$\Rightarrow \underline{\underline{N(A)^{\perp} = V(A^T)}}$

A var godt, så byt A mot A^T

$\Rightarrow \underline{N(A^T) = V(A)^{\perp}}$

$\Rightarrow \underline{N(A^T)^{\perp} = V(A)}$

▨

Minstakvadratmetoden

Vill lösa $Ax=b$ med A $m\times n$-matris, $b\in\mathbb{R}^m$

Ibland finns ej lösningar, typiskt om $m>n$

Vi söker då x så att Ax blir så nära b som möjligt, d.v.s. minimerar $\|Ax-b\|$.

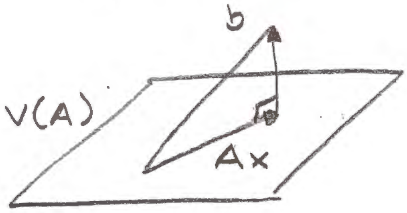
$$\|Ax-b\|^2=([Ax]_1-b_1)^2+\dots+([Ax]_m-b_m)^2$$

(minsta kvadratsumma)

Obs: Alla vektorer Ax utgör $V(A)$.

Vi söker $b'=Ax\in V(A)$ som minimerar $\|b'-b\|$.

Sats 2.7 $\longrightarrow$ $\iff b-b'\in V(A)^\perp$ d.v.s. b' ortoproj av b på $V(A)$.



Sats [2.12]

Det finns minst en minstakvadratlösning till $Ax=b$. x är minstakvadratlösning omm

$$A^TAx=A^Tb \quad \text{(Normalekvationerna)}$$

Bevis

Enl. sats 2.7: x MK-lösn $\iff Ax=b'$ där $b-b'\in V(A)^\perp$

Enl. sats 2.8: $\exists!\, b'\in V(A) \Rightarrow \exists x: Ax=b'$

$$x \text{ MK-lösn} \overset{\text{Sats 2.11}}{\iff} b-Ax\in V(A)^\perp=N(A^T)$$

$$\iff A^T\underbrace{(b-Ax)}_{\text{residual, felet}}=0 \iff A^TAx=A^Tb \qquad ▨$$

"residualen ligger i nollrummet till A^T"

Obs: b' behöver inte beräknas, men $b' = Ax$ för godt MK-lösn x.

b' är entydig, men $Ax = b' \Leftrightarrow A^TAx = A^Tb$ kan ha flera lösningar.

Obs: Antag att $Ax = b$ är lösbar

$\Rightarrow b \in V(A) \Rightarrow b' = b$

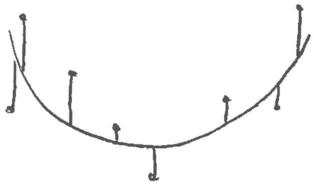
$A^TAx = A^Tb \Leftrightarrow Ax = b$

(Normalekvationen har samma lösningar)

Exempel Anpassa andragradspolynom

$y = c_1 + c_2 t + c_3 t^2$ till mätdata

t	t_1	t_2	t_3	$\cdots$	t_n
y	y_1	y_2	y_3	$\cdots$	y_n



d.v.s. Minimera $\sum_{i=1}^{n} (c_1 + c_2 t_i + c_3 t_i^2 - y_i)^2 =$

$$= \left\| \begin{pmatrix} 1 & t_1 & t_1^2 \\ & \cdots & \\ 1 & t_n & t_n^2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} - \begin{pmatrix} y_1 \\ \cdots \\ y_n \end{pmatrix} \right\|^2 = \| Ax - b \|^2$$

d.v.s. lös normalekv. $A^TAx = A^Tb$

ger bästa $x = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$

Anm.

Tag t.ex. $n = 4$ $t_1 = 0$ $t_2 = 1$ $t_3 = 10$ $t_4 = 100$

ger $A^TA = \begin{pmatrix} 4 & \cdots & \\ \vdots & \ddots & \\ \vdots & & \sim 10^8 \end{pmatrix} \Rightarrow$ stor skillnad mellan elementen.

Vi får lätt numeriska problem.

Ortogonala matriser

Def En kvadratisk matris kallas ORTOGONAL om $A^TA = I$

Lemma

En kvadratisk matris A är ortogonal omm dess kolonner är parvis ortagonala och har längd 1.

Bevis

Om $A = \begin{pmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{pmatrix}$:

$$A^TA = \begin{pmatrix} - a_1^T - \\ \vdots \\ - a_n^T - \end{pmatrix} \begin{pmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{pmatrix} = \begin{pmatrix} a_1 \cdot a_1 & \cdots & a_1 \cdot a_n \\ \cdots & & \cdots \\ a_n \cdot a_1 & \cdots & a_n \cdot a_n \end{pmatrix}$$

$$a_i \cdot a_j = [A^TA]_{ij} = I_{ij} = \begin{cases} 0 & \text{om } i \neq j \\ 1 & \text{om } i = j \end{cases}$$

Så parvis ortagonala $\Leftrightarrow$ $A^TA = I$

OBS: A ortogonal om kolonnerna är <u>ortonormala</u>

A ortagonal $\Leftrightarrow A^TA = I \Rightarrow A^{-1} = A^T$

$\Rightarrow I = AA^{-1} = AA^T = (A^T)^T A^T$

$\Rightarrow A^T$ ortogonalmatris

$\Rightarrow$ raderna ortonormala

ON-basbyten

Låt $\{e_1, \dots, e_n\}$ vara ON-bas för V.

Låt $u, v \in V$ och $[u]_e = x \in \mathbb{R}^n$ o̲ $[v]_e = y \in \mathbb{R}^n$

Skalärprodukten

$$\langle u, v \rangle = \left\langle \sum_{i=1}^{n} x_i e_i , \sum_{j=1}^{n} y_j e_j \right\rangle =$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} x_i y_j \langle e_i, e_j \rangle \overset{\text{ON-bas}}{=}$$

$$= \sum_{i=1}^{n} x_i y_i =$$

$$= x^T y = x \cdot y$$

Så $\boxed{\langle u, v \rangle = [u]_e \cdot [v]_e}$ om e ON-bas!

Låt $\{e_1', \dots, e_2'\}$ vara en annan ON-bas.

Transformationsmatrisen

$$T = T_{e \leftarrow e'} = \begin{pmatrix} | & & | \\ [e_1']_e & \cdots & [e_n']_e \\ | & & | \end{pmatrix}$$

$$[e_i']_e \cdot [e_j']_e = \langle e_i', e_j' \rangle \underset{\substack{\uparrow \\ e' \text{ ON-bas}}}{=} \begin{cases} 0 & \text{om } i \neq j \\ 1 & \text{om } i = j \end{cases}$$

d.v.s. kolonnerna i T är parvis ortagonala med längd 1 => T ortogonalmatris.

Lana 200416

Matrisrepresentation av linjära avbildningar.

$F(x) = Ax$ är typexempel på linj. avb.

Om $F: U \to V$ och U, V ändligdim så kan den representeras av matris!

Låt $n = \dim U$ och $m = \dim V$.

$F: U \to V$ linj. avb. $\mathbb{e}$ bas för U, $\mathbb{f}$ bas för V

$u \in U$ godt., $v = F(u)$, $x = [u]_{\mathbb{e}} \in \mathbb{R}^n$, $y = [v]_{\mathbb{f}} \in \mathbb{R}^m$

$$v = F(u) = F\left(\sum_{j=1}^{n} x_j e_j\right) = \sum_{j=1}^{n} x_j F(e_j)$$

$$\Rightarrow y = [v]_{\mathbb{f}} = \sum_{j=1}^{n} x_j [F(e_j)]_{\mathbb{f}} =$$

$$= \underbrace{\begin{pmatrix} | & & | \\ [F(e_1)]_{\mathbb{f}} & \dots & [F(e_n)]_{\mathbb{f}} \\ | & & | \end{pmatrix}}_{=:A} x$$

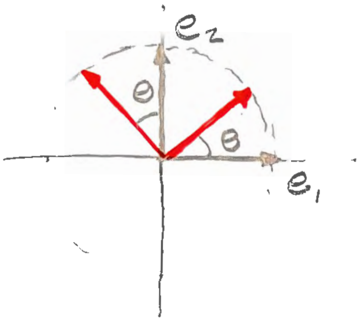
A är en $m \times n$-matris som relaterar y till x.

Kolonnerna i A är koordinatvektorerna för $F(e_1) \dots F(e_n)$ i basen $\mathbb{f}$. A kallas "matrisen för avbildningen F i baserna $\mathbb{e}$ & $\mathbb{f}$".
$y = Ax$ matrisrepresentation av $F(u) = v$. Om $U = V$ och $\mathbb{e} = \mathbb{f}$ är A matrisen för F i basen $\mathbb{e}$.

Samma princip som vanligt. "Var hamnar basvektorerna?"

Exempel: Vridning i planet

$\mathbb{e} = \mathbb{f} = \{e_1, e_2\}$ $U = V =$ geom. planet



$$F(e_1) = \cos\theta\, e_1 + \sin\theta\, e_2$$
$$\Rightarrow [F(e_1)]_{\mathbb{e}} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

$$F(e_2) = -\sin\theta\, e_1 + \cos\theta\, e_2$$
$$\Rightarrow [F(e_2)]_{\mathbb{e}} = \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix}$$

$$\Rightarrow A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Egenvärden och egenvektorer

Exempel

Låt $A = \begin{pmatrix} -1 & 6 \\ 0 & 2 \end{pmatrix}$, $F(x) = Ax$

Hur avbildar F vektorer? Finns vektorer som avbildas på ett enkelt sätt?

$$F\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 6 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} = 2\begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$F\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 & 6 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} = -1\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ kallas egenvektor med egenvärde 2.

$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ kallas egenvektor med egenvärde -1.

Egenvärden kan användas för att förstå F.

Exempel

Låt $V = C^\infty(\mathbb{R})$ och $F(f) = f'$ (derivata)

$F(e^{\lambda t}) = \lambda e^{\lambda t}$ så $f(t) = e^{\lambda t}$ är egenvektor för F med egenvärde λ.

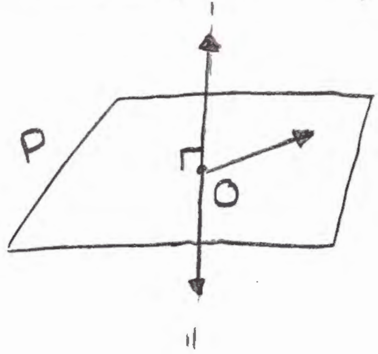
Def En EGENVEKTOR för en linjär avbildning $F: V \to V$ är en vektor $u \neq 0$ så att $F(u) = \lambda u$ för något tal λ som kallas EGENVÄRDE. Alla vektorer u (även 0) som uppfyller $F(u) = \lambda u$ för ett egenvärde λ utgör ett underrum $E(\lambda)$ som kallas EGENRUMMET till egenvärdet λ.

Om V är ändligdimensionellt, A är matrisen för F i en bas e och $x = [u]_e$ där $u \in V$:

$$F(u) = \lambda u \iff \boxed{Ax = \lambda x} \iff (A - \lambda I)x = 0$$

Så $E(\lambda)$ motsvarar $N(A - \lambda I)$.

Exempel spegling i plan i rummet



Planet är egenrummet till egenvärdet 1 (Vektorerna avbildas på sig själva). $E(1) = P$

En normal till planet är egenvektor med egenvärde -1. $E(-1) = P^\perp$

Exempel

$F(f) = f''$ $F(\sin(\alpha t)) = -\alpha^2 \sin(\alpha t)$

Så $\sin(\alpha t)$ är egenvektor med egenvärde $-\alpha^2$

Egenvärden och egenvektorer viktiga för lösning av differentialekvationer.

Exempel

Schrödingerekvationen $H\Psi = E\Psi$ (där E är ett tal) är en egenvärdesekvation. Detta är grundläggande för kvantmekanik. Egenvärden ger spektrallinjer, mängden av egenvärden kallas spektrum.

Exempel

Rotationen av en stelkropp beskrivs m.h.a. principalaxlar = egenvektorer till tröghetsmatris.

Exempel

Resonans sker vid frekvenser som är egenvärden till ett egenvärdesproblem.

Beräkning av egenvektorer och egenvärden

Idé

λ är egenvärde till $A \Leftrightarrow (A-\lambda I)x = 0$ har icke-trivial lösning $\Leftrightarrow A-\lambda I$ ej inverterbar $\Leftrightarrow \det(\underbrace{A-\lambda I}_{\text{polynom i } \lambda \text{ av grad } n}) = 0$

Sats

Låt V vara ändligdim med bas $\mathbb{e}$.
Låt $F: V \rightarrow V$ vara en lin. avb. med matris A i basen $\mathbb{e}$. Då är λ egenvärde till F omm $\det(A-\lambda I)=0$

Def $\det(A-\lambda I)$ kallas KARAKTÄRISTISKA POLYNOMET till A eller F.
$\det(A-\lambda I)=0$ kallas KARAKTÄRISTISKA EKVATIONEN.

Exempel

Beräkna egenvärden o̲ vektorer till $A = \begin{pmatrix} 2 & 3 \\ 3 & -6 \end{pmatrix}$

Kar. pol.: $\det(A-\lambda I) = \begin{vmatrix} 2-\lambda & 3 \\ 3 & -6-\lambda \end{vmatrix} =$
$= (2-\lambda)(-6-\lambda) - 9 = (\lambda-3)(\lambda+7)$

Så A har egenvärden 3 o̲ -7,
ty karaktäristiska ekvationen har rötter 3 o̲ -7

→

Egenvektorer till $\lambda_1 = 3$:

$Ax = \lambda_1 x \iff (A - \lambda_1 I)x = 0 \iff$

$$\iff \left(\begin{array}{cc|c} -1 & 3 & 0 \\ 3 & -9 & 0 \end{array}\right) \sim \left(\begin{array}{cc|c} -1 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right) \iff$$

$-x_1 + 3x_3 = 0 \iff x = \begin{pmatrix} 3 \\ 1 \end{pmatrix} t \quad t \in \mathbb{R}$

Så $v_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ är egenvektor med egenvärde 3.
$E(3) = \text{span}\{v_1\}$

Egenvektorer till $\lambda_2 = -7$:

$$(A + 7I)x = 0 \iff \left(\begin{array}{cc|c} 9 & 3 & 0 \\ 3 & 1 & 0 \end{array}\right) \sim \left(\begin{array}{cc|c} 3 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right)$$

$\iff x = \begin{pmatrix} -1/3 \\ 1 \end{pmatrix} t \quad t \in \mathbb{R}$

Så $v_2 = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$ egenvektor med egenvärde -7.
$E(-7) = \text{span}\{v_2\}$

Anm! Flera parametrar ⟺ Högre dim på egenrummet.

Anm! Metoden fungerar för små matriser och handräkning. Fungerar dåligt numeriskt. Determinantberäkningar dyra och numeriskt instabilt att hitta rötter till polynom av hög grad.

Om A $n \times n$-matris:

$$(*) \quad \det(A-\lambda I) = \begin{vmatrix} a_{11}-\lambda & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn}-\lambda \end{vmatrix} =$$

$$= (a_{11}-\lambda)\ldots(a_{nn}-\lambda) + \ldots =$$

$$= (-1)^n \lambda^n + (-1)\lambda^{n-1} \underbrace{(a_{11}+a_{22}+\ldots+a_{nn})}_{Sp(A)} \lambda^{n-1} +$$

$$+ \ldots + \det(A)$$

Vi får ett polynom av grad n.
Algebrans fundamentalsats ger exakt
n rötter (räknat med multiplicitet).
Dessa är egenvärdena λ_k.
=> Alla matriser har (komplexa) egenvärden.
=> Behöver räkna med komplexa vektorrum

$$(**) \quad \det(A-\lambda I) = (-1)^n(\lambda-\lambda_1)\ldots(\lambda-\lambda_n) =$$

$$= (-1)^n \lambda^n - (-1)^{n-1}(\lambda_1+\lambda_2+\ldots+\lambda_n)\lambda^{n-1} + \ldots +$$

$$+ \ldots + \lambda_1\lambda_2\ldots\lambda_n$$

Jämför (*) med (**):

$$\boxed{\begin{aligned} \lambda_1+\lambda_2+\ldots+\lambda_n &= Sp(A) \\ \lambda_1\lambda_2\ldots\lambda_n &= \det(A) \end{aligned}}$$

Speciellt: $\det(A) = 0 \iff$ minst ett egenvärde 0

Om A är triangulär:

$$\det(A-\lambda I)=\begin{pmatrix} a_{11}-\lambda & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn}-\lambda \end{pmatrix}=$$

$$=(a_{11}-\lambda)(a_{22}-\lambda)\dots(a_{nn}-\lambda)$$

=> egenvärdena är diagonalelementen.

Om A är blocktriangulär:

$$A=\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}.$$

$$\det(A-\lambda I)=\det\begin{pmatrix} A_{11}-\lambda I & A_{12} \\ 0 & A_{22}-\lambda I \end{pmatrix}=$$

$$=\det(A_{11}-\lambda I)\det(A_{22}-\lambda I)$$

<u>Sats</u> [4.2]

A och A^T har samma egenvärden

<u>Bevis</u> $\det(A^T-\lambda I)=\det(A^T-\lambda I)^T=\det(A-\lambda I)$

Samma kar. ekv => samma egenvärden ▨

Komplexa vektorrum

Behöver komplexkonjugat $\bar{u}$ av u

Def En skalärprodukt $\langle u,v \rangle$ i ett komplext vektorrum V är komplexvärd och uppfyller:

(i) $\langle u,v \rangle = \overline{\langle v,u \rangle} \quad \forall u,v \in V$

(ii) $\langle u,\alpha v \rangle = \alpha \langle u,v \rangle \quad \forall u,v \in V,\ \alpha \in \mathbb{C}$

(iii) $\langle u+v,w \rangle = \langle u,w \rangle + \langle v,w \rangle \quad \forall u,v,w \in V$

(iv) $\langle u,u \rangle \geq 0 \quad \forall u \in V$, likhet om $u=0$

OBS: (i) & (ii) ger $\langle \alpha u,v \rangle = \bar{\alpha} \langle u,v \rangle$

Komplexa matriser:

Konjugattransponat $A^* := \bar{A}^T$

Se sektion 4.8 för mer komplexa rum med skalärprodukt.

$$V = \mathbb{C}^2 \quad x = \begin{pmatrix} a+ci \\ b+di \end{pmatrix}$$

Lana 200420

Matrisrepresentation av linjära avbildningar

$F: U \to V$ linjär avbildning, $\mathbb{e}$ bas för U

$\mathbb{f}$ bas för V. Matrisen för F är

$$A = \begin{pmatrix} | & & | \\ [F(e_1)]_{\mathbb{f}} & \cdots & [F(e_n)]_{\mathbb{f}} \\ | & & | \end{pmatrix}$$

Vi får då $F(u) = v \iff [v]_{\mathbb{f}} = A[u]_{\mathbb{e}}$

λ egenvärde och u egenvektor om $F(u) = \lambda u$ $(U=V)$

Exempel Komplexa egenvärden

Låt $V = \text{span}\{\sin t, \cos t\}$, $F: V \to V$ $F(f) = f'$

$\{e_1, e_2\} = \{\sin t, \cos t\}$ bas för V

Vad är matrisen för F?

$F(e_1) = \cos t = e_2$ $F(e_2) = -\sin t = -e_1$

$$\Rightarrow A = \begin{pmatrix} | & | \\ [e_2]_{\mathbb{e}} & [-e_1]_{\mathbb{e}} \\ | & | \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Sök egenvärden:

[Kar. ekv.] $0 = \det(A - \lambda I) = \begin{vmatrix} -\lambda & -1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 + 1$

Egenvärden: $\underline{\underline{\lambda_1 = -i}}$ $\underline{\underline{\lambda_2 = i}}$

Egenvektor till λ_1: $(A + iI)x = 0$

$$\iff \left[\begin{array}{cc|c} i & -1 & 0 \\ 1 & i & 0 \end{array}\right] \iff \left[\begin{array}{cc|c} i & -1 & 0 \\ 0 & 0 & 0 \end{array}\right] \Rightarrow x = \begin{pmatrix} i \\ 1 \end{pmatrix} t \quad t \in \mathbb{C}$$

Egenfunktion $e_1 + ie_2 = \sin t + i\cos t = \underline{\underline{e^{it}}}$

Egenvektor till λ_2 p.s.s. $x = \begin{pmatrix} 1 \\ -i \end{pmatrix} t$ $t \in \mathbb{C}$

Egenfunktion $\sin t - i\cos t = \underline{\underline{e^{-it}}}$

Exempel [4.11]

Låt $V = C^{\infty}(\mathbb{R})$ och $F(f) = f'$

Bestäm egenvärde och egenfunktioner!

f egenfunktion $\Leftrightarrow F(f) = \underset{\text{ODE}}{\boxed{f' = \lambda f}} \Leftrightarrow f(t) = Ce^{\lambda t}$

Alla $\lambda \in \mathbb{C}$ är egenvärden med egenfunktion $e^{\lambda t}$.

OBS: Behöver ej bas för egenvektorer/värden

Sammansättningar av linjära avbildningar

Låt $F: U \to V$, $G: U \to V$ linj. avb.

Låt $(F+G)(u) := F(u) + G(u) \quad \forall u \in U$

$(\alpha F)(u) := \alpha F(u) \quad \forall u \in U$

Detta ger ett linjärt rum (se storgruppsövning 3)

Om $F: U \to V$, $H: V \to W$ lin. avb. är

$HF: U \to W$ avbildningen $(HF)(u) = H(F(u)) \quad \forall u \in U$

HF är uppenbart linjär (Låt verka på $\alpha u + \beta v$).

Hur ser matrisen ut?

Om U, V, W ändlig dim och e, f, g deras baser

F har matris A, H har matris B m.a.p. baserna.

$\Rightarrow$ **HF har matris BA** matrisprodukt

(Detta är grunden för matrisproduktens definition)

Sats [3.1]

Låt $F: U \to V$ lin. avb., U ändligdim. lin. rum, V lin. rum (som kan vara oändligdimensionellt). Då är $V(F)$ ändligdimensionellt och

$$\dim N(F) + \dim V(F) = \dim U$$

(Analog till Sats [1.14] för matriser, liknande bevis)

Bevisidé

Bilda en bas $e_1, \dots, e_n$ för U så att

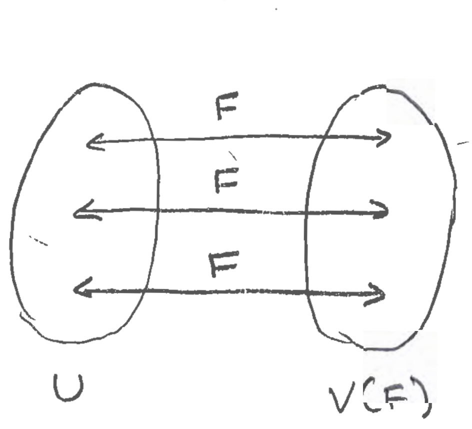
$\text{span}\{e_1, \dots, e_p\} = N(F)$

Vi får då $\{F(e_{p+1}), \dots, F(e_n)\}$ bas för $V(F)$, (men måste visa att de är linj. ober)

$\Rightarrow \dim V(F) = n - p = \dim U - \dim N(F)$ □

Anm. Om $N(F) = \{0\} \Leftrightarrow \dim N(F) = 0$ får vi att för varje $v \in V(F)$ finns ett unikt $u \in U$ så att $F(u) = v$. och vice versa.

Avbildningen kallas då ISOMORFISM mellan U och $V(F)$. U och $V(F)$ kallas ISOMORFA.



Jämför med bijektivitet.

"Isomorfa rum har i någon mening samma form."

Omformulering av sats [1.10]:
$\dim V(F) \leq \dim(U)$, likhet omm $V(F)$ och U är isomorfa.

Om $\mathbf{e}$ är bas för U är $F: U \to \mathbb{R}^{\dim U}$
$F(u) = [u]_{\mathbf{e}}$ en isomorfism. Så

$$\boxed{\text{Alla } n\text{-dimensionella rum är isomorfa med } \mathbb{R}^n}$$

"Kan vi göra något i $\mathbb{R}^n$ kan vi göra samma sak i alla ändligdimensionella rum."

<u>Exempel</u> Låt
$f_1(x) = \sin x \quad f_2(x) = x\cos x \quad f_3(x) = x^2 \sin x$
$g_1(x) = \cos x \quad g_2(x) = x \sin x \quad g_3(x) = x^2 \cos x$

$\mathbf{e} = \{f_1, f_2, f_3\}$ är lin. ober. i $C^1(\mathbb{R})$ och utgör därför en bas för $U = \operatorname{span}\{f_1, f_2, f_3\}$
$\mathbf{f} = \{g_1, g_2, g_3\}$ p.s.s. bas för $V = \operatorname{span}\{g_1, g_2, g_3\}$

Låt $D: U \to V$ vara deriveringsoperatorn $D(f) = f'$.
Derivera $\mathbf{e}$:
$f_1'(x) = \cos x = g_1(x)$
$f_2'(x) = \cos x - x\sin(x) = g_1(x) - g_2(x)$
$f_3'(x) = 2x\sin x + x^2\cos x = 2g_2(x) + g_3(x)$
Så D har i baserna $\mathbf{e}$ o $\mathbf{f}$ matris:

$$A = \left([De_1]_{\mathbf{f}} \; [De_2]_{\mathbf{f}} \; [De_3]_{\mathbf{f}} \right) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

→

$\longrightarrow$

A är en inverterbar matris

$\Rightarrow \dim(N(D)) = 0 \Rightarrow$ D är en isomorfism mellan U och $V(D) = V$

T.ex. $h(x) = (3+x^2)\cos x - 2x\sin x =$

$= 3g_1(x) - 2g_2(x) + g_3(x) \in V$

Sök $k(x) \in U$ s.a. $Dk = h$, d.v.s. finn en primitiv funktion $k(x) = \int h(x)\,dx \in U$

$\uparrow$ OBS => ingen konstant-term

$[h]_f = \begin{pmatrix} 3 \\ -2 \\ 1 \end{pmatrix}$ Lös $A[k]_e = [h]_f \Leftrightarrow$

$$\Leftrightarrow \left[\begin{array}{ccc|c} 1 & 1 & 0 & 3 \\ 0 & -1 & 2 & -2 \\ 0 & 0 & 1 & 1 \end{array}\right] \Rightarrow [k]_e = \begin{pmatrix} -1 \\ 4 \\ 1 \end{pmatrix}$$

Alltså $\int h(x)\,dx = -f_1(x) + 4f_2(x) + f_3(x) =$

$= -\sin x + 4x\cos x + x^2 \sin x$

<u>OBS</u>: Om konstanta funktioner hade ingått i U hade $\dim N(D) > 0$ och vi hade inte fått en isomorfism.

Basbyten vid linjära avbildningar.

Sats

Låt $\mathbb{e} = \{e_1, \dots, e_n\}$ och $\mathbb{e}' = \{e_1', \dots, e_n'\}$ vara två baser för det linjära rummet V. Om den linjära avbildningen $F: V \to V$ har matris A m.a.p. $\mathbb{e}$ ≗ matrisen A' m.a.p. $\mathbb{e}'$ så gäller att $\boxed{A' = T_{\mathbb{e}' \leftarrow \mathbb{e}} A T_{\mathbb{e} \leftarrow \mathbb{e}'}}$.

Bevis

Låt $u \in V$ godt. och $v = F(u)$.

Låt $x = [u]_{\mathbb{e}}$, $x' = [u]_{\mathbb{e}'}$ $y = [v]_{\mathbb{e}}$, $y' = [v]_{\mathbb{e}'}$

Vi har att $y = Ax$ ≗ $y' = A'x'$ per def. av A och A'.

$$\underline{\underline{A'x'}} = y' = T_{\mathbb{e}' \leftarrow \mathbb{e}} y = T_{\mathbb{e}' \leftarrow \mathbb{e}} A x = \underline{\underline{T_{\mathbb{e}' \leftarrow \mathbb{e}} A T_{\mathbb{e} \leftarrow \mathbb{e}'} x'}}$$

Men $x' = [u]_{\mathbb{e}'}$ och u godt, så

$$A' = T_{\mathbb{e}' \leftarrow \mathbb{e}} A T_{\mathbb{e} \leftarrow \mathbb{e}'} \qquad ▨$$

Def Två matriser A, B kallas SIMILÄRA om $B = T^{-1}AT$ för någon matris T. (Anm. måste vara kvadratiska matriser)

Determinanten är invariant under similaritetstransformationer?

$$\det A' = \cancel{\det T^{-1}} \det A \, \cancel{\det T} = \det A$$

Determinanten för en linjär avbildning är oberoende av val av bas.

Egenvärden och egenvektorer under basbyten

Om $A' = T^{-1}AT$ d.v.s. A och A' similära (matriser för en avbildning F i två olika baser)

Karaktäristiska polynomet:

$$\det(A' - \lambda I) = \det(T^{-1}AT - \lambda T^{-1}T) =$$

$$= \cancel{\det T^{-1}} \det(A - \lambda I) \cancel{\det T} =$$

$$= \det(A - \lambda I)$$

Similära matriser har samma kar. pol

=> samma egenvärden

Egenvärden för F är oberoende av bas

Om $A'y = \lambda y$ d.v.s. y egenvektor till A'

$\Rightarrow T^{-1}ATy = \lambda y \Rightarrow ATy = T\lambda y \Rightarrow A(Ty) = \lambda(Ty)$

Så $Ty = T_{e \leftarrow e'} y$ är egenvektor till A.

$$\boxed{\begin{array}{c} y' \text{ egenvektor till } A' \\ \Updownarrow \\ y = T_{e \leftarrow e'} y' \text{ egenvektor till } A \end{array}}$$

Lana 200421

Ortogonala matriser

Repetition

A ortagonal $\underset{\text{def.}}{\Longleftrightarrow}$ $A^TA = I$

Vid basbyte mellan två ON-baser blir transformationsmatrisen ortagonal.

$\langle u, v \rangle = [u]_e \cdot [v]_e$ om e ON-bas.

Om $F : V \to V$ och två baser e, e' till V och A, A' matriserna till F i resp. bas så

$$A' = T^{-1}AT$$

Om e och e' är ON-baser:

T ortagonal $\Rightarrow$ $T^TT = I$ $\Rightarrow$ $T^T = T^{-1}$

$\Rightarrow \boxed{A' = T^TAT}$

Detta är mycket enklare att beräkna.

L1

Linjära avbildningar med ortogonala matris

Ska se: A ortogonalmatris

$\Rightarrow$ F bevarar avstånd och vinklar

Sats [3.3]

Antag A är $n \times n$-ortogonalmatris. Då gäller

a) $\det A = 1$ eller -1

b) $(Ax) \cdot (Ay) = x \cdot y \quad \forall x, y \in \mathbb{R}^n$

b') $\|Ax\| = \|x\| \quad \forall x \in \mathbb{R}^n$

b'') x och y ortagonala $\Rightarrow$ Ax och Ay ortagonala

c) A, B ortagonala $n \times n$-matriser $\Rightarrow$ AB ortagonal

Bevis

Per def $A^T A = I$

a) $1 = \det I = \det A^T \det A = (\det A)^2$

$\Rightarrow \det A = \pm 1$

b) $(Ax) \cdot (Ay) = (Ax)^T (Ay) = x^T A^T A y = x^T y = x \cdot y$

b') Låt $y = x$ i b) $\Rightarrow \|Ax\|^2 = \|x\|^2$

$\Rightarrow \|Ax\| = \|x\|$

b'') $x \perp y \Leftrightarrow x \cdot y = 0 \overset{b)}{\Rightarrow} (Ax) \cdot (Ay) = 0 \Leftrightarrow Ax \perp Ay$

c) $(AB)^T (AB) = B^T A^T A B = B^T B = I$ ▨

Sats

Om A $n \times n$-matris med $\|Ax\| = \|x\|$ så är A en ortogonalmatris.

Bevis tag $x, y \in \mathbb{R}^n$

(1) $\|Ax + Ay\|^2 = \|Ax\|^2 + 2(Ax)\cdot(Ay) + \|Ay\|^2 =$
$= \|x\|^2 + 2(Ax)\cdot(Ay) + \|y\|^2$

(2) $\|Ax + Ay\|^2 = \|A(x+y)\|^2 = \|x+y\|^2 =$
$= \|x\|^2 + 2x\cdot y + \|y\|^2$

Sätt (1) = (2)

$\Rightarrow \quad (Ax)\cdot(Ay) = x\cdot y \quad \forall x, y \in \mathbb{R}^n \quad (*)$

Speciellt för standardbasvektorer e_i:

$Ae_j = a_j$, d.v.s. kolonn j i A

$$a_i \cdot a_j = (Ae_i)\cdot(Ae_j) \overset{(*)}{=} e_i \cdot e_j = \begin{cases} 1 & \text{om } i = j \\ 0 & \text{om } i \neq j \end{cases}$$

$\Rightarrow \quad A^T A = I$ ▨

Geometrisk tolkning av ortagonala matriser

Vinklar och avstånd bevaras.

2D - fallet

Sats [3.4]

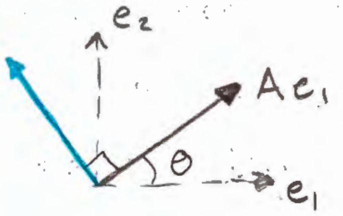
Om A är en ortagonal 2×2 matris är motsvarande avbildning på $\mathbb{R}^2$ en rotation eller spegling i en linje genom origo.

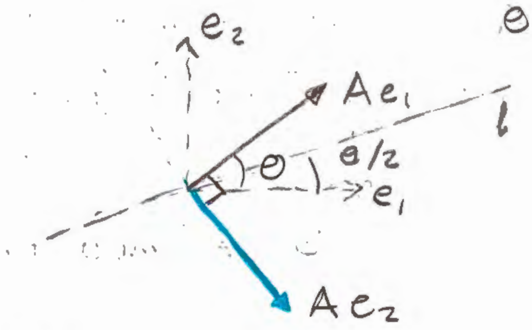
Förklaring.

Hur avbildas e_1 och e_2?

Sats [3.3] $\Rightarrow$ Ae_1 och Ae_2 ortagonala och längd 1.

Två möjligheter:





θ godt.

Rotation med θ grader kring O:

$$A = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

$$\det A = \cos^2\theta + \sin^2\theta = 1$$

Spegling i ℓ genom O

$$A = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} =$$

$$\det A = -\cos^2\theta - \sin^2\theta = -1$$

Anm. Det gäller att

$\det A = 1 \Rightarrow F$ rotation

$\det A = -1 \Rightarrow F$ spegling

3D - Fallet

Sats [3.5]

Antag A är ortagonal 3x3 - matris.

Om det A = 1 motsvarar A rotation i $\mathbb{R}^3$

Om det A = 1 motsvarar A rotation följd av spegling i origo.

Bevisidé

Om det A = 1 :

$$\det(A - I) = \det(A - A^TA) = \det((I - A^T)A) =$$

$$= \det(I - A^T)\overset{1}{\cancel{\det A}} = \det(-I(A - I)) =$$

$$= \det(-I)\det(A - I) = -\det(A - I)$$

$\Rightarrow$ $\det(A - I) = 0$ så $\lambda = 1$ egenvärde

$(A - I)x = 0$ har icke-trivial lösn. $x = Ax$

x bevaras och är rotationsaxeln.

Resterande reduceras till 2D-fallet, efter ett basbyte med en axel parallell med x.

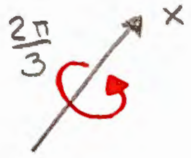
Exempel

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{ortagonal } 3\times 3 \text{ m.} \quad \det A = 1$$

Hitta rotationsaxeln.

Lösning: Hitta egenvärdet, d.v.s. lös $(A - I)x = 0$

$$\Rightarrow \ldots \Rightarrow x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} t \quad t \in \mathbb{R}$$



Bonus: $y = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \perp x$. Hur vrids y ?

$$\cos\theta := \frac{Ay \cdot y}{\|Ay\|\,\|y\|} = \frac{-1}{\sqrt{2}\sqrt{2}} = -\frac{1}{2} \Rightarrow \theta = \frac{2\pi}{3}$$

Matrisnormer

Frobeniusnorm (Mindre vanlig)

$$\|A\|_F := \sqrt{sp(A^TA)} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}$$

Def Låt A vara en $m \times n$-matris.

NORMEN av A är

$$\|A\| = \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{\|Ax\|}{\|x\|} \overset{(*)}{=} \sup_{\|x\|=1} \|Ax\|$$

((*) ty $\|A\alpha x\| = \alpha\|Ax\|$ speciellt då $\alpha = 1/\|x\|$)

Anm. $\|Ax\|$ kontinuerlig funktion av x och $\|x\| = 1$ är kompakt => sup antas => sup = max

Så $\|A\| = \max_{\|x\|=1} \|Ax\|$

Egenskaper A, B, C matriser

$\|A+B\| \leq \|A\| + \|B\|$ (triangelolikheten)

$\|\alpha A\| = |\alpha|\|A\|$

$\|A\| \geq 0$ likhet omm. $A = \mathbb{0}$

$\|Ax\| \leq \|A\|\,\|x\|$

$\|AC\| \leq \|A\|\,\|C\|$

$\|A^T\| = \|A\|$

$\|P\| = 1$ om P ortogonal

Diagonalisering

Repetition

A $n \times n$-matris, $x = 0$ egenvektor med egenvärde $\lambda \iff Ax = \lambda x$. Egenvärdena är rötter till kar. ekv. $\det(A - \lambda I) = 0$

Egenvektorer är lätta att avbilda. Vi vill därför hitta en bas av egenvektorer. Antalet linjört oberoend egenvektorer för en matris är vesäntligt.

Exempel

Låt $a \in \mathbb{C}$ och $A = \begin{pmatrix} a & 1 \\ 0 & a \end{pmatrix}$. Sök egenvärden och egenvektorer. Lösning:

$$\det(A - \lambda I) = \begin{vmatrix} a-\lambda & 1 \\ 0 & a-\lambda \end{vmatrix} = (a-\lambda)^2 \Rightarrow \lambda_1 = \lambda_2 = a$$

Egenvektorer: $(A - aI)x = 0 \iff \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$

$$\Rightarrow x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} t \quad t \in \mathbb{C}$$

Endast 1 linj. ober egenvektor. $\dim E(a) = 1$

(Men $\lambda = a$ är en dubbelrot till kar. ekv.)

Slutsats: Bas för $\mathbb{R}^2$ av egenvektorer till A existerar inte. ☹

Def För ett egenvärde λ_k är den ALGEBRAISKA MULTIPLICITETEN m_a multipliciteten av roten λ_k i det kar. pol. (d.v.s. kar. pol. har faktor $(\lambda_k - \lambda)^{m_a}$.)

Den GEOMETRISKA MULTIPLICITETEN är $m_g = \dim(E(\lambda_k))$.

(d.v.s max antal lin. ober. egenvektorer till λ_k)

Lemma 4.1

$m_g \leq m_a$.

Bevis m.h.a. variabelbyte

Anm. Om $m_g < m_a$ kallas λ_k DEFEKT.

Lana 200422

Diagonalisering

Repetition

Om $(\lambda_k - \lambda)^{m_a}$ är faktor i kar. pol. $\det(A - \lambda I)$ så är m_a algebraiska multipliciteten.

$m_g = \dim E(\lambda_k)$ är geometriska multipliciteten.

Def En $n \times n$-matris A är DIAGONALISERBAR (kan DIAGONALISERAS) om A är similär med en diagonal matris D, d.v.s. $D = T^{-1}AT$ för någon inverterbar matris T.

Diagonalisering kan göras m.h.a. en bas av egenvektorer.

Sats [4.5]

Antag att A är en $n \times n$-matris med n linjärt oberoende egenvektorer $\{e_1, \dots, e_n\}$ med tillhörande egenvärden $\{\lambda_1, \dots, \lambda_n\}$. Då är A diagonaliserbar, $T^{-1}AT = D$ med

$$T = \begin{pmatrix} | & & | \\ e_1 & \dots & e_n \\ | & & | \end{pmatrix} \text{ och } D = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}$$

Varje egenvärde förekommer i D lika många gånger som den algebraiska multipliciteten.

Bevis

Låt T och D vara som i satsen.

T har linj. ober. kolonner och är kvadratisk

$\Rightarrow$ T inverterbar.

$$AT = A\begin{pmatrix} | & & | \\ e_1, & \dots, & e_n \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ Ae_1 & \dots & Ae_n \\ | & & | \end{pmatrix} \overset{\text{egenvektorer}}{=} \begin{pmatrix} | & & | \\ \lambda_1 e_1 & \dots & \lambda_n e_n \\ | & & | \end{pmatrix}$$

$$TD = \begin{pmatrix} | & & | \\ e_1 & \dots & e_n \\ | & & | \end{pmatrix}\begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{pmatrix} = \begin{pmatrix} | & & | \\ \lambda_1 e_1 & & \lambda_n e_n \\ | & & | \end{pmatrix}$$

Så vi har

$$AT = TD \iff T^{-1}AT = D$$

Så A och D är similära. Detta ger:

$$\det(A-\lambda I) = \det(D-\lambda I) = (\lambda_1-\lambda)\dots(\lambda_n-\lambda)$$

$\Rightarrow$ m_a är antal ggr egenvärdet förekommer i D.

▨

Sats [4.6]

Om $\lambda_1, \dots, \lambda_k$ är olika egenvärden och $e_1, \dots, e_k$ motsvarande egenvektorer så är $e_1, \dots, e_k$ linjärt oberoende.

Bevis Antag att $e_1, \dots, e_k$ är linj. ber.

Låt m = max antal linj. ober vektorer bland $e_1, \dots, e_k$. Efter omnumrering antag att $e_1, \dots, e_m$ är linj. ober.

$$\Rightarrow e_k = \alpha_1 e_1 + \dots + \alpha_m e_m \quad (*)$$

$$A(*): \quad \lambda_k e_k = \alpha_1 \lambda_1 e_1 + \dots + \alpha_1 \lambda_m e_m$$

$$\lambda_k(*): \quad \lambda_k e_k = \alpha_1 \lambda_k e_1 + \dots + \alpha_1 \lambda_k e_m$$

⟶

⟶

Sätt de två uttrycken lika:

$0 = \alpha_1(\lambda_1 - \lambda_k)e_1 + \dots + \alpha_m(\lambda_m - \lambda_k)e_m$

Men $e_1, \dots, e_m$ linj. ober,

$\lambda_i - \lambda_k \neq 0 \quad \forall i = 1, \dots, m < k$ enl. ant.

Så $\alpha_1 = \alpha_2 = \dots = \alpha_m = 0$

Sätt in i $(*)$ $\Rightarrow$ $e_k = 0$

Motsägelse ty e_k är egenvektor!

Så $e_1, \dots, e_k$ måste vara lin. ober. ▨

Sats [4.7]

Om $n \times n$-matrisen A har n olika egenvärden (d.v.s. alla egenvärden har $m_a = 1$) så kan A diagonaliseras.

Bevis

Sats [4.6] ger att de n egenvektorerna som motsvarar egenvärdena är linj. ober.

$\Rightarrow$ utgör en bas

Sats [4.5] ger diagonaliseringen.

▨

Om alg. mult. $m_a = 1$ för alla egenvärden så kan vi diagonalisera. Detta är dock inte nödvändigt.

Sats [4.8]

A är diagonaliserbar omm alg. mult m_a är lika med geom. mult m_g för varje egenvärde. D.v.s. inga defekta egenvärden.

Exempel $\begin{pmatrix} a & 1 \\ 0 & a \end{pmatrix}$ ej diagonaliserbar ty a dubbelrot men $\dim(E(a)) = 1$.

Def En linj. avb. $F: V \to V$ kallas SYMMETRISK om $\langle u, F(v)\rangle = \langle F(u), v\rangle \ \forall u,v \in V$

En matris kallas SYMMETRISK om $A^T = A$.

Sats [4.9]

OBS

Om $\mathbf{e}$ är en ON-bas för V, $F: V \to V$ lin. avb. med matris A i basen $\mathbf{e}$ så är F symmetrisk omm A är symmetrisk.

Bevis

Med $x = [u]_{\mathbf{e}}$, $y = [v]_{\mathbf{e}}$ $\langle u, v\rangle \underset{\mathbf{e}\text{ ON-bas}}{=} x^T y$.

F symmetrisk

$\iff \langle u, F(v)\rangle = \langle F(u), v\rangle$

$\iff x^T A y = (Ax)^T y = x^T A^T y \quad \forall x, y$

$\iff A = A^T$

▨

Sats [4.10]

Om A är en reell symmetrisk $n \times n$-matris så är alla egenvärden till A reella.

Bevis Låt λ vara egenvärde till A med egenvektor x (eventuellt komplexa).

$$Ax = \lambda x \underset{A \text{ reell}}{\overset{\text{komplex-konjugat}}{\Longrightarrow}} A\bar{x} = \bar{\lambda}\bar{x} \overset{\text{trans-ponera}}{\Longrightarrow} (A\bar{x})^T = \bar{\lambda}\bar{x}^T$$

A sym.

$$\Longrightarrow \bar{x}^T A = \bar{\lambda}\bar{x}^T \overset{\text{mult, m. } x}{\Longrightarrow} \bar{x}^T \underbrace{Ax}_{\lambda x} = \bar{\lambda}\bar{x}^T x$$

$$\Longrightarrow \lambda \bar{x}^T x = \bar{\lambda}\bar{x}^T x$$

Antingen $\bar{x}^T x = 0$ eller $\lambda = \bar{\lambda}$. Men

$$\bar{x}^T x = \bar{x}_1 x_1 + \dots + \bar{x}_n x_n = |x_1|^2 + \dots + |x_n|^2 > 0$$

$\Longrightarrow \lambda = \bar{\lambda}$ d.v.s. λ reell ▨

$E(\lambda)$ fås av att lösa $\overbrace{(A - \lambda I)}^{\text{reell}} x = 0$ så $E(\lambda)$ har en bas av reella vektorer: Reell Gausselimination ger minst en parameter $\Longrightarrow$ minst en lösning $x \neq 0$.

Sats [4.11]

Om F är symmetrisk och u_1, u_2 är egenvektorer för F med egenvärden $\lambda_1 \neq \lambda_2$ så är u_1 och u_2 ortogonala.

Bevis $\langle u_1, \overbrace{F(u_2)}^{\lambda_2 u_2} \rangle = \langle \overbrace{F(u_1)}^{\lambda_1 u_1}, u_2 \rangle$

$$\Longleftrightarrow \lambda_2 \langle u_1, u_2 \rangle = \lambda_1 \langle u_1, u_2 \rangle$$

$$\Longleftrightarrow \underbrace{(\lambda_2 - \lambda_1)}_{\neq 0} \langle u_1; u_2 \rangle = 0 \Longleftrightarrow u_1 \perp u_2 \qquad \square$$

Spektralsatsen

Formulering 1:

Sats [4.12] Spektralsatsen

Om $F: V \to V$ symmetrisk och V ändlig dim. så finns en ON-bas för V bestående av egenvektorer till F.

Formulering 2:

Sats [4.13]

Om A är en reell symmetrisk matris så finns en ortogonal matris T och en diagonal matris D så att $T^{-1}AT = D$. Egenvärdena utgör diagonalelementen i D och kolonnerna i T är egenvektorerna.

Bevisidé (Bevis i boken) (Formulering 1)

Karaktäristiska polynomet har minst en rot λ_1

λ_1 är reell enligt sats 4.10

Reell Gausselimination ger motsvarande reell egenvektor u_1.

Bilda första baselementet $e_1 = \frac{u_1}{\|u_1\|}$.

Låt $U = \text{span}\{e_1\}^{\perp}$

Om $u \in U$ får vi $\langle F(u), e_1 \rangle \overset{F \text{ sym.}}{=} \langle u, F(e_1) \rangle \overset{F(e_1) = \lambda_1 e_1}{=}$

$$= \lambda_1 \underbrace{\langle u, e_1 \rangle}_{0 \text{ ty } u \in U} = 0$$

Så $F(u) \in U$. F ger lin. avb. $F: U \to U$

$\dim U = \dim V - 1$. Upprepa argument m. U ist. för V.

Vi får till slut ON bas till V. ▨

Exempel

Ortogonalt diagonalisera $A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

Sök egenvärden:

$$\det(A - \lambda I) = \begin{vmatrix} 2-\lambda & 1 & 1 \\ 1 & 2-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{vmatrix} = \dots$$

$$\dots = (4-\lambda)(1-\lambda)^2 = 0$$

Egenvärden $\lambda_1 = 4$, $\lambda_2 = 1$ med mult. 2.

Sök egenvektorer:

$$(A-4I)x = 0 \iff \left[\begin{array}{ccc|c} -2 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -2 & 0 \end{array}\right] \sim$$

$$\sim \left[\begin{array}{ccc|c} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \iff \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{v_1} x_3$$

$$(A-I)x = 0 \iff \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array}\right] \sim$$

$$\sim \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \iff \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}}_{v_2} x_1 + \underbrace{\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}}_{v_3} x_3$$

v_1 egenvektor med egenvärde 4.

v_2 & v_3 egenvektorer med egenvärde 1.

Sats [4.11] ger $v_1 \cdot v_2 = 0$ & $v_1 \cdot v_3 = 0$ ty olika egenvärden. $v_2 \cdot v_3 = 1 \neq 0$ så vi använder Gram-Schmidt för ON-bas.

$$\Rightarrow e_1 = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad e_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \quad e_3 = \frac{1}{\sqrt{6}}\begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}$$

$$\Rightarrow T = \begin{pmatrix} | & | & | \\ e_1 & e_2 & e_3 \\ | & | & | \end{pmatrix} \quad D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad D = T^TAT \,//$$

Lana 200427

Tillämpningar av diagonalisering

Rep: A diagonaliserbar

$\Leftrightarrow \exists T, D: \quad D = T^{-1}AT$ eller $A = TDT^{-1}$

där D diagonal m. egenvärden på diagonalen och T har motsvarande egenvektorer som kolonner.

Om $A^T = A$ $\exists$ diagonalisering och T ortogonal

$\Rightarrow A = TDT^T$

Diagonalisering i många sammanhang.

Ex: geometrisk tolkning av lin. avb.

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} = TDT^{-1} \quad D = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{3} & 0 & 2/\sqrt{6} \end{bmatrix}$$

A ger skalning med faktor 4 i riktning $\frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

(oförändrat i ortogonala riktningar)

Ex: matrispotenser, beräkna A^k

$T^{-1}AT = D \Leftrightarrow A = TDT^{-1}$

$$A^k = (TD\underbrace{T^{-1})(T}_{I}DT^{-1})\ldots(TDT^{-1}) = TDD\ldots DT^{-1}$$

$$= TD^kT^{-1}$$

$$D^k = \begin{pmatrix} \lambda_1^k & & 0 \\ & \ddots & \\ 0 & & \lambda_n^k \end{pmatrix}$$ eftersom D diagonal.

Ex: System av linjära differentialekvationer av ordning 1 med konstanta koeff.

$$\begin{cases} x_1(t) = a_{11}x_1(t) + \dots + a_{1n}x_n(t) \\ \vdots \\ x_n(t) = a_{n1}x_1(t) + \dots + a_{nn}x_n(t) \end{cases} \iff x'(t) = A\,x(t)$$

vektorvärda funktioner

Begynnelsevärden $x(0) = x_0 \in \mathbb{R}^n$

Systemet är kopplat och kan vara svårlöst, men om A är diagonaliserbar, $A = TDT^{-1}$:

Gör variabelbytet $y(t) = T^{-1}x(t) \iff x(t) = Ty(t)$

$x'(t) = Ax(t) \iff$

$$\underline{y'(t)} = T^{-1}x'(t) = T^{-1}Ax(t) = T^{-1}ATy(t) = \underline{Dy(t)}$$

Lätt!

$$\iff \begin{cases} y_1'(t) = \lambda_1 y_1(t) \\ \dots \\ y_n'(t) = \lambda_n y_n'(t) \end{cases} \iff \begin{cases} y_1(t) = c_1 e^{\lambda_1 t} \\ \dots \\ y_n(t) = c_n e^{\lambda_n t} \end{cases}$$

Löst!

där $\begin{pmatrix} c_1 \\ \dots \\ c_n \end{pmatrix} = y(0) = T^{-1}x(0) = T^{-1}x_0$

Så $c_1, \dots, c_n$ ges av $T\begin{pmatrix} c_1 \\ c_n \end{pmatrix} = x_0$

$\underline{\text{Lösningen till systemet: } x(t) = Ty(t).}$

Om $T = (u_1, \dots, u_n)$ (egenvektorer) blir

$$x(t) = Ty(t) = (u_1, \dots, u_n)y(t)$$
$$= c_1 e^{\lambda_1 t}u_1 + \dots + c_n e^{\lambda_n t}u_n$$

Utan begynnelsevillkor blir $\{e^{\lambda_1 t}, \dots, e^{\lambda_n t}\}$ en bas för systemets lösningsrum.

$\underline{\text{OBS}}$: Behöver ej T^{-1}, bara lösn. till $T\begin{pmatrix} c_1 \\ c_n \end{pmatrix} = x_0$

$\underline{\text{OBS}}$: A behöver bara vara diagonaliserbar, inte symmetrisk. Dock kan vi få kplxa egenvärden => kplxa exp. funktioner.

Lite mer om vektor- och matrisnormer

Def För $x \in \mathbb{R}^n$ definierar vi normerna:

- Summanormen / 1-normen:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

- Euklidiska normen / 2-normen:

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2} = \sqrt{x^T x}$$

- Maximumnormen, ∞-normen:

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

Hittills har vi använt $\|x\| = \|x\|_2$

I matlab: norm(x, 1), norm(x, 2), norm(x, Inf)

Varje vektornorm ger en matrisnorm:

$$\|A\|_\star = \max_{x \neq 0} \frac{\|Ax\|_\star}{\|x\|_\star} \quad \text{där } \star = 1, 2, \infty$$

Summanorm:

$$\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$ (maximala kolonnsumman till belopp)

Euklidisk norm:

$$\|A\|_2 = \sqrt{\lambda_1}$$ där λ_i största egenvärde till A^TA.

Maximumnormen:

$$\|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$ (maximala radsumman till belopp)

2-norm "bäst", 1 o ∞ kan vara snabbast,
mer behändiga vid beräkning

Kvadratiska former

Def En KVADRATISK FORM på $\mathbb{R}^n$ är en funktion, $Q : \mathbb{R}^n \to \mathbb{R}$ som kan skrivas på formen $Q(x) = x^T A x$ där A är en symmetrisk $n \times n$-matris.

d.v.s. $Q(x_1, \dots, x_n)$ är ett andragrads-polynom i $x_1, \dots, x_n$ med bara andra-gradstermer.

Ex: $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$ diagonalelement

$$Q(x) = (x_1 \ x_2) \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2x_1^2 + 3x_2^2$$

summa av 2 (-1):or

$$A = \begin{pmatrix} 3 & -1 \\ -1 & 5 \end{pmatrix} \Rightarrow Q(x) = \dots = 3x_1^2 - 2x_1x_2 + 5x_2^2$$

Ex: Söker A givet Q:

$$Q(x) = 2x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3 + 2x_3x_1$$

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

OBS: Finns fler A så $x^T A x = Q$ men bara en symmetrisk.

Vi väljer att alltid ha A symmetrisk för enkelhetens skull. Kan då diagonalisera.

Vi vill skriva Q som en summa av kvadrater för att avgöra dess form. Q är en summa av kvadrater om A är diagonal.

Vi vet sedan tidigare $A = TDT^T$ med

$$D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad T = \begin{pmatrix} 1/\sqrt{3} & -1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{3} & 0 & 2/\sqrt{6} \end{pmatrix}$$

→

$\longrightarrow$

Inför variabelbytet $y = T^T x \iff x = Ty$

$$Q = \underbrace{x^TAx}_{\text{Blandade termer}} = (Ty)^TATy = = y^TT^TATy = \underbrace{y^TDy}_{\text{bara kvadrater!}}$$

OBS: Q är en summa av kvadrater så

$Q \geq 0$ med likhet omm $y = 0 \iff x = 0$

Kallas POSITIVT DEFINIT.

Def En kvadratisk form $Q(x)$ är

- POSITIVT DEFINIT om $Q(x) > 0 \quad \forall x \neq 0$
  (positivt semidefinit om $Q(x) \geq 0$)
- NEGATIVT DEFINIT om $Q(x) < 0 \quad \forall x \neq 0$
  (negativt semidefinit om $Q(x) \leq 0$)
- INDEFINIT om Q antar både positiva och negativa värden.

Sats [4.14]

$Q(x) = x^TAx$ är

a) pos. def. omm alla egenvärden är positiva

b) neg. def. omm alla egenvärden är negativa

c) indefinit omm A har både positiva och negativa egenvärden.

semidefinit tillåter egenvärde 0.

Bevisidé: A symetrisk ⇒ diagonaliserbar

$Q = x^TAx = y^TDy = \lambda_1 y_1^2 + \ldots + \lambda_n y_n^2$

T inverterbar ger $x = 0$ omm $y = 0$

Tecken på Q avgörs av tecken på λ_i.

Sats [4.15]

A reell symmetrisk med λ_{min}, λ_{max} största resp. minsta egenvärde. Då är

$$\lambda_{min} \|x\|^2 \le x^T A x \le \lambda_{max} \|x\|^2$$

Likhet för motsvarande egenvektorer.

Bevisidé:

$$x^T A x = y^T D y = \lambda_1 y_1^2 + \ldots + \lambda_1 y_2^2 \le$$
$$\le \lambda_{max}(y_1^2 + \ldots - + y_2^2) = \lambda_{max} \|y\|^2$$

Men $\|y\|^2 = \|Tx\|^2 = \|x\|^2$ ty T ortagonal

Så $x^T A x \le \lambda_{max} \|x\|^2$ och p.s.s. för min.

Tillämpning för matrisnormer:

$$\|A\| = \max_{x \ne 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

$$\|Ax\|^2 = (Ax)^T Ax = x^T A^T A x \overset{\text{Satsen}}{\le} \lambda_1 \|x\|^2$$

där λ_1 är största egenvärdet av den symmetriska matrisen $A^T A$.

Låt $\|x\| = 1$ och x egenvektor till största egenvärdet:

$$\|A\| = \sqrt{\lambda_1} \quad \text{där } \lambda_1 = \lambda_{max} \text{ för } A^T A.$$

($\|A\|$ betecknar euklidiska normen av A)

Lana 200504

Numerisk lösning av icke-linjära ekv. sys.

Repetition

Söker rot x^* till ekvationen $f(x)=0$.

Någon metod ger talföljd $\{x_k\}_{k=0}^{\infty}$ som

som konvergerar om $x_k \to x^*$ med

konvergensordning $q>0$ om

$$\lim_{k\to\infty} \frac{|x_{k+1}-x^*|}{|x_k - x^*|^q} = C < \infty$$

Newtons metod $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ $k=0,1,\dots$

konvergerar lokalt kvadratiskt $(q=2)$

mot enkelrötter.

System av icke-linjära ekvationer

Studera system $\underline{f}(\underline{x})=0$, $\underline{f}: \mathbb{R}^n \to \mathbb{R}^n$

(lika många ekvationer som variabler)

Istället för $f'(x)$ får vi Jacobianmatrisen $J(\underline{x})$.

$$(J(\underline{x}))_{ij} := \frac{\partial f_i}{\partial x_j}(\underline{x})$$

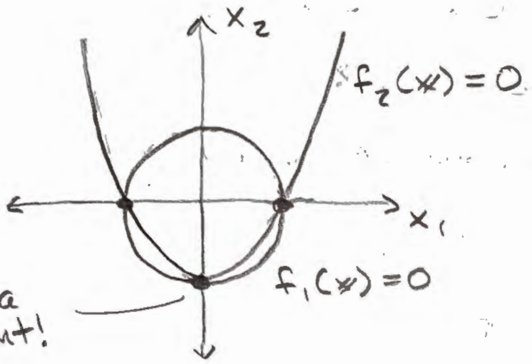
En rot $\underline{x}^* \in \mathbb{R}^n$ kallas SINGULÄR om

$J(\underline{x}^*)$ är singulär (ej inverterbar matris).

Annars kallas $\underline{x}^*$ REGULJÄR.

Exempel $f : \mathbb{R}^2 \to \mathbb{R}^2 \quad x \in \mathbb{R}^2$

$$f(\underline{x}) = \begin{pmatrix} x_1^2 + x_2^2 - 1 \\ x_1^2 - x_2 - 1 \end{pmatrix} = \begin{pmatrix} f_1(\underline{x}) \\ f_2(\underline{x}) \end{pmatrix}$$



samma gradient!

Vi får grafiskt tre rötter till $f(x)=0$:

$$\overset{1}{\underline{x}}{}^* = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \overset{2}{\underline{x}}{}^* = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \overset{3}{\underline{x}}{}^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$J(\underline{x}) = \begin{pmatrix} 2x_1 & 2x_2 \\ 2x_1 & -1 \end{pmatrix}$$

$$J(\overset{1}{\underline{x}}{}^*) = \begin{pmatrix} -2 & 0 \\ -2 & -1 \end{pmatrix}$$ inverterbar

$\Rightarrow \overset{1}{x}{}^*$ reguljär rot

$$J(\overset{2}{\underline{x}}{}^*) = \begin{pmatrix} 0 & -2 \\ 0 & -1 \end{pmatrix}$$ ej inverterbar

$\Rightarrow \overset{2}{x}{}^*$ singulär rot

Grafiskt ser vi detta på att $f_1(\underline{x})=0$ och $f_2(\underline{x})=0$ tangerar varandra i $\overset{2}{x}{}^*$.

$$J(\overset{3}{\underline{x}}{}^*) = \begin{pmatrix} 2 & 0 \\ 2 & -1 \end{pmatrix}$$ inverterbar

$\Rightarrow \overset{3}{x}{}^*$ reguljär rot.

Newtons metod för system

Idé: Flervariabel taylorutveckling kring approximativ rot $\mathbf{x}_k$. (k iterationsindex)

$$f_i(\mathbf{x}) \approx f_i(\mathbf{x}_k) + \nabla f_i(\mathbf{x}_k)^T(\mathbf{x}-\mathbf{x}_k)$$

$$\Rightarrow \mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$$

En linjär modell: $\mathbf{0} = \mathbf{f}(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k)$

med lösning $\mathbf{x}_{k+1}$ ger, med $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$

$$\boxed{J(\mathbf{x}_k)\mathbf{s}_k = -\mathbf{f}(\mathbf{x}_k)}$$

Detta är vår nya Newtons metod.

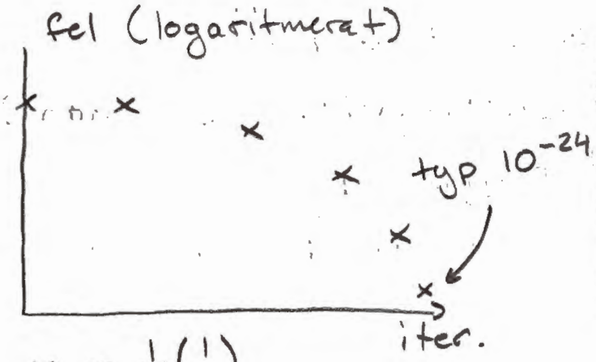
Lös ekvationssystemet

$\Rightarrow$ för "sökriktning" $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$

Likheter med envariabelfallet:

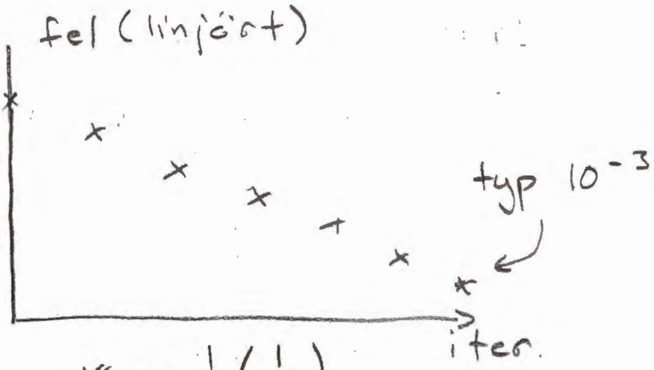
- Endast lokal konvergens
- Kvadratisk konvergens mot reguljär rot, annars linjär konvergens.

För vårat exempel: $\mathbf{x}_0 = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}$ ger kvadratisk konvergens mot $\overset{3}{\mathbf{x}}{}^*$ men $\mathbf{x}_0 = \frac{1}{2}\begin{pmatrix}1\\-2\end{pmatrix}$ ger linjär konvergens mot $\overset{2}{\mathbf{x}}{}^*$



$\mathbf{x}_0 = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}$

$\mathbf{x}^* = \begin{pmatrix}1\\0\end{pmatrix}$



$\mathbf{x}_0 = \frac{1}{2}\begin{pmatrix}1\\-2\end{pmatrix}$

$\mathbf{x}^* = \begin{pmatrix}0\\-1\end{pmatrix}$

Alternativ till Newtons metod

Newtons metod har problem:

- **Endast lokal konvergens**

Divergens beror ofta på att

$\|s_k\|$ ej avtar => "går förbi lösningen"

Lösning: Modifiera steglängden

Dämpad Newtons metod

$$\boxed{\begin{array}{c} J(x)\, s_k = -f(x_k), \quad x_{k+1} = x_k + \alpha_k s_k \\ \text{där } \alpha_k \leq 1 \text{ (dämpar steglängd)} \end{array}}$$

Välj varje α_k s.a. $\|f(x_{k+1})\| < \|f(x_k)\|$ (*)

(d.v.s. ny punkt alltid bättre än gammal)

Kan t.ex. halvera α_k tills (*) uppfylls

OBS: $\alpha_k < 1$ ger långsammare konvergens

=> Övergå till $\alpha_k = 1$ nära roten för att bibehålla Newtons konvergenshastighet.

- **Newtons metod kräver lösning av ett nytt ekvationssystem för varje iteration.**

Långsamt för stora matriser.

Det är bättre om bara HL uppdateras

Kan använda LU-faktorisering (14 maj)

Faktorisering kräver $\mathcal{O}(n^3)$ operationer

Varje nytt högerled kräver $\mathcal{O}(n^2)$ operationer.

Använd $J(x_0)$ istället för $J(x_k)$

(samma matris varje iteration)

→

$$\longrightarrow \boxed{J(x_0)\, s_k = -f(x_k) \qquad x_{k+1} = x_k + s_k}$$

Ej kvadratisk konvergens

=> fler iterationer (men varje iter. snabb)

Newtons metod kräver känd Jacobian

Broydens metod

Om B_k approximerar $J(x_k)$ får vi

kvasi-Newtonmetoder:

$$\boxed{B_k s_k = -f(x_k) \qquad x_{k+1} = x_k + s_k}$$

Bra om $J(x_k)$ är svår/omöjlig att beräkna

Börja med $B_0 = J(x_0)$ eller en approximation

Broydens: Uppdatera bara B_{k+1} i riktning s_k

$$B_{k+1} s_k \approx J(x_{k+1})\, s_k$$

Taylorutv. kring x_{k+1}:

$$f(x_k) \approx f(x_{k+1}) + J(x_{k+1})\overbrace{(x_k - x_{k+1})}^{-s_k}$$

$$\Longleftrightarrow \quad f(x_{k+1}) - f(x_k) \approx J(x_{k+1})\, s_k$$

Vill alltså ha $B_{k+1} s_k = f(x_{k+1}) - f(x_k)$

men oförändrad i ortogonala riktningar:

$$B_{k+1} z = B_k z \qquad \forall z : s_k \cdot z = 0$$

Detta ger Broydens metod:

$$\boxed{B_{k+1} = B_k + \frac{1}{\|s_k\|^2}\left(f(x_{k+1}) - f(x_k) - B_k s_k \right) s_k^T}$$

Vi slipper beräkna derivator!

(Jämför med sekantmetoden)

Fixpunktsiterationer

Som i envariabelfallet:

$f(x) = 0$ skrivs om som $x = g(x)$

$f, g : \mathbb{R}^n \to \mathbb{R}^n$ $x \in \mathbb{R}^n$

ger metoden $\boxed{x_{k+1} = g(x_k)}$

När konvergerar metoden?

Om $G(x)$ är Jacobianen för $g(x)$ får vi lokal konvergens om

matris-norm $\longrightarrow$ $\| G(x) \| \leq \mu < 1$ nära x^*

Feluppskattning:

$$\boxed{\begin{array}{l} \| x_{k+1} - x^* \| \leq \frac{\mu}{1-\mu} \| x_{k+1} - x_k \| \\ \text{för } x_k \text{ nära } x^* \end{array}}$$

Metodoberoende feluppskattning

Som i envariabelfallet (nästan)

$\delta \mathbf{x} = \tilde{\mathbf{x}} - \mathbf{x}^*$ $\quad \tilde{\mathbf{x}} \in \mathbb{R}^n$ approximation

Taylorutveckling kring $\mathbf{x}^*$:

$$f(\hat{\mathbf{x}}) = f(\mathbf{x}^* + \delta \mathbf{x}) =$$
$$= f(\mathbf{x}^*) + J(\mathbf{x}^*)\delta \mathbf{x} + \mathcal{O}(\|\delta \mathbf{x}\|^2) \approx$$
$$\approx J(\mathbf{x}^*)\delta \mathbf{x}$$

Vid reguljär rot existerar $J(\mathbf{x}^*)^{-1}$

$$\Rightarrow \boxed{\delta \mathbf{x} \approx J(\mathbf{x}^*)^{-1} f(\tilde{\mathbf{x}}) \approx J(\tilde{\mathbf{x}})^{-1} f(\hat{\mathbf{x}})}$$

Metodoberoende feluppskattning

I MATLAB

fzero söker rot till $f(x)=0$ i envariabelfallet

fsolve motsvarande för flervariabelfallet.

Lana 200505

Funktionsapproximation, interpolation & splines

Approximera funktioner med polynom av en viss grad, alt. styckvisa polynom (inkl. styckvis linjära funktioner)

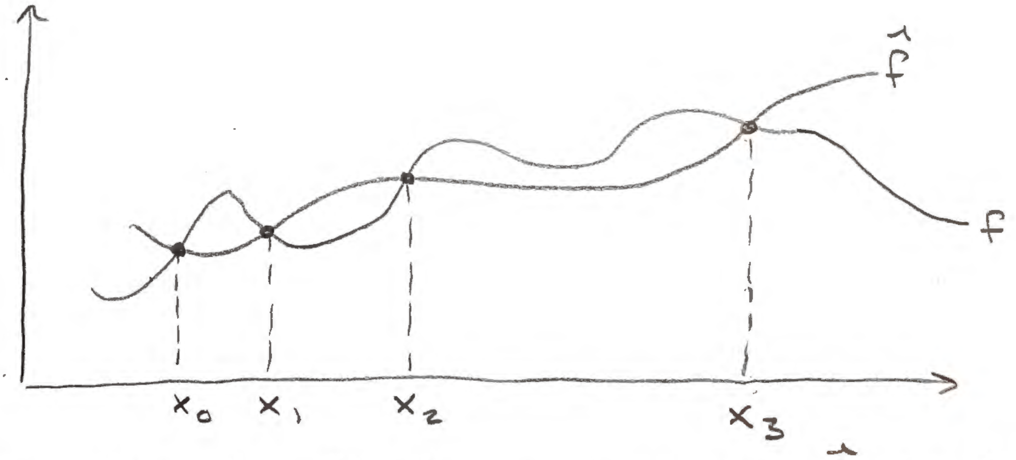
- Ger en ändlig representation
- Snabba att evaluera
- Enkla att derivera och integrera

Polynominterpolation

En funktion f approximeras med $\hat{f}$ så att de stämmer överens i vissa punkter.

Def Om $f: \mathbb{R} \to \mathbb{R}$ och $x_0, \dots, x_n$ är $n+1$ st punkter och $\hat{f}(x_i) = f(x_i)$ $i = 0, \dots, n$ sägs $\hat{f}$ INTERPOLERA f i punkterna $x_0, \dots, x_n$.

Med $f_i = f(x_i)$ och $\hat{f}_i = \hat{f}(x_i)$ har vi interpolationsvillkoren $\hat{f}_i = f_i$ $i = 0, \dots, n$



OBS: för de flesta punkter är $\hat{f}(x) \neq f(x)$ kan dock gälla att $f(x) \equiv \hat{f}(x)$

"Sats"

Det finns entydigt polynom p_n med grad $p_n \leq n$ som går genom de $n+1$ punkterna (x_i, f_i) $i = 0, \dots, n$.

OBS: $x_0, \dots, x_n$ måste vara olika

p_n kallas INTERPOLATIONSPOLYNOMET till f i punkterna.

Förklaring

p_n entydigt, ty om $\tilde{p}_n$ uppfyller samma villkor har $p_n - \tilde{p}_n$ nollställen $x_0, \dots, x_n$ ($n+1$:st) men det är ett polynom med grad $\leq n$.

$\Rightarrow$ $p_n - \tilde{p}_n \equiv 0$ $\Rightarrow$ $p_n \equiv \tilde{p}_n$

Olika konstruktioner

- Lagranges form (ej praktisk)
- Newtons form

Ansats:

$$p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \dots + \\ + \dots + c_n(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

OBS: $p_n(x)$ har grad $\leq n$

Interpolationsvillkoren ger $p_n(x_i) = f_i$

$$\begin{cases} f_0 = p_n(x_0) = c_0 \\ f_1 = p_n(x_1) = c_0 + c_1(x_1 - x_0) \\ f_2 = p_n(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2-x_0)(x_2-x_1) \\ \vdots \\ f_n = p_n(x_n) = c_0 + \dots + c_n(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1}) \end{cases}$$

Lös med framåtsubstitution!

Egenskaper för Newtons interpolationsmetod:

- Enkelt att lägga till en punkt
  (c_i tidigare kan återanvändas)
- För evaluering krävs $0+1+2+\dots+n = \frac{n(n+1)}{2}$
  multiplikationer. Men!

  Skriv om:

  $$p_n(x) = (x-x_0)(c_1+(x-x_1)(c_1+(x-x_2)\dots)))\dots)$$

  Nu har vi bara n multiplikationer!

  $\longrightarrow$ Snabb evaluering

Exempel

Interpolera $f(x) = \frac{3x+1}{1+x^2}$ i $(x_0, x_1, x_2, x_3) = (-1, 0, 1, 3)$

Vi får $f_0 = -1$, $f_1 = 1$, $f_2 = 2$, $f_3 = 1$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 1 & 4 & 12 & 24 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ -1/2 \\ 0 \end{pmatrix}$$

$$\Rightarrow \hat{f}(x) = c_0 + (x-x_0)(c_1 + (x-x_1)(c_2 + (x-x_2)c_3)) =$$

$$= \dots = -1 + (x-1)\left(2 - \frac{1}{2}x\right) =$$

$$= -1 + 2x - \frac{1}{2}x^2 - 2 + \frac{1}{2}x =$$

$$= -\frac{1}{2}x^2 + \frac{5}{2}x - 3$$

Interpolationsfel

Fel p.g.a. ändlig representation kallas ofta trunkeringsfel. I vårt fall interpolationsfel

Sats

Om $f \in C^{n+1}$ ($n+1$ kont. derivator) och p_n interpolerar f i $x_0, \dots, x_n$ är interpolationsfelet

$$p_n(x) - f(x) = -\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)$$

där ξ ligger mellan $x_0, x_1, \dots, x_n$, och x

d.v.s. ξ större än minsta och mindre än största.

OBS: Interpolationsfelet är 0 i $x_0, \dots, x_n$

Om f är polynom av grad $\leq n \Rightarrow p_n \equiv f$

(eftersom $f^{(n+1)}(\xi) = 0$)

Bevisidé: Låt x vara godtycklig fix punkt.

Konstruera interpolationspolynom $p_{n+1}(t)$ som även interpolerar f i punkten x.

$$p_{n+1}(t) = p_n(t) + \underbrace{c_{n+1}(x)}_{\text{beror av } x}(t-x_0)(t-x_1)\dots(t-x_n)$$

Argument med medelvärdessatsen eller Rolles sats ger

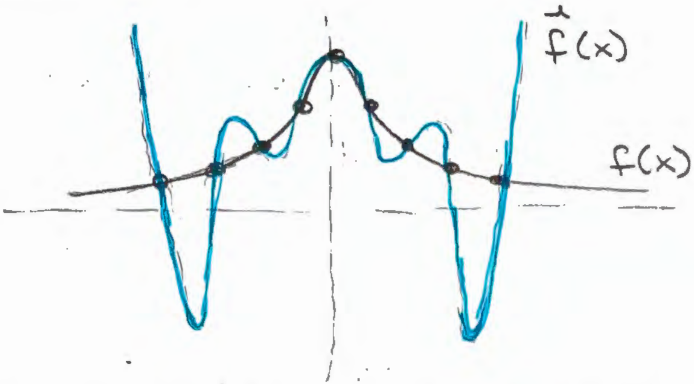
$$c_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

$f(x) = p_{n+1}(x)$ ty p_{n+1} interpolerar f i x

$\Rightarrow$ ger satsen.

Runges fenomen

Interpolationspolynom med högt gradtal ger dåligt resultat. Mellan punkterna kan felen bli stora. Utanför "punkternas intervall" blir det ännu värre.



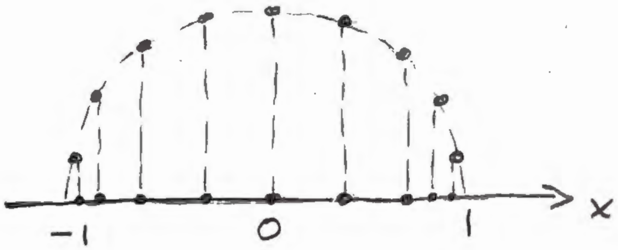
Jämt utspridda punkter är dåligt.

Chebyshev: Om interpolationspolynom på $[-1,1]$ är $x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right)$ $i = 1, \dots, n$ bästa valet för n punkter. (tätare i kanter).

Andra lösningar:

- Lägre gradtal
- Styckvisa funktioner
  - Linjär styckvis
  - Kubisk spline

Chebyshevspridningen:



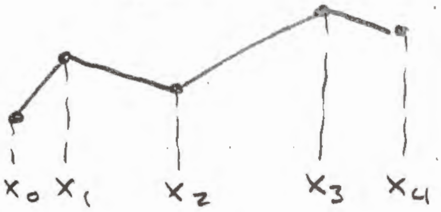
Jämt utspridda
på halvcirkel
projecerade
på x-axeln

Interpolation med splines

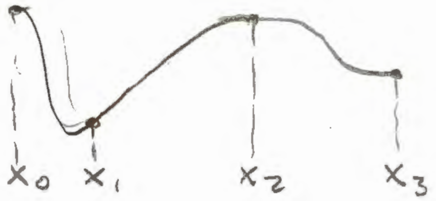
Def En SPLINE $s(x)$ av grad k är ett styckvis polynom av grad k med kontinuerliga derivator av ordning $k-1$, över integrationspunkterna.

- $k=1$: linjär spline, styckvis linjär funktion
- $k=2$: svårt (parabel lämpar sig dåligt)
- $k=3$: kubisk spline, venligast.

$k=1$



$k=3$



$s(x)$ spline av grad 3.

$$s(x) = \begin{cases} s_1(x) & x \in [x_0, x_1] \\ s_2(x) & x \in [x_1, x_2] \\ s_3(x) & x \in [x_2, x_3] \end{cases}$$

Splinevillkor:

$s_1(x_0) = f(x_0)$

$s_1(x_1) = s_2(x_1) = f(x_1)$, $s_1'(x_1) = s_1'(x_2)$, $s_1''(x_1) = s_2''(x_2$

$s_2(x_2) = s_3(x_2) = f(x_2)$ $s_2'(x_2) = s_3'(x_2)$, $s_2''(x_2) = s_3'(x_2$

$s_3(x_3) = f(x_3)$

splines är ej entydigt bestämda

Behöver *ändpunktsvillkor* eller *randvillkor*.

Olika alternativ:

1) Rätta randvillkor

$s'(x_0) = f'(x_0) \qquad s'(x_n) = f'(x_n)$

2) Naturliga randvillkor

$s''(x_0) = 0 \qquad s''(x_n) = 0$

3) Periodiska randvillkor

$s'(x_0) = s'(x_n), \quad s''(x_0) = s''(x_n)$

4) "Not a knot" / Ingen nod

$s_1'''(x_1) = s_2'''(x_1), \quad s_{n-1}'''(x_{n-1}) - s_n'''(x_{n-1})$

ger $s_1(x) = s_2(x), \quad s_{n-1}(x) = s_n(x)$

Man tar bort noderna x_1 och x_{n-1}

Kubiska splines ger ofta bra approximationer.

MATLAB:

polyfit för polynominterpolation

spline för kubisk spline (spline toolbox)

Lana 200507

Numerisk integration

Repetition

$p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) \ldots\ldots)(x-x_{-1})$

Interpolationsvillkoren $p_n(x_i) = f_i$ ger $c_0, \ldots, c_n$

Interpolationsfelet ges om $f \in C^{n+1}$ av

$$p_n(x) - f(x) = -\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)\ldots(x-x_n)$$

där ξ mellan största och minsta av $x_0, \ldots, x_n, x$

Vi vill beräkna $\int_a^b f(x)\,dx$ numeriskt.

Varför numeriskt?

- f explicit men saknar elementär primitiv
  Ex: $f(x) = e^{-x^2}$
- f inte explicit men beräkningsbar $\forall x \in [a,b]$
- f given endast i vissa punkter
- Analytisk lösning finns men ger numeriska problem.
  Ex, $\int_{1000}^{1001} \frac{1}{x^2-1}\,dx = \underbrace{\arctan(1001)}_{\underline{1,569797}73\ldots} - \underbrace{\arctan(1000)}_{\underline{1,569797}63\ldots}$

Risk för stort kancellationsproblem.

Kvadraturformler

Vill bara använda funktionsvärden

Approximera integral som viktad summa:

$$\int_a^b f(x)\,dx = \sum_{i=1}^{n} w_i f(x_i) - \underbrace{R_T}_{\text{trunkeringsfel}}$$

Kallas <u>kvadraturformler</u> (w_i för ej bero på f).

Approximation av $f(x)$ med interpolationspolynom $p_n(x)$ följt av integration av $p_n(x)$ ger

Newton-Cotes kvadraturformler

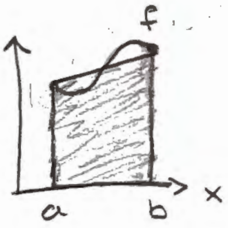
Enklaste fallet: Förstagradspolynom ($n=1$)

Interpolationspunkter a & b

$$p_1(x) = c_0 + c_1(x-a) \qquad c_0 = f(a) \quad c_1 = \frac{f(b)-f(a)}{b-a}$$

$$\int_a^b p_1(x)\,dx = \Big[c_0(x-a) + \frac{c_1}{2}(x-a)\Big]_a^b =$$

$$= c_0(b-a) + \frac{c_1}{2}(b-a)^2 = \underbrace{\frac{f(a)-f(b)}{2}}_{\text{medelvärde}}\underbrace{(b-a)}_{\text{bredd}}$$



Kallas <u>trapetsmetoden</u>

Trunkeringsfelet:

$$R_T := \int_a^b (p_1(x) - f(x))\,dx \overset{\text{formel för fel vid interpolation}}{=}$$

$$= -\int_a^b \frac{f''(\xi)}{2!}(x-a)(x-b)\,dx \overset{\text{integralkalkylens generaliserade MV-sats}}{=}$$

$\eta \in [a,b]$

$$= -\frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)\,dx = \ldots =$$

$$= \frac{f''(\eta)}{12}(b-a)^3$$

⟶

⟶

Trapetsregeln

$$\int_a^b f(x)dx = \frac{(b-a)}{2}((f(a)-f(b)) - \underbrace{\frac{f''(\eta)}{12}(b-a)^3}_{R_T}$$

för något $\eta \in [a,b]$

OBS: $R_T = 0$ om f är förstagradspolynom.

$R_T = \mathcal{O}((\text{intervallängd})^3)$

Simpsons regel

Interpolera med ett andragradspolynom $p_2(x)$
i punkterna a & b men även $\frac{a+b}{2}$

$$p_2(x) = p_1(x) + c_2(x-a)(x-b)$$

$$p_2\left(\frac{a+b}{2}\right) = f\left(\frac{a+b}{2}\right) \Rightarrow c_2 = \frac{2f(a)+2f(b)-4f\left(\frac{a+b}{2}\right)}{(b-a)^2}$$

$$\int_a^b p_2(x))dx = \int_a^b p_1(x)dx + c_2\int_a^b (x-a)(x-b)dx =$$

$$= \frac{f(a)-f(b)}{2}(b-a) - \frac{c_2}{6}(b-a)^3 =$$

$$= \frac{f(a)+4f\left(\frac{a+b}{2}\right)+f(b)}{6}(b-a)$$

Kan bestämma R_T som för trapetsregeln.

Simpsons regel

$$\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) + R_T$$

$$R_T = \frac{f^{(4)}(\eta)}{2880}(b-a)^5 \quad \text{för något } \eta \in [a,b]$$

OBS: $R_T = 0$ för kubiska polynom (oväntat!)
Blir ej bättre med $p_3(x)$ approximation.

Trapetsformeln

Dela upp $[a,b]$ i delintervall med längd $h = (b-a)/n$ och använd trapetsregeln på varje.

$$T(h) = h\left(\underbrace{\frac{f(x_0)}{2} + \frac{f(x_1)}{2}}_{[x_0,x_1]} + \underbrace{\frac{f(x_1)}{2} + \frac{f(x_2)}{2}}_{[x_1,x_2]} + \ldots + \underbrace{\frac{f(x_n)}{2}}_{[x_{n-1},x_n]}\right)$$

$$= h\left(\frac{f(x_0)}{2} + \frac{f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i)\right)$$

$$\int_a^b f(x)\,dx = T(h) + R_T$$

där $R_T = \frac{b-a}{12} h^2 f''(\xi)$ där $\xi \in [a,b]$

Som Riemannintegral men medelvärde på varje intervall!

Förklaring till R_T:

summa av fel

$$R_T = \sum_{i=0}^{n-1} \frac{h^3}{12} f''(\eta_i) \quad \text{där} \quad \eta_i \in [x_i, x_{i+1}] \subset [a,b]$$

$$\Rightarrow R_T = \frac{(b-a)h^2}{12} \cdot \underbrace{\frac{1}{n}\sum_{i=0}^{n-1} f''(\eta_i)}_{\text{medelvärde av } f''(\eta_i)}$$

Satsen om mellanliggande värde ger

$$\frac{1}{n}\sum_{i=0}^{n-1} f''(\eta_i) = f''(\xi) \quad \text{för något } \xi \in [a,b]$$

$$\Rightarrow \boxed{R_T = \frac{b-a}{12} h^2 f''(\xi)}$$

bra grej: $h \xrightarrow{n\to\infty} 0 \;\hat{=}\; R_T \xrightarrow{h\to 0} 0$

Simpsons formel

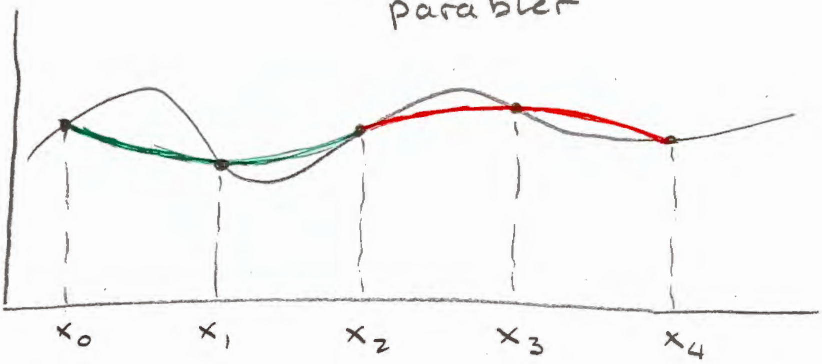
Dela upp $[a,b]$ i ett jämnt antal n delintervall med längd $h = (b-a)/n$

$$S(h) = \frac{h}{3}\left(\sum_{i=0}^{\frac{n}{2}-1} f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right) =$$

$$= \frac{h}{3}\left(f(x_0) + f(x_n) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i})\right)$$

$$\boxed{\begin{array}{l}\int_a^b f(x)\,dx = S(h) + R_T \\ R_T = \frac{b-a}{180} h^4 f^{(4)}(\xi) \text{ för ngt } \xi \in [a,b]\end{array}}$$

$n=4$ => approximera med två parabler



Richardsonextrapolation

Smart val av viktning för varannanpunkt ökar exponenten för ordningen av felet

=> bättre precision

$$S^{(2)}(h) = S(h) + \frac{S(h) - S(2h)}{15}$$

Ger $R_T = O(h^6)$ istället för $O(h^4)$

Fungerar bra för reguljära funktioner.

Lana 200508

Numerisk beräkning av derivator och lösning av ODE:er

Approximation av derivator

Vill beräkna derivata från funktionsvärden.

Tre enkla differenskvoter:

$$\left\{\begin{array}{ll} \circ \text{ Framåtdifferens} & f'(x) \approx \dfrac{f(x+h)-f(x)}{h} \\[2ex] \circ \text{ Bakåtdifferens} & f'(x) \approx \dfrac{f(x)-f(x-h)}{h} \end{array}\right.$$

medel-värde

$$\circ \text{ Centraldifferens} \quad f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$$

Om steglängden $h \to 0$ går approximationerna mot $f'(x)$. Trunkeringsfelet R_T kan analyseras:

Ex: Centraldifferens

$$R_T = \frac{f(x+h)-f(x-h)}{2h} - f'(x) \approx$$

$$\approx \frac{1}{2h}\left(\sum_{i=0}^{4}\frac{1}{i!}f^{(i)}(x)h^i - \sum_{i=0}^{4}(-1)^i\frac{1}{i!}f^{(i)}(x)h^i\right) - f'(x)$$

$\approx$ {Termer med jämn i stryks} $\approx$

$$\approx \frac{1}{2h}\left(\cancel{2f'(x)h} + 2\frac{1}{3!}f'''(x)h^3\right)\cancel{-f'(x)}$$

$$R_T = \frac{1}{3!}f'''(x)h^2 + O(h^4)$$

För centraldifferens: $R_T = O(h^2)$

För framåt-/bakåtdifferens: $R_T = O(h)$

Kan man välja h litet för att få litet fel?

Nej! $f(x+h) \approx f(x)$

=> Cancellationsfel vid cancellation

Om δ är relativa felet i funktionsberäkningarna

d.v.s. $\frac{|\delta f|}{|f|} \le \delta$ så blir felet vid framåtdiff:

$$|R_f| \le \frac{|f(x+h)+f(x)|}{|h|}\delta$$ (av triangelolikhet)

Motsvarande formler för bakåt-/central diff.

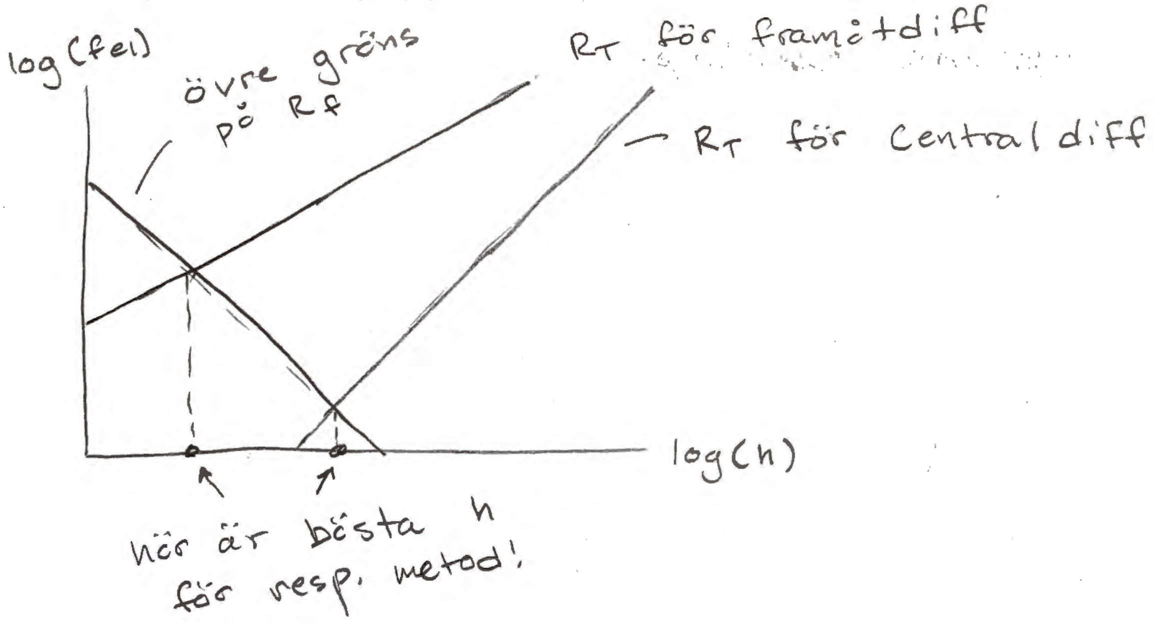
Problem

- h litet => $|R_f|$ stort
- h stort => $|R_T|$ stort

Exempel

$f(x) = \frac{1}{1+x}$. Beräkna $f'(1)$ numeriskt

Sök h som minimerar $|R_f| + |R_T|$



Man kan använda Richardsonextrapolation för att reducera $|R_T|$ utan att använda för kort steglängd h. (R_T-linje brantare)

Ex. Central differens med steglängd h

$F(h)$ har $R_T = a_1 h^2 + a_2 h^4 + \mathcal{O}(h^6)$

$F(2h)$ har $R_T = 2^2 a_1 h^2 + 2^4 a_2 h^4 + \mathcal{O}(h^6)$

$$\frac{F(h) - F(2h)}{3} =$$

$$= \frac{\cancel{f'(x)} + a_1 h^2 + a_2 h^4 - \cancel{f'(x)} - 4a_1 h^2 - 16 a_2 h^2}{3} + \mathcal{O}(h^6)$$

$$= -a_1 h^2 - \frac{15}{3} a_2 h^4 + \mathcal{O}(h^6)$$

$$\Rightarrow \quad F(h) + \frac{F(h) - F(2h)}{3} = f'(x) + \underbrace{\mathcal{O}(h^4)}_{R_T}$$

Vi har minskat R_T (men behöver fler funktionsberäkningar

Begynnelsevärdesproblem för ODE:er

Studera ett begynnelsevärdesproblem för en första ordnings differentialekvation:

$$(\text{BVP}) \begin{cases} y'(t) = f(t, y(t)), & a \leq t \leq b \\ y(a) = c & (\text{begynnelsevillkor}) \end{cases}$$

f är en given funktion, c konstant

Vi tillåter $y(t) \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $t \in \mathbb{R}$

d.v.s. system av differentialekvationer.

Generellt kan f vara icke-linjär. Om f är linjär m.a.p. y kan (BVP) skrivas.

$$\begin{cases} y'(t) = Ay(t) + b(t) & A \in \mathbb{R}^{n\times n},\ y, y', b \in \mathbb{R}^n \\ y(a) = c \end{cases}$$

Stabilitet

Om närliggande lösningar närmar sig varandra då t växer så är systemet stabilt.

Små fel i indata och beräkningsfel avtar med t om systemet är stabilt.

Om systemet är instabilt kan vi inte lita på lösningen eftersom felet ökar med t.

Högre ordnings ekvationssystem

Högre ordnings ekvationer kan skrivas om som första ordnings system.

Exempel

$$(*)\quad \begin{cases} y''' - yy'' - 2y' - 3y = 0, & 0 \le t \le 1 \\ y(0) = a,\ y'(0) = b,\ y''(0) = c \end{cases}$$

Inför $y_1 = y,\ y_2 = y',\ y_3 = y''$

Nu är $(*)$ ekvivalent med

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_1 y_3 + 2y_2 + 3y_1 \\ y_1(0) = a,\ y_2(0) = b,\ y_3(0) = c \end{cases}$$

Första ordningens system!

Differensmetoder för ODE:er

Vill beräkna approximativ lösning till (BVP)

Vi stegar oss fram från en tidpunkt till nästa.

$a = t_0 < t_1 < t_2 < \ldots < t_N = b$

Beräkna succesivt approximationer $y_k \approx y(t_k)$

Generellt OK om steglängd $t_{k+1} - t_k$ beror av k.

Vi använder dock konstant steglängd h.

$$\Rightarrow \quad t_{k+1} = t_k + h = t_0 + h(k+1)$$

Approximation med framåtdiff. ger:

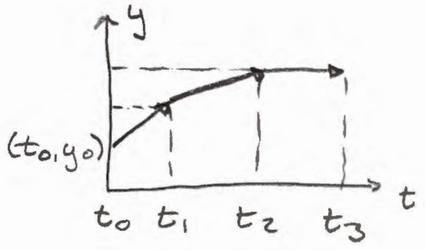
$$\frac{y(t_k + h) - y(t_k)}{h} \approx y'(t_k) = f(t, y_k(t))$$

$$\Rightarrow \boxed{y_{k+1} = y_k + hf(t_k, y_k)}$$

Eulers Framåtmetod

Geometrisk tolkning

Följ fältet $F(t,y) = \begin{bmatrix} h \\ hf(t,y) \end{bmatrix}$



F kallas riktningsfält (skalat med h)

$$\begin{bmatrix} t_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} t_k \\ y_k \end{bmatrix} + F(t_k, y_k)$$

Anm: y_{k+1} explicit i VL (smidigt)

Approximation med bakåt differens:

$$\frac{y(t_k) - y(t_k - h)}{h} \approx f(t_k, y(t_k))$$

skifta k-index

$$\Rightarrow \boxed{y_{k+1} = y_k + h f(t_k, y_{k+1})}$$

Eulers bakåtmetod

Anm. y_{k+1} ej explicit, behöver lösa ekvation. för varje iteration: kallas implicit metod.

Approximation med centraldifferens:

$$\frac{y(t_k + h) - y(t_k - h)}{2h} \approx f(t_k, y(t_k))$$

$$\Rightarrow \boxed{y_{k+1} = y_{k-1} + 2h f(t_k, y_k)}$$

Mittpunktsmetoden

Explicit metod (

Behöver två begynnelsevärden.

Integrera $y'(t) = f(t, y(t))$ från t_k till t_{k+1}

$$\Rightarrow y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t))\,dt$$

Ger flera metoder, t.ex. via trapetsregeln:

$$\approx y(t_k) + \frac{h}{2}\left(f(t_k, y(t_k)) + f(t_{k+1}, y(t_{k+1})\right)$$

$$\Rightarrow \boxed{y_{k+1} = y_k + \frac{h}{2}\left(f(t_k, y_k) + f(t_{k+1}, y_{k+1})\right)}$$

Trapetsmetoden

Implicit metod, y_{k+1} i HL & VL

Vad krävs för olika metoder?

Euler framåt, Euler bakåt & trapetsmetoden behöver bara y_k för att beräkna y_{k+1}.
Kallas enstegsmetoder

Mittpunktsmetoden kräver y_{k-1} och y_k
Avancerade approximationer av derivator ger också metoder som kräver flera värden.
Kallas flerstegsmetoder

Explicita metoder => Enkla beräkningar
Implicita metoder kräver ekv. lösn. för varje iteration men kan ha större steglängd. Om det är bättre beror på problemet. Särskilt bra för styva problem.

Lana 200511

Lösningsnoggrannhet och stabilitet för differensmetoder

Repetition

Studera BVP:

$$\begin{cases} y'(t) = f(t, y(t)), & a \le t \le b \\ y(a) = c & \text{begynnelsevillkor} \end{cases}$$

y_k approximerar exakt lösning $y(t_k)$ i $t = t_k$

$t_{k+1} = t_k + h \Leftrightarrow t_k = t_0 + kh, \quad t_0 = a$

Euler framåt

$$y_{k+1} = y_k + h f(t_k, y_k)$$

Euler bakåt

$$y_{k+1} = y_k + h f(t_{k+1}, y_{k+1})$$

Mittpunktsmetoden

$$y_{k+1} = y_{k-1} + 2h f(t_k, y_k)$$

Trapetsmetoden

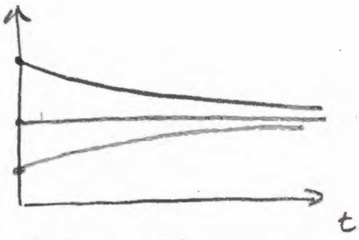
$$y_{k+1} = y_k + \frac{1}{2} h \left(f(t_k, y_k) + f(t_{k+1}, y_{k+1}) \right)$$

Problemstabilitet

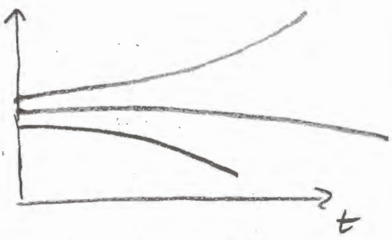
Def Problemet (BVP) är STABILT om närliggande lösningskurvor inte divergerar.
Det är ASYMPTOTISKT STABILT om lösningskurvorna asymptotiskt konvergerar.
Om $z(t)$ lösning till (BVP) med andra begynnelsevärden sådana att $|c - z(t_0)|$ är litet, så är (BVP) stabilt om

$$|y(t_{k+1}) - z(t_{k+1})| \leq |y(t_k) - z(t_k)|$$

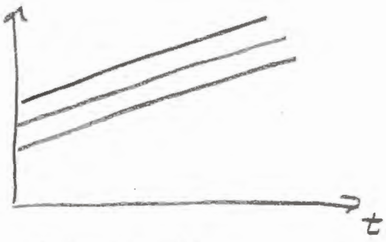
Asymptotiskt stabilt om $|y(t) - z(t)| \xrightarrow{t \to \infty} 0$



Stabilt
(asymptotiskt)



Instabilt



Stabilt
(ej asympotiskt)

Exempel $y' = \lambda y, \quad y(t) \in \mathbb{C} \quad y(0) = C_1$

Har exakt lösning $y(t) = C_1 e^{\lambda t}$

Annan lösning: $z(t) = C_2 e^{\lambda t} \qquad z(0) = C_2$

$$|y(t_{k+1}) - z(t_{k+1})| = |C_1 - C_2| \, |e^{\lambda t_{k+1}}| =$$

$$= |C_1 - C_2| \, |e^{\lambda t_k}| \, |e^{\lambda h}| = |y(t_k) - z(t_k)| \, |e^{\lambda h}|$$

$|e^{\lambda h}| = e^{Re(\lambda)h}$ (komplexdelen ger bara rotation)

Så problemet är stabilt omm $e^{Re(\lambda)h} \leq 1$
vilket är ekvivalent med $Re(\lambda)h \leq 0$

Lösningsnoggrannhet

Def: Låt $u_k(t)$ vara lösningskurvan som går genom (t_k, y_k) d.v.s exakt lösning till BVP med $a = t_k$, $c = y_k$.

Det LOKALA TRUNKERINGSFELET i t_k är

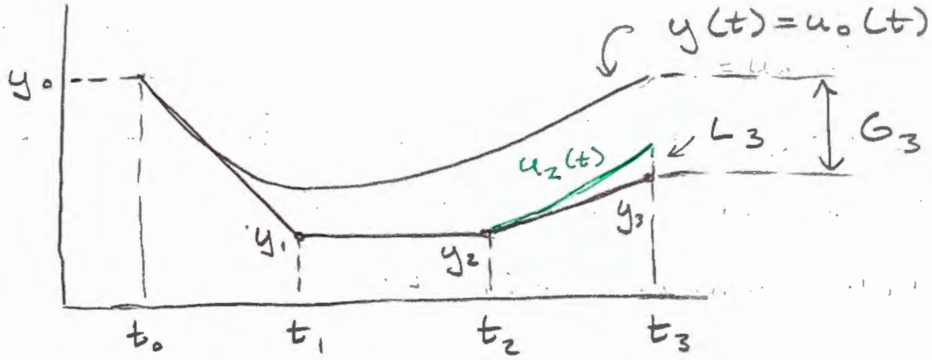
$$L_k = y_k - u_{k-1}(t_k)$$

d.v.s skillnaden mellan att använda differensmetoden eller exakt lösning från föregående steg (t_{k-1}).

Det GLOBALA TRUNKERINGSFELET i t_k är

$$G_k = y_k - u_0(t_k) = y_k - y(t_k)$$

d.v.s. skillnaden mellan approximationen och exakt lösning från startpunkten.



Def En metod som har lokalt trunkeringsfel $L_k = \mathcal{O}(h^{p+1})$ har APPROXIMATIONSORDNING p.

Om approximationsordningen är p och problemet stabilt så är $G_k = \mathcal{O}(h^p)$.

$$|G_1| = |y_1 - u_0(t_1)| = |L_1|$$

$$\begin{aligned} |G_k| &= |y_k - u_0(t_k)| \le \\ &\le |y_k - u_{k+1}(t_k)| + |u_{k+1}(t_k) - u_0(t_k)| \overset{\text{Stabilt}}{\le} \\ &\le |L_k| + |u_{k-1}(t_{k-1}) - u_0(t_{k-1})| = \\ &= |L_k| + |G_{k-1}| \end{aligned}$$

Antal delintervall $N \propto h^{-1}$. Summera N delintervall ger $G_N = h^{-1}\mathcal{O}(h^{p+1}) = \mathcal{O}(h^p)$.

Approximationsordning för differensmetoder

Studera $y'(t) = \lambda y(t) \quad y(t_0) = c$

Approximationsordning för Euler framåt

$y' = \lambda y$ ger metod

$$y_{k+1} = y_k + \lambda h y_k = (1 + h\lambda) y_k$$

$$\Rightarrow y_k = (1 + h\lambda)^k y_0 = (1 + h\lambda)^k c$$

Faktorn $(1 + h\lambda)$ kallas tillväxtfaktor.

Den exakta lösningen är $u_k(t) = y_k e^{\lambda(t - t_k)}$

Lokala trunkeringsfelet:

$$\begin{aligned} L_{k+1} &= y_{k+1} - u_k(t_{k+1}) = (1 + h\lambda) y_k - y_k e^{\lambda h} = \\ &= (1 + h\lambda - e^{\lambda h}) y_k = \{\text{Taylorutveckla } e^{\lambda h}\} \\ &= \left(\frac{(h\lambda)^2}{2} + \mathcal{O}(h^3) \right) y_k = \mathcal{O}(h^2) \end{aligned}$$

Alltså är lokalt trunkeringsfel $\mathcal{O}(h^2)$

$\Rightarrow$ approximationsordning 1.

"Halvera h => Halvera globalt fel"

Approx. ordn. för Euler bakåt

$$y' = \lambda y \Rightarrow y_{k+1} = y_k + h\lambda y_{k+1}$$

$$\Rightarrow y_{k+1} = \frac{1}{1-h\lambda} y_k$$

Tillväxtfaktor: $\frac{1}{1-h\lambda}$

som förut

Taylorutveckla båda

$$L_{k+1} = \left(\frac{1}{1-h\lambda} - e^{h\lambda}\right) y_k =$$

$$= \left(\cancel{1+h\lambda} + (h\lambda)^2 - \cancel{1-h\lambda} - \frac{1}{2}(h\lambda)^2 + \mathcal{O}(h^3)\right) y_k =$$

$$= \mathcal{O}(h^2)$$

Lokalt trunkeringsfel $\mathcal{O}(h^2)$,

approximationsordning 1.

Approx. ordn. för Trapetsmetoden

$$y' = \lambda y \Rightarrow y_{k+1} = y_k + \frac{h}{2}(\lambda y_k + \lambda y_{k+1})$$

$$\Rightarrow \left(1 - \frac{h\lambda}{2}\right) y_{k+1} = \left(1 + \frac{h\lambda}{2}\right) y_k \Rightarrow y_{k+1} = \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}} y_k$$

Tillväxtfaktor: $\frac{2+h\lambda}{2-h\lambda}$

Taylor

$$L_{k+1} = \left(\frac{2+h\lambda}{2-h\lambda} - e^{h\lambda}\right) y_k = \left(\frac{1}{12}(h\lambda)^3 + \mathcal{O}(h^4)\right) y_k$$

Lokalt trunkeringsfel $\mathcal{O}(h^3)$

approximationsordning 2.

Behöver inte lika korta steg!

Stabilitet av differensmetoder

Testa med exemplet $y' = \lambda y$, $y(t_0) = c$

Problemet bara stabilt om $Re(\lambda) \le 0$

Krövs för att fel i indata inte ska växa exponentiellt. Vill alltid ha detta.

Def: En metod är STABIL för steglängd h om den ger en icke-växande approximation för $y' = \lambda y$ med $Re(\lambda) \le 0$, d.v.s. $|\text{tillväxtfaktorn}| \le 1$.

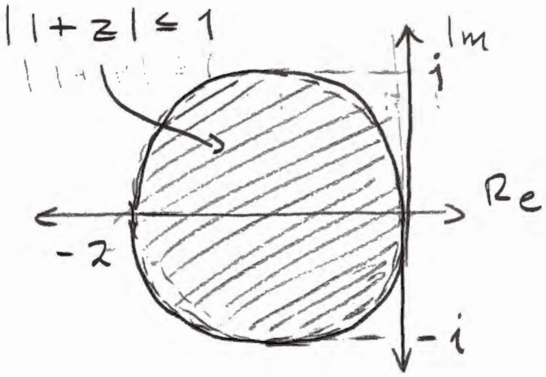
Eulers framåtmetod på $y' = \lambda y$ har tillväxtfaktor $1 + h\lambda$ så vi får stabilitet av metoden omm $|1 + h\lambda| \le 1$.

Om λ reellt: $-1 \le 1 + h\lambda \underset{\uparrow \atop \text{uppenbar}}{\le} 1 \iff h \le -\frac{2}{\lambda}$

→ Om λ är stort (negativt) måste h vara liten.

För komplexa λ:

Låt $z = h\lambda$. Vi får stabilitetsområdet



Om h väljs för stort kan vi hamna utanför stabilitetsområdet och få en instabil metod.

Detta är ett litet område jämfört med för de andra metoderna.

Lana 200512

Stabilitet av differensmetoden (forts.)

Repetition

Studera (BVP) $y'(t) = f(t, y(t))$ $a \le t \le b$.

y_k approximation vid $t_k = t_0 + kh$; $t_0 = a$

Lokalt trunkeringsfel: $L_k = \mathcal{O}(h^{p+1})$ ger approximationsordning p.

Stabilitet av metoder studeras mha $y' = \lambda y$ $y(t_0) = c$. Euler framåt har approx. ordn. 1 och stabilitetsområdet $\{z = h\lambda \in \mathbb{C} \mid |z+1| \le 1\}$

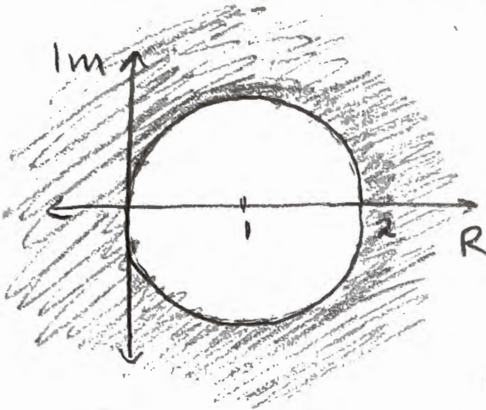
Bästa stabilitetsområdet är hela vänstra halvplanet $Re(z) \le 0$ $\Leftrightarrow$ $Re(h\lambda) \le 0$ $\Leftrightarrow$ $Re(\lambda) \le 0$. Vi har i sådana fall metodstabilitet för alla stabila problem.

Def En metod med stabilitetsområde som innehåller hela vänstra halvplanet kallas A-STABIL.

Euler framåt ej A stabil

$\Rightarrow$ Krävs korta steg för stabilitet.

Eulers bakåtmetod på $y' = \lambda y$, $y(t_0) = c$ har tillväxtfaktor $\frac{1}{1-h\lambda}$ så vårt stabilitetsområde ges av



$$\left|\frac{1}{1-z}\right| \leq 1 \iff 1 \leq |1-z|$$

Innehåller hela vänstra halvplanet!

$\Rightarrow$ Euler bakåt är A-stabil.

Trapetsmetoden på $y' = \lambda y$ har tillväxtfaktor $\frac{2+h\lambda}{2-h\lambda}$ så stabilitetsområdet ges av

$$\left|\frac{2+z}{2-z}\right| \leq 1$$

Ansätt $a+bi = z$ och kvadrera:

$$\Rightarrow 1 \geq \frac{|2+a+bi|^2}{|2-a-bi|^2} = \frac{(2+a)^2+b^2}{(2-a)^2+b^2}$$

$$\iff (2-a)^2 + \cancel{b^2} \geq (2+a)^2 + \cancel{b^2}$$

$$\iff -4a \geq 0 \iff a < 0 \iff Re(z) \leq 0$$

Så stabilitetsområdet är hela vänstra halvplanet

$\Rightarrow$ Trapetsmetoden är A stabil.

Sammanfattning

Metod	Approx. Ordn.	Stabilitets-område	A stabil
Euler framåt	1	$\|1+z\| \leq 1$	Nej.
Euler bakåt	1	$\|1-z\| \geq 1$	Ja
Trapets-metoden	2	$Re(z) \leq 0$	Ja

Euler framåt är explicit => enkla beräkningar
Ej A stabil => kort steglängd för stabilitet.
Låg approx. ordn => kort steglängd för noggrannhet.

Euler bakåt och trapetsmetoden är implicita => ekvationslösning för varje iteration. (Långsammare)
A-stabila => Lång steglängd OK för stabilitet.
Trapetsmetoden har bättre approximations-ordning men kräver två funktionsberäkningar av f.

Systemlösning med differensmetoder

Problemet $y' = \lambda y$ stabilt om $Re(\lambda) \leq 0$.

Linjära system enklare jämfört med icke-linjära.

Problemet $y' = Ay$, $y(0) = c$, A $n \times n$ diag. matris $y(t) \in \mathbb{R}^n$, $c \in \mathbb{R}^n$ har lösning

$$y(t) = \sum_{i=1}^{n} d_i e^{\lambda_i t} u_i$$

där u_i egenvektorer med egenvärde λ_i.
d_i bestäms av begynnelsevärdena.

$$\boxed{\text{Asymptotiskt stabilt d.v.s. } y(t) \xrightarrow{t \to \infty} 0 \text{ om } Re(\lambda_i) < 0 \ \forall i}$$

$Re(\lambda_i) = 0$ ger oscillerande lösninger.
Gäller även om A inte är diagonaliserbar, men fallet $Re(\lambda_i) = 0$ mer känsligt
(Vi kommer inte visa det i denna kurs)

För icke-linjära system kan vi få en uppfattning om stabiliteten genom linjärisering (d.v.s. Taylorutv. av f)

$$y' = f(y) \approx Jy + b$$

där J är jacobian för f med avseende på y.
Egenvärdena av J analyseras som ovan, d.v.s. om egenvärdena λ_i till J uppfyller $Re(\lambda_i) < 0$ har vi stabilitet.

OBS: Ger "lokal" stabilitet, välj punkter väl.

Tillbaka till linjära fallet.
Olika storleksordning på λ_i kallas <u>styva problem</u>. Ger överlagring av snabba och långsamma förlopp.

Euler framåt kräver $|1-h\lambda_i| \le 1 \quad \forall \lambda_i$
Om t.ex. $|\lambda_1|$ stort och $|\lambda_2|$ litet krävs kort steglängd p.g.a. λ_1 kan ge instabilitet men lång tid för att se effekterna av λ_2. (långsamma förlopp)

<u>Exempel</u> $y'(t) = Ay(t) \quad y(0) = \begin{pmatrix}1\\0\end{pmatrix}$

där $A = \begin{pmatrix} 0 & 1 \\ -100 & -101 \end{pmatrix}$

$\det(A - \lambda I) = \ldots = (100+\lambda)(1+\lambda) = 0$
$\Rightarrow$ Egenvärden $\lambda_1 = -100, \; \lambda_2 = -1$ (styvt!)
Euler framåt kräver
$|1 + h\lambda_i| \le 1 \quad \forall i \Rightarrow -1 \le 1 - 100h \Rightarrow \underline{\underline{h \le 2 \cdot 10^{-2}}}$
för stabilitet.
Euler bakåt alltid stabil om problemet är stabilt, d.v.s. $\mathrm{Re}(\lambda_i) < 0$

$y_{k+1} = y_k + hAy_{k+1} \Rightarrow \underline{(I - hA)y_{k+1} = y_k}$

lös för varje iteration med \ i matlab.

Prediktor - Korrektor metoder

Finns inga A stabila explicita metoder.
=> Behöver använda implicita metoder.
Om f icke-linjär kan Newton's metod eller annan iterationsmetod användas, men en bra startgissning behövs.

Använd en explicit metod för startgissningen (Predikator) som sedan används i implicita metoden (Korrektor).

Vi får samma stabilitet och noggrannhet som korrektorn.

Exempel:

Predikator = Euler framåt
Korrektor = Trapets med Newtons metod för ekvationslösn. med startgissning från prediktorn.

Runge - Kuttametoder

Explicita enstegsmetoder smidiga, men Euler framåt har dålig approxordn. 1.

Finns bättre?

Def: En metod på formen

$$y_{k+1} = y_k + h\sum_{i=1}^{s} b_i k_i$$

med $k_i = f\left(t_k + c_i h,\ y_k + h\sum_{j=1}^{s} a_{ij} k_j\right)$

som definieras av ett antal NIVÅER $s \in \mathbb{N}$, en $s \times s$ matris $A = (a_{ij})$ och två vektorer $b \in \mathbb{R}^s$ och $c \in \mathbb{R}^s$ kallas en RUNGE-KUTTAMETOD.

Exempel $s = 1$, $a_{11} = 0$, $b_1 = 0$, $c_1 = 0$ ger

$$y_{k+1} = y_k + hk_1 \quad \text{med} \quad k_1 = f(t_k, y_k)$$

Alltså är Eulers metod en RK-metod

Exempel [Heuns metod]

$s = 2$, $A = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, $b = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$, $c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ger

$$y_{k+1} = y_k + \frac{h}{2}(k_1 + k_2) \quad \text{med}$$

$$k_1 = f(t_k, y_k)$$

$$k_2 = f(t_k + h,\ y_k + hk_1)$$

Test med $y' = \lambda y$ ger $k_1 = \lambda y_k$, $k_2 = \lambda y_k + h\lambda^2 y_k$

$$\Rightarrow y_{k+1} = \dots = \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2\right) y_k$$

$$L_k = \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2 - e^{h\lambda}\right) y_k = \mathcal{O}(h^3)$$

$\Rightarrow$ approx. ordn. 2. Bättre!

Exempel [Klassisk Runge-Kutta]

$$s = 4, \quad A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, b = \frac{1}{6}\begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \end{pmatrix}, c = \begin{pmatrix} 0 \\ 1/2 \\ 1/2 \\ 1 \end{pmatrix}$$

ger $y_{k+1} = \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ med

$$k_1 = f(t_k, y_k)$$
$$k_2 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_1)$$
$$k_3 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_2)$$
$$k_3 = f(t_k + h, y_k + hk_3)$$

bara funktions-beräkningar!

Inga ekvationer behöver lösas (eftersom A valts smart)

Har approximationsordning 4, kräver 4 beräkningar av f. Euler framåt kräver bara 1.

Matlabfunktionen ode45 använder den här Runge-Kutta metoden och en med 5 nivåer för feluppskattning.

Har tillväxtfaktor $1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!}$

=> stabilitetsområde $\left|1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!}\right| \leq 1$

För styva problem fungerar ode15s bättre.

Lana 200514

Numerisk linjär algebra

Tre huvudproblem:

- Linjära ekvationssystem
- Linjära minstakvadratproblem
- Egenvärdesproblem

Linjära ekvationssystem

Vill lösa $A\mathbf{x} = \mathbf{b}$

Två numeriska aspekter:

- Antal operationer (flyttalsaddition º multiplikation
- Numerisk stabilitet

Om man vill lösa $A\mathbf{x} = \mathbf{b}$ med många olika högerled används LU-faktorisering för att minska antal operationer.

LU-faktorisering

$A = LU$ där A $m \times n$-matris, L nedåt-triangulär $m \times m$-matris med ettor på diagonalen (Lower) och U $m \times n$-matris med trappstegsform (som vi får av Gausselimination) (Upper)

Exempel

$$\underset{4\times5}{A} = \underset{\substack{L\\4\times4}}{\begin{pmatrix} 1 & 0 & 0 & 0 \\ * & 1 & 0 & 0 \\ * & * & 1 & 0 \\ * & * & \times & 1 \end{pmatrix}} \underset{\substack{U\\4\times5}}{\begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}$$

Om $A = LU$ kan vi lösa $Ax = b$ effektivt

Låt $y = Ux$ så $b = Ax = LUx$

1) Lös $Ly = b$ med framåtsubstitution

2) Lös $Ux = y$ med bakåtsubstitution

I steg 2 är det möjligt att få parameterlösning

För varje nytt högerled ca $2n^2$ operationer om A $n \times n$-matris. (n^2 från fram/bak-subs.)

Gausselimination kräver ca $\frac{2}{3}n^3$ operationer

LU faktorisering billigare om man har många högerled b ty man behöver bara göra LU faktorisering (Gausselimination) en gång.

<u>Beräkning av LU-faktorisering</u>

Som vanlig Gausselemination men sparar information om vad man gjort. Skillnad:

- Inga omskalningar av rader (behövs ej) pivotelement i U behöver inte vara 1.
- Antag att radbyten inte behövs (återkommer till detta)

Radreducera A till trappstegsmatris U.

Kan skrivas $E_p \cdots E_1 A = U$ där E_k motsvarar radoperation, d.v.s. samma op. på I.

Alla E_k nedåt triangulära med 1:or på diagonalen ty använder multiplar av rader ovanför. Ger $E_p \cdots E_1$ nedåt triangulär med 1:or på diagonalen.

Radoperationer inverterbara

$\Rightarrow$ E_k inverterbara

$\Rightarrow$ $E_p \cdots E_1$ inverterbar

$$\boxed{A = \underbrace{(E_p \cdots E_1)^{-1}}_{L}\underbrace{(E_p \cdots E_1) A}_{U} = LU}$$

Bilda L stegvis så att $E_p \cdots E_1 L = I$

Exempel LU-faktorisera A

$$\underbrace{\begin{pmatrix} 2 & 4 & -1 \\ -4 & -5 & 3 \\ 2 & 10 & 4 \end{pmatrix}}_{A} \begin{matrix} \\ \overset{2}{\leftarrow} \\ \overset{-1}{\leftarrow} \end{matrix} \sim \underbrace{\begin{pmatrix} 2 & 4 & -1 \\ 0 & 3 & 1 \\ 0 & 6 & 5 \end{pmatrix}}_{E_1 A} \begin{matrix} \\ \\ \overset{-2}{\leftarrow} \end{matrix} \sim \underbrace{\begin{pmatrix} 2 & 4 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix}}_{E_2 E_1 A = U}$$

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

Om $L = \begin{pmatrix} | & | & | \\ v_1 & v_2 & v_3 \\ | & | & | \end{pmatrix}$ vill vi ha $E_2 E_1 L = I$

Men E_2 påverkar inte första kolonnen

$$\Rightarrow E_1 v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow v_1 = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

det vill säga byt tecken på radoperationerna.

$$\text{P.s.s.} \quad E_2 v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow v_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Alltså $A = LU$ med $L = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$, $U = \begin{pmatrix} 2 & 4 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix}$

Kolonnerna i L ges av omvänt tecken på radoperationerna.

Pivotering

Radbyten kan behövas, kallas här *pivotering*.

För numerisk stabilitet byter vi rader så att det till beloppet största värdet blir pivotelement innan vi gör radoperationer.

Gör att alla element i L blir $|\cdot| \leq 1$

Radbytena håller man reda på m.h.a. en permutationsmatris P, som beskriver alla radbyten i alla eleminationssteg.

Man får då: $\boxed{PA = LU}$

I MATLAB: `[L,U,P] = lu(A)`

Så lösning av $Ax=b$ med många högerled:

1) Faktorisera $PA = LU$
   $\sim \frac{2}{3}n^3$ operationer om A $n \times n$
2) Framåtsubstitution $Ly = Pb$
   $\sim n^2$ operationer
3) Bakåtsubstitution $Ux = y$
   $\sim n^2$ operationer

Steg 2) & 3) utförs för varje högerled.

Fungerar för att:

$$Ax = b \iff PAx = Pb \iff LUx = Pb$$

$$\iff \begin{cases} Ly = Pb \\ Ux = y \end{cases}$$

Varför pivotering?

Exempel utan pivotering

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 2 & 4 \end{array}\right] \begin{array}{l} \urcorner \, -1 \\ -10^{20} \\ \leftarrow \end{array} \sim \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & 2-10^{20} & 4-10^{20} \end{array}\right]$$

Flyttalsberäkningar ger utskiftning

Tappar info om 2:a raden

$$\approx \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & -10^{-20} & -10^{20} \end{array}\right] \Rightarrow x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Sätt in i ursprunglig ekvation => stort fel

Exempel med pivotering

Kasta om raderna

$$\left[\begin{array}{cc|c} 1 & 2 & 4 \\ 10^{-20} & 1 & 1 \end{array}\right] \begin{array}{l} \urcorner \\ -10^{-20} \\ \leftarrow \end{array} \sim \left[\begin{array}{cc|c} 1 & 2 & 4 \\ 0 & 1-10^{-20} & 1-4\cdot 10^{-20} \end{array}\right] \approx$$

$$\approx \left[\begin{array}{cc|c} 1 & 2 & 4 \\ 0 & 1 & 1 \end{array}\right] \Rightarrow x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Ger exakt lösning till stört problem

$(A+E)x = b$ med $E = \begin{pmatrix} 0 & 0 \\ -10^{-20} & 0 \end{pmatrix}$

och $A = \begin{pmatrix} 1 & 2 \\ 10^{-20} & 1 \end{pmatrix}$

Generellt är elementen i E lika stora som avrundningselementen. (maskintalet)

Gausselimination (eller LU-faktorisering) med pivotering är en stabil algoritm

Lösningsnoggrannhet och konditionstal

Osäkerhet i A och b från mätningar, avrundning, algoritmen ger approximationer $\hat{A}$ och $\hat{b}$. Vi löser alltså $\hat{A}\hat{x} = \hat{b}$ istället för $Ax = b$. Hur stort blir felet i $\hat{x}$?

Låt $\delta A = \hat{A} - A$, $\delta b = \hat{b} - b$, $\delta x = \hat{x} - x$

Felfortplantning $\delta b \to \delta x$

Antag $\delta A = 0$ och A^{-1} existerar

$$\begin{cases} A(x + \delta x) = b + \delta b & (\mathrm{I}) \\ Ax = b & (\mathrm{II}) \end{cases}$$

$(\mathrm{I}) - (\mathrm{II})$ ger $A\delta x = \delta b$

$$\Rightarrow \delta x = A^{-1}\delta b \Rightarrow \|\delta x\| = \|A^{-1}\delta b\| \le \|A^{-1}\|\,\|\delta b\|$$

Vi har även $\|b\| = \|Ax\| \le \|A\|\,\|x\| \Rightarrow \frac{1}{\|x\|} \le \frac{\|A\|}{\|b\|}$

Tillsammans får vi:

$$\frac{\|\delta x\|}{\|x\|} \le \|A^{-1}\|\,\|A\| \frac{\|\delta b\|}{\|b\|} = \overset{\overset{\text{rcond(A) i ML}}{\downarrow}}{\kappa(A)} \frac{\|\delta b\|}{\|b\|}$$

där $\kappa(A) := \|A\|\,\|A^{-1}\|$, kallas konditionstalet

$$\boxed{\text{Relativa felet i lösningen} \le \text{kondtal} \cdot \text{Relativa felet i högerledet}}$$

Problemet är välkonditionerat om $\kappa(A)$ är "litet" (alltid ≥ 1). Hur litet beror på situationen.

Problemet är illakonditionerat om $\kappa(A)$ är "stort"

Felfortplantning - $\delta A \to \delta x$

Antag $\delta b = 0$

$$\begin{cases} (A+\delta A)(x+\delta x) = b & (\mathrm{I}) \\ Ax = b & (\mathrm{II}) \end{cases}$$

$$(\mathrm{I}) - (\mathrm{II}) \Rightarrow A\delta x + \delta A(x+\delta x) = 0$$

$$\Leftrightarrow \delta x = -A^{-1}\delta A(x+\delta x)$$

$$\Rightarrow \|\delta x\| = \|A^{-1}\delta A(x+\delta x)\| \leq \|A^{-1}\|\|\delta A\|\|x+\delta x\|$$

$$\Rightarrow \frac{\|\delta x\|}{\|x\|} \approx \frac{\|\delta x\|}{\|x+\delta x\|} \leq \|A^{-1}\|\|A\|\frac{\|\delta A\|}{\|A\|}$$

$$\Rightarrow \boxed{\frac{\|\delta x\|}{\|x\|} \lessapprox \kappa(A)\frac{\|\delta A\|}{\|A\|}}$$

Nästan som $\delta b - \delta x$ fallet.

Lana 200515

Minstakvadratproblem

Om $A \in \mathbb{R}^{m \times n}$ med $m > n$ är $Ax = b$ i regel ej lösbart. Minimera 2-normen av residualen $r = Ax - b$. Kan lösas med normalekvationerna $A^T A x = A^T b$, men $\kappa(A^T A) = (\kappa(A))^2$ kan bli mycket stort, så normalekvationerna blir numeriskt instabila. Alternativa metoder:

- QR-faktorisering
- Singulärvärdesfaktorisering (SVD)

QR-faktorisering

Låt A vara $m \times n$-matris med rang $A = n$, $m \geq n$. Sök faktorisering $A = Q\hat{R}$ med Q ortogonal $m \times m$-matris $(Q^T = Q^{-1})$ och $\hat{R} = \begin{pmatrix} R \\ 0 \end{pmatrix}$ $m \times n$-matris där R uppåt triangulär matris. Kallas full QR-faktorisering. I Matlab qr(A).

Dela upp $Q = (Q_1\ Q_2)$ där Q har n kolonner och Q_2 har $m-n$ kolonner, ger

$$A = (Q_1\ \ Q_2)\begin{pmatrix} R \\ 0 \end{pmatrix} = \overset{m \times n}{Q_1}\overset{n \times n}{R}$$

$A = Q_1 R$ kallas kompakt QR-faktorisering. I Matlab qr(A,0).

rang $A = n$ ger R inverterbar.

Kan lösa MK-problemet, d.v.s. minimera $\|r\|_2$ där $r = Ax - b$.

$$\|r\|^2 = \|Ax-b\|^2 = \|Q\hat{R}x - b\|^2 = \leftarrow Q^T = Q^{-1}$$

$$= \|Q(\hat{R}x - Q^Tb)\|^2 = \longleftarrow \text{ ortogonal matris bevarar längd}$$

$$= \|\hat{R}x - Q^Tb\|^2 = \left\| \begin{pmatrix} R \\ 0 \end{pmatrix} x - \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} b \right\|^2 = \leftarrow \text{2-norm}^2 = \text{kvadrat av komponenter}$$

$$= \underbrace{\|Rx - Q_1^Tb\|^2}_{\substack{=0 \text{ om} \\ Rx = Q_1^Tb}} + \underbrace{\|Q_2^Tb\|^2}_{\text{beror ej på } x, \text{ alltid } \geq 0}$$

Alltså: Lösning av det uppåt triangulära systemet $\boxed{Rx = Q_1^Tb}$ ger minstakvadratlösningen. R^{-1} existerar om rang $A = n$

Konditionstalet $\kappa(R) = \kappa(A)$ är betydligt bättre än för normalekvationerna.

Minsta residualens storlek ges av $\|Q_2^Tb\|$

För att beräkna minstakvadratlösningen behövs bara Q_1 och R, d.v.s. kompakt QR-faktorisering

Hur beräknar man en QR-faktorisering?

Kan ses som basbyte för $V(A)$ till ON-bas.

Alternativ 1: Gram-Schmidt

Om $A = (a \;\cdots\; a_n)$ $\quad a_i \in \mathbb{R}^m$

Vill få ON-bas $\{q_1, \ldots, q_n\}$ för $V(A) = \text{span}\{a_1, \ldots, a_n\}$. Använd Gram-Schmidt!

$$u_1 = a_1, \quad q_1 = \frac{u_1}{\|u_1\|}$$

$$u_2 = a_2 - q_1 q_1^T a_2, \quad q_2 = \frac{u_2}{\|u_2\|}$$

$$u_3 = a_3 - q_1 q_1^T a_3 - q_2 q_2^T a_3, \quad q_3 = \frac{u_3}{\|u_3\|}$$

$$\cdots$$

Återskapa:

$$\left.\begin{aligned} a_1 &= \|u_1\| q_1 \\ a_2 &= \|u_2\| q_2 + q_1 q_1^T a_2 \\ a_3 &= \|u_3\| q_3 + q_1 q_1^T a_3 + q_2 q_2^T a_3 \\ &\cdots \end{aligned}\right\} \text{linjär komb av } q_1, \ldots, q_n$$

Låt $r_{jj} = \|u_j\|$ och $r_{ij} = q_i^T a_j$

Då får vi:

$$A = \underbrace{(q_1 \;\cdots\; q_n)}_{Q_1} \underbrace{\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ 0 & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}}_{R}$$

ger kompakt QR-faktorisering.

I praktiken: gör GS-metod och spara vissa värden

Addera m-n extra vektorer så att Gram-Schmidt ger ON-bas för hela $\mathbb{R}^m$. Vi får då full QR-faktorisering

$$A = (q_1 \cdots q_n \cdots q_m) \begin{pmatrix} R \\ 0 \end{pmatrix}$$

En bas för $N(A^T)$ ger m-n lin. ober. vektorer som är ortagonala mot $V(A) = \text{span}\{q_1, \dots, q_n\}$ d.v.s. ortogonala mot $q_1, \dots, q_n$. Bra val för att generera $q_{n+1} \cdots q_m$ med.

Householderspeglingar

Tyvärr ger Gram-Schmidt numeriska problem vid QR-faktorisering om kolonnerna i A är nästan parallella. Modern programvara använder istället Householderspeglingar. Konstruerar en sekvens av matriser H_i så att

$$\underbrace{H_n \cdots H_1}_{Q^T} A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad Q \text{ ortagonal}$$

H_i ger önskad form på kolonn i

Def En matris på formen $H = I - 2vv^T$ $v \in \mathbb{R}^n$ ($\Rightarrow$ $H \in \mathbb{R}^{n \times n}$), där $\|v\| = 1$ kallas HOUSEHOLDERMATRIS.

Egenskaper för $H = I - 2vv^T \quad \|v\| = 1$:

- H symmetrisk ty

$$H^T = (I - 2vv^T)^T = I - 2vv^T = H$$

- H ortogonal ty

$$H^TH = H^2 = (I - 2vv^T)(I - 2vv^T) =$$

$$= I - 4vv^T + 4v\underbrace{\cancel{v^Tv}}_{v^Tv = I}v^T = I$$

- Motsvarar spegling ty

$$Hv = v - 2vv^Tv = v - 2v = -v \leftarrow \text{egenvärde } -1$$

Om $y \perp v$ $\quad Hy = y - 2v\overset{0}{\cancel{v^Ty}} = y \leftarrow$ egenvärde 1

Så H ger spegling i planet ortogonalt mot v

Givet två vektorer $y, z \in \mathbb{R}^n$ vill man välja v så att $Hy = z$.

$$z = Hy = (I - 2vv^T)y = y - 2(v^Ty)v = y - \overset{\text{skalär}}{\overset{\downarrow}{\alpha}} v$$

$\Rightarrow \quad v = \dfrac{y - z}{\alpha} \qquad \|v\| = 1$ ger $\alpha = \|y - z\|$

Alltså $\boxed{v = \dfrac{y - z}{\|y - z\|} \quad \text{ger} \quad Hy = z}$

För QR-faktoriseringen

Om $A = (a_1 \dots a_n)$, välj H_1 så att $H_1 a$

$$H_1 a_1 = \begin{pmatrix} r_{11} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = r_{11} e_1 \quad (\text{första kolonnen i } \hat{R})$$

$(r_{11} > 0)$

Då blir $H_1 A = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & \hat{a}_2 & & \hat{a}_n \\ \vdots & \vdots & & \vdots \\ 0 & & & \end{pmatrix}$

$$\|a_1\| = \|\overset{\text{H ortogonal}}{\overset{\downarrow}{H_1}} a_1\| = \|r_{11} e_1\| = r_{11}$$

⟶

→

d.v.s. välj $r_{11} = \|a_1\|$ och $v_1 = \frac{a_1 - r_{11} e_1}{\|a_1 - r_{11} e_1\|}$

$H_1 = I - 2 v_1 v_1^T$. fixar första kolonnen i $\hat{R}$

Låt $\hat{a}_2 = \begin{pmatrix} 0 \\ \tilde{\tilde{a}}_2 \\ \vdots \end{pmatrix}$ d.v.s. (kolonn 2 i $H_1 A$) $- \begin{pmatrix} r_{12} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

Välj $r_{22} = \|\hat{a}_2\|$ och $v_2 = \frac{\hat{a}_2 - r_{22} e_2}{\|\hat{a}_2 - r_{22} e_2\|}$

ger H_2 så att $H_2 \hat{a}_2 = r_{22} e_2$.

H_2 påverkar inte första raden eftersom komponent 1 i v_2 är 0.

$$H_2 H_1 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ & r_{22} & r_{23} & \cdots & r_{2n} \\ & \bigcirc & \tilde{\tilde{a}}_3 & \cdots & \tilde{\tilde{a}}_n \\ & & \vdots & & \vdots \end{pmatrix} \leftarrow$$

Två första rader OK!

Fortsätt med $\hat{a}_3 = \begin{pmatrix} 0 \\ 0 \\ \tilde{\tilde{a}}_3 \\ \vdots \end{pmatrix}$ p.s.s. $\Rightarrow H_3 \hat{a}_3 = r_{33} a_3$

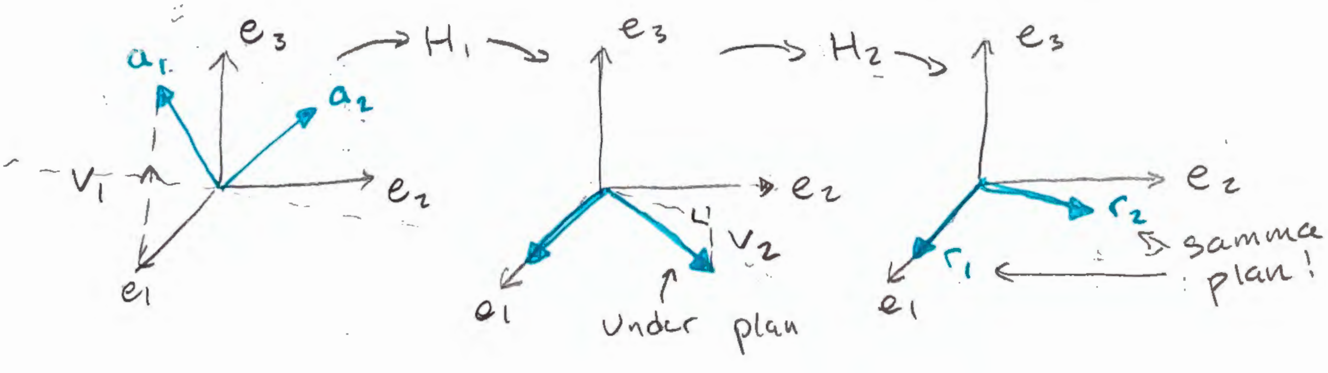
till slut får vi $H_n \cdots H_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix} \begin{matrix} \}\, n \\ \}\, m-n \end{matrix}$

Kalla $Q^T = H_n \cdots H_1$

$\Rightarrow Q^T A = \hat{R} \Leftrightarrow A = Q\hat{R}$ ty Q ortogonal.

Hela processen är multiplikation med ortogonala matriser H_i. $\kappa(H_i) = 1$ (gäller alla ortogonala matriser).

Bästa möjliga numeriska egenskaper.



Lana 200518

Singulärvärdesfaktorisering (SVD)

Liknar diagonalisering fast för $m \times n$-matriser

Låt A vara godt. reell $m \times n$-matris. Sök faktorisering

$$\boxed{A = U \Sigma V^T}$$

- U ortogonal $m \times m$
- V ortogonal $n \times n$
- Σ "diagonal" $m \times n$

Antag $m \geq n$. Då är Σ på formen

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \quad \text{där } \sigma_i \geq 0 \ \forall i$$

σ_i kallas singulära värden. Omordning av kolonnerna i U och V och elementen i Σ ger $\boxed{\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n}$. Antar detta generellt.

$A = U\Sigma V^T$ kallas <u>full SVD faktorisering</u>.

Kan alltid göras.

Exempel Om A är symmetrisk och positivt semidefinit är diagonaliseringen $A = TDT^T$ ett specialfall $U = V = T$, $\Sigma = D$. Om negativa egenvärden gör teckenbyte i Σ och V (eller U)

Om rang $A = r$ blir $\sigma_i = 0$ för $i = r+1, \dots, n$

$$\Rightarrow \quad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{matrix} \}r \\ \}m-r \end{matrix} \qquad \Sigma_1 = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix}$$

$$\underbrace{}_{r} \quad \underbrace{}_{n-r}$$

Dela upp $U = (\underbrace{U_1}_{r} \quad \underbrace{U_2}_{m-r})$ $\quad V = (\underbrace{V_1}_{r} \quad \underbrace{V_2}_{n-r})$

$$A = U\Sigma V^T = (U_1 \quad U_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} =$$

$$= \underline{\underline{U_1 \Sigma_1 V_1^T}}$$

Kallas <u>kompakt SVD faktorisering</u>.

I Matlab: svd(A) för full, svd(A,0) för kompakt SVD.

Minstakvadratproblem med SVD

Minimera $\|r\|$ där $r = Ax - b$,

$U^{-1} = U^T$

$$\|r\|^2 = \|Ax - b\|^2 = \|U\Sigma V^T x - b\|^2 =$$

$$= \|U(\Sigma V^T x - U^T b)\|^2 = \leftarrow \text{ U ortogonal}$$

$$= \|\Sigma V^T x - U^T b\|^2 =$$

$$= \left\| \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} x - \begin{pmatrix} U_1^T \\ U_2^T \end{pmatrix} b \right\|^2 =$$

$$= \left\| \begin{pmatrix} \Sigma_1 V_1^T \\ 0 \end{pmatrix} x - \begin{pmatrix} U_1^T \\ U_2^T \end{pmatrix} b \right\|^2 = \leftarrow \text{ pythagoras}$$

$$= \underbrace{\|\Sigma_1 V_1^T x - U_1^T b\|^2}_{= 0 \text{ om } \Sigma_1 V_1^T x = U_1^T b} + \underbrace{\|U_2^T b\|^2}_{\text{Oberoende av } x}$$

Så x minimerar $\|r\|$ om $\Sigma_1 V_1^T x = U_1^T b$

$$\Leftrightarrow \boxed{V_1^T x = \Sigma_1^{-1} U_1^T b \quad (*)}$$

OBS: Σ_1 alltid inverterbar men V_1^T ej inverterbar om rang $A = r < n$. Får icke entydiga lösningar.

$$I = \begin{array}{cc} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \\ \underbrace{\quad}_{r} \underbrace{\quad}_{n-r} \end{array} = V^T V = \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} (V_1 \ V_2) = \begin{array}{c} \begin{pmatrix} V_1^T V_1 & V_1^T V_2 \\ V_2^T V_1 & V_2^T V_2 \end{pmatrix} \\ \underbrace{\qquad}_{r} \ \underbrace{\qquad}_{n-r} \end{array}$$

$\nwarrow$ V ortogonal

$$\Rightarrow V_1^T V_1 = V_2^T V_2 = I$$

Men detta betyder inte att V_1 är ortogonal. Behöver inte vara kvadratisk. Dock löser

$$\boxed{x = V_1 \Sigma_1^{-1} U_1^T b}$$

alltid $(*)$. Entydigt om rang $A = n$

Om rang $A = r < n$ är lösningen till $V_1^T x = \Sigma_1^{-1} U_1^T b$ ej entydig.

$$\boxed{x = V_1 \Sigma_1^{-1} U_1^T b}$$

ger dock alltid den lösning med minst $\|x\|$.

Residualens norm är $\|r\| = \|U_2^T b\|$

Kompakt SVD räcker för att beräkna x.

<u>Def</u> Matrisen $A^{+} = V_1 \Sigma_1^{-1} U_1^T$ kallas MOORE-PENROSE PSEUDOINVERS

"Närmaste invers man kan få för icke-kvadratiska matriser"

Om A^{-1} existerar är $A^{+} = A^{-1}$.

$x = A^{+}b$ är minstakvadratlösning till $Ax = b$.

I Matlab ges A^{+} av pinv(A).

Konditionstal för icke-inverterbara matriser kan definieras $\kappa(A) = \|A\| \|A^{+}\|$

Egenvärden och singulära värden

Om $A = U\Sigma V^T$

$$\Rightarrow A^TA = V\Sigma^T\underbrace{U^TU}_{I}\Sigma V^T = V\Sigma^T\Sigma V^T =$$

$$\Sigma^T\Sigma = \begin{pmatrix} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_n & \end{pmatrix}\begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ & 0 & \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{pmatrix}$$

V ortogonal

$\Rightarrow$ A^TA similär med $\Sigma^T\Sigma$

$\Rightarrow$ A^TA och $\Sigma^T\Sigma$ har samma egenvärden.

A^TA är positivt semidefinit så $\lambda_i \geq 0$

(ty $x^TA^TAx = (Ax)^T(Ax) = \|Ax\|^2 \geq 0$)

$\Rightarrow$ de singulära värdena ges av $\sigma_i = \sqrt{\lambda_i}$

där λ_i är egenvärdena till A^TA.

(Ger ett sätt att konstruera Σ)

Hur konstruerar vi U och V?

Om $V = (v_1 \ldots v_n)$ och $U = (u_1 \cdots u_m)$ så

$$A^TA = V\Sigma^T\Sigma U^T = \sum_{i=1}^{n} v_i\sigma_i^2 v_i^T$$

$$\Rightarrow A^TAv_j = \sum_{i=1}^{n} v_i\sigma_i^2 v_i^T v_j \overset{V \text{ ortogonal}}{=} \sigma_j^2 v_j$$

Så v_j egenvektor med egenvärde σ_j^2 till A^TA.

(ger ett sätt att bilda V)

P.s.s. $AA^T = \sum_{i=1}^{n} u_i\sigma_i^2 u_i^T \Rightarrow AA^Tu_j = \sigma_j^2 u_j$

Så u_j egenvektor till AA^T för $1 \leq j \leq n$

Vi kan välja resterande u_j så U ortogonal.

⟶

$\longrightarrow$

OBS: A^TA och AA^T för samma nollskilda egenvärden men AA^T har $m-n$ extra egenvärden som är 0. (om $m > n$)

Trunkerad SVD

$$A = U_1 \Sigma_1 V_1^T \text{ (kompakt SVD) kan skrivas}$$

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad \text{där } r = \text{rang}(A)$$

Innehåller all information om A. Ibland vill vi approximera A med mindre information. (till exempel bildkompression eller signalbehandling)

Eftersom σ_i är avtagande (och positiva) är de sista σ_i mindre betydelsefulla.
För $k < r$ kan vi bilda trunkerade SVD-approximationen:

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \qquad k = \text{rang}(A_k) < r = \text{rang}(A)$$

detta är bästa möjliga rank k approx. av A.

$$A - A_k = U_1 \begin{pmatrix} 0_{\ddots} & & 0 \\ & \sigma_{k+1} & \\ 0 & & \ddots \, \sigma_r \end{pmatrix} V_1^T$$

Normen av B är $\|B\| = \sqrt{\lambda_{max}}$ där λ_{max} är största egenvärdet till B^TB, d.v.s.
$\|B\| = \sigma_{max}$.

$$\Rightarrow \|A - A_k\| = \sigma_{k+1}$$

Så σ_{k+1} litet $\Rightarrow$ bra approx.

Konditionstal och singulära värden

$\kappa(A) = \|A\| \, \|A^+\|$

$\|A\| = \sigma_{max}$ där σ_{max} största σ_i för A.

$\|A^+\| = \sqrt{\lambda_{max}}$ där λ_{max} största egenvärde till $(A^+)^T(A^+)$.

$$(A^+)^T(A^+) = (V_1\Sigma_1^{-1}U_1^T)^T V_1\Sigma_1^{-1}U_1^T =$$

$$= U_1\Sigma_1^{-1}\cancel{V_1^T V_1}\Sigma_1^{-1}U_1^T =$$

$$= U_1 \begin{pmatrix} \sigma_1^{-2} & & 0 \\ & \ddots & \\ 0 & & \sigma_r^{-2} \end{pmatrix} U_1^T$$

$\Rightarrow \lambda_{max} = \sigma_r^{-2} \Rightarrow \|A^+\| = \sigma_{min}^{-1}$

Slutsats: $\boxed{\kappa(A) = \frac{\sigma_1}{\sigma_r} = \frac{\sigma_{max}}{\sigma_{min}}}$

Trunkering ger alltså bättre konditionering.

Om $r<n$ kan små störningar av A göra att $\sigma_{r+1} = 0$ blir $\sigma_{r+1} > 0$ (litet)

$\Rightarrow \kappa(A)$ stort (delar med litet men nollskilt)

Trunkera små singulära värden

$\Rightarrow$ bättre stabilitet.

Minstakvadratlösningen

$$x = V_1\Sigma_1^{-1}U_1 b = \sum_{i=1}^{r} (u_i^T b)\,\sigma_i^{-1} v_i$$

kan trunkeras

$$x = \sum_{i=1}^{k} (u_i^T b)\,\sigma_i^{-1} v_i = A_k^+ b$$

Blir mindre exakt men också mindre känsligt för störningar.

Lana 200519

Numerisk beräkning av egenvärden och egenvektorer

Vill numeriskt beräkna egenpar (λ, v) d.v.s. egenvektor v med egenvärde λ.

Karaktäristiska ekvationen inte bra för större system.

- För A $n \times n$-matris får determinanten $n!$ termer.
- Numeriskt instabilt att hitta rötter till polynom av hög grad.

Alternativa metoder

- Iterativa metoder (stegvis förbättrar approx)
  - Potensmetoden ger till beloppet största λ
  - Inversiteration ger till beloppet minsta λ
  - Inversiteration med skift ger egenvärde närmast ett givet värde
- Transformationsmetoder

  $B = T^{-1} A T$ har samma egenvärden som A.

  Bra om B har t.ex. triangulär form (egenvärden kan läsas av enkelt)

Potensmetoden (Power iteration)

Exempel

$A = \begin{pmatrix} 1 & 3 \\ 2 & 2 \end{pmatrix}$. Låt $x_{k+1} = Ax_k$, $x_0 = \begin{pmatrix} -5 \\ 5 \end{pmatrix}$

$\Rightarrow x_1 = \begin{pmatrix} 10 \\ 0 \end{pmatrix}$ $x_2 = \begin{pmatrix} 10 \\ 20 \end{pmatrix}$ $x_3 = \begin{pmatrix} 70 \\ 60 \end{pmatrix}$ $x_4 = \begin{pmatrix} 250 \\ 260 \end{pmatrix}$

x_n verkar gå mot $\alpha \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Är $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ egenvektor?

Facit: $\lambda_1 = 4$ $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\lambda_2 = -1$ $v_2 = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$

Idé:

$$x_0 = \begin{pmatrix} -5 \\ 5 \end{pmatrix} = 1 v_1 + 2 v_2$$

$$\Rightarrow x_n = \underbrace{1 (4)^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}}_{\text{Dominerar!}} - 2(-1)^n \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

Så $x_n = A^n x_0 \longrightarrow \alpha v_1$ om λ_1 största egenvärde.

Generellt:

Antag A diagonaliserbar $n \times n$-matris med egenvärden $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$.

(λ_1, v_1) kallas <u>dominerande egenpar</u>

Låt $x_{k+1} = Ax_k$, x_0 givet. $\Rightarrow x_k = A^k x_0$

$\{v_1, \dots, v_n\}$ bas för $\mathbb{R}^n$ (A diagonaliserbar)

$\Rightarrow x_0 = \sum_{j=1}^{n} c_j v_j$ för $c_j \in \mathbb{R}$. Antag $c_1 \neq 0$

$$\Rightarrow x_k = A^k x_0 = \sum_{j=1}^{n} c_j A^k v_j = \sum_{j=1}^{n} c_j \lambda_j^k v_j =$$

$$= \lambda_1^k \left(c_1 v_1 + \sum_{j=2}^{n} c_j \left(\frac{\lambda_j}{\lambda_1} \right)^k v_j \right) \longrightarrow c_1 \lambda_1^k v_1$$

ty $\left(\frac{\lambda_j}{\lambda_1} \right) < 1$ $\forall j = 2, \dots, n$

→

x_k får riktning v_1 då $k \to \infty$ men λ_1^k problematisk (blir för stor eller liten) Normering löser problemet.

Potensmetoden

Välj startvektor x_0

Iterera:

$$y_{k+1} = Ax_k$$

$$x_{k+1} = \|y_{k+1}\|^{-1} y_{k+1}$$

Avbryt när $\|x_{k+1} - x_k\| <$ önskad tolerans

OBS!

Om $c_1 \neq 0$ konvergerar x_k mot egenvektorn $u_1 = \frac{v_1}{\|v_1\|}$. (upp till tecken), annars ej. Om $|\lambda_1| = |\lambda_2|$ konvergerar metoden inte.

Beräkning av motsvarande egenvärde

$$Ax = \lambda x \quad \Rightarrow \quad x^T A x = \lambda x^T x = \lambda \|x\|^2$$

$$\Rightarrow \quad \boxed{\lambda = \frac{x^T A x}{\|x\|^2}} \quad \text{Rayleighkvot}$$

Om u_1 egenvektorn från potensmetoden är motsvarande egenvärde

$$\lambda = \frac{u_1^T A u_1}{\|u_1\|^2} = u_1^T A u_1$$

Näst största egenvärdet (Deflation)

Antag A symmetrisk och $A = TDT^T$

med $T = (u_1 \dots u_n)$,

$$\Rightarrow \quad A = \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

Om vi bestämt λ_1 och u_2 kan vi bilda

$$A_2 = A - \lambda_1 u_1 u_1^T = \sum_{i=2}^{n} \lambda_i u_i u_i^T$$

Om $|\lambda_2| > |\lambda_3|$ kan vi bestämma (λ_2, u_2) med potensmetoden ty λ_2 största egenvärde till A_2. Kan upprepas.

Tekniken kallas <u>Deflation</u>

Invers iteration

Om $|\lambda_n| > 0 \Rightarrow \lambda_i = 0 \;\forall i$

$\Rightarrow A^{-1}$ existerar och har egenvärden λ_i^{-1}

$Av_i = \lambda_i v_i \Leftrightarrow v_i = A^{-1} v_i \Rightarrow A^{-1} v_i = \lambda_i^{-1} v_i$

Så (λ_i^{-1}, v_i) egenpar till A^{-1}.

Dominerande egenparet är (λ_n^{-1}, v_n)

Förslag: finn λ_n med potensmetoden för A^{-1}

Vill dock undvika A^{-1}. Använd

$$A y_{k+1} = x_{k+1} \quad \text{ist. för} \quad y_{k+1} = A^{-1} x_{k+1}$$

Många lika högerled → LU-faktorisering

Inversiteration

Välj startvektor x_0

LU-faktorisera med pivotering: $PA = LU$

Iterera:

Lös $Lz_k = Px_k$

Lös $Uy_{k+1} = z_k$

Bilda $x_{k+1} = \|y_{k+1}\|^{-1} y_{k+1}$

Tillsammans med Rayleighkvot får man (λ_n, u_n)

Anm: Kan ej använda deflation ty alla egenvärden måste vara nollskillda.

Invers iteration med skift

OBS: Om $\{\lambda_i\}_{i=1}^{n}$ egenvärden till A

$$\det(A - \sigma I - \lambda I) = (\lambda_1 - \sigma - \lambda)(\lambda_n - \sigma - \lambda)\cdots$$

$\Rightarrow \{\lambda_i - \sigma\}_{i=1}^{n}$ är egenvärden till $A - \sigma I$

Inversiteration på $A - \sigma I$ ger egenparet med minst $|\lambda_i - \sigma|$ d.v.s. egenvärdet närmast σ är det vi hittar.

Flera egenvärden lika nära $\Rightarrow$ metod fungerar ej.

Konvergensen blir snabbare om σ ligger närmare egenvärdet.

Modifikation: Uppdatera σ med approx. av egenvärdet (t.ex. med rayleighkvot) $\Rightarrow$ snabbar upp processen.

Rayleighkvot-iteration

Välj startskift σ_0

Välj startvektor x_0

Iterera:

LU-faktorisera $P(A-\sigma_k I) = LU$

Lös $Lz_k = Px_k$

Lös $Uy_{k+1} = z_k$

Bilda $x_{k+1} = \|y_{k+1}\|^{-1} y_{k+1}$

Bilda $\sigma_{k+1} = x_{k+1}^T A x_{k+1}$

OBS: Om $\sigma = \lambda_i$ blir $A - \sigma I$ singulär $\Rightarrow$ Numeriska problem om σ_k blir för nära λ_i. Behöver eventuellt lägga på en liten konstant på σ_k

Snabb men kan få stabilitetsproblem

QR-iteration

Transformationsmetoder är stabilare och kan ge alla egenpar samtidigt.

QR-iteration

Låt $A_0 = A$

Iterera:

QR-faktorisera $Q_k R_k = A_{k-1}$

$A_k = R_k Q_k$

Vi får Q_k ortogonal och

$$Q_k A_k \underbrace{Q_k^T}_{Q_k^{-1}} = Q_k R_k Q_k Q_k^T = A_{k-1}$$

Så A_k och A_{k-1} similära => samma egenvärden

Om alla egenvärden har olika storlek konvergerar A_k mot en uppåt triangulär matris => egenvärden på diagonalen.

A symmetrisk ger $A_k \to D$ diagonal

$$\Rightarrow \quad A = QDQ^T$$

där $Q = Q_1 Q_2 \cdots Q_n$

(Teori visas ej i denna kurs)

Konvergens snabbas upp med skift

QR-iteration med skift

Låt $A_0 = A$

Iterera:

QR-faktorisera $Q_k R_k = A_{k-1} - \sigma_k I$

$A_k = R_k Q_k + \sigma_k I$

Knepigt att välja σ_k men ger snabb konvergens. Deflation kan också användas.

Ortogonal iteration.

Om man vill beräkna de p största egenparen kan man använda detta "mellanting".

Ortogonal iteration

Välj X_0 $n \times p$-matris med rang p (startgissning)

Iterera:

QR-faktorisera kompakt $Q_k R_k = X_{k-1}$

$X_k = A Q_k$

OBS: om v första kolonn i x_{k-1} så är $\frac{v}{\|v\|}$ första kolonn i Q_k. Metoden ger växelvis normering och multiplikation med A på första kolonnen d.v.s. potensmetoden.

Valet $p = n$ och $X_0 = I$ ger QR-metoden

A verkar på ortogonala vektorer för att förhindra att alla kolonner i X_k konvergerar mot största egenparet.

Lana 200525

Optimering

Matematiskt formulerade problem:
Maximera eller minimera en funktion under givna bivillkor (kallas matematisk programmering)

Generellt optimeringsproblem

minimera $f(x)$

då $\begin{cases} g(x) = 0 \\ h(x) \leq 0 \end{cases}$

$x \in \mathbb{R}^n$, n variabler att optimera

$f: \mathbb{R}^n \to \mathbb{R}$ OBJEKTFUNKTION

$g: \mathbb{R}^n \to \mathbb{R}^m$ LIKHETSBIVILLKOR

$h: \mathbb{R}^n \to \mathbb{R}^k$ OLIKHETSBIVILLKOR

Om $m=0$ och $k=0$: optimering utan bivillkor

$h(x) \leq 0$ avser elementvis olikhet

För maxproblem: $\max f \Leftrightarrow \min(-f)$

$$h \geq 0 \Leftrightarrow -h \leq 0$$

Relationer av typen $\leq$, $\geq$ o $=$ kan hanteras, strikt olikhet ej praktiska.

Olikhetsbivillkor beskriver ofta resursbegränsningar, icke-negativitetsvillkor etc.
Kräver speciella metoder, ej i denna kurs.

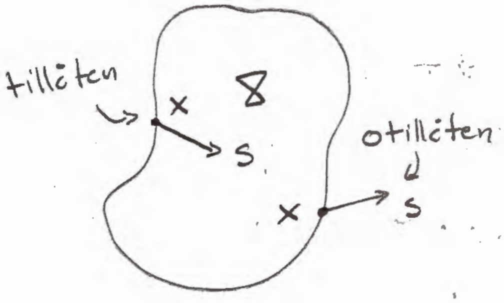
Generella begrepp

Mängden $\mathcal{X} = \{x \in \mathbb{R}^n \mid g(x) = 0,\ h(x) \leq 0\}$ kallas TILLÅTNA OMRÅDET. Problemet kan formuleras $\min_{x \in \mathcal{X}} f(x)$.

- $x^* \in \mathcal{X}$ är ett GLOBALT MINIMUM om
  $f(x^*) \leq f(x) \quad \forall x \in \mathcal{X}$
- $x^* \in \mathcal{X}$ är ett LOKALT MINIMUM om
  $f(x^*) \leq f(x) \quad \forall x \in \mathcal{X}$ i en omgivning av x^*

Om $f(x^*) < f(x)$ för $x \neq x^*$ har vi STRIKT eller UNIKT globalt/lokalt min.
(Brukar ej säga unikt lokalt minimum)

Riktningar



En riktning $s \in \mathbb{R}^n$ i x kallas TILLÅTEN om

$$\exists \delta_1 > 0 : \ x + \alpha s \in \mathcal{X} \quad \forall \alpha \in (0, \delta_1)$$

annars OTILLÅTEN

En riktning $s \in \mathbb{R}^n$ kallas DESCENTRIKTNING eller AVTAGANDERIKTNING i x om

$$\exists \delta_2 > 0 : \ f(x + \alpha s) < f(x) \quad \forall \alpha \in (0, \delta_2)$$

Vanliga metoder för optimering.

1) Börja i $x = x_0 \in \mathbb{X}$
2) Hitta en tillåten descentriktning s i x
3) Ta ett steg i riktning s, d.v.s

$$x \longrightarrow x + hs$$

För vissa metoder: h väljs så att $f(x+hs)$ minimeras m.a.p. h. kallas LINJESÖKNING (endimensionell optimering)

4) Upprepa tills det inte finns någon tillåten descentriktning. då är man i lokalt minimum.

Matematiska verktyg.

Antag $f \in C^2$, Inga bivillkor

Gradienten: $\nabla f(x) = \left(\frac{\partial f}{\partial x_1} \; \cdots \; \frac{\partial f}{\partial x_n} \right)^T$

Jacobian av gradient

Hessianen: $H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots \\ \vdots & & \end{pmatrix} = J(\nabla f(x))$

Andraderivator kommuterar ty $f \in C^2$

$\Rightarrow H^T = H$ d.v.s. H symmetrisk

$\Rightarrow H$ har reella egenvärden

Taylorutveckla f i $x \in \mathbb{R}^n$ kring $a \in \mathbb{R}^n$ (ordn. 2)

$$T_2(x) = f(a) + \underbrace{(\nabla f(a))^T (x-a)}_{\text{skalärprod.}} + \frac{1}{2} \underbrace{(x-a)^T H(a) (x-a)}_{\text{kvadratisk form}}$$

$$T_2(x) = f(a) + (\nabla f(a))^T (x-a) + \frac{1}{2}(x-a)^T H(a)(x-a)$$

Om $\nabla f(a) = 0$ är a en kritisk punkt.

Om $\nabla f(a) = 0$ och $H(a)$ positivt definit,

d.v.s. positiva egenvärden har $T_2(x)$ globalt minimum i a $\Rightarrow$ f har lokalt minimum

(ty restterm försumbar nära a)

P.s.s

- $\nabla f(a) = 0$ & $H(a)$ neg. def
  $\Rightarrow$ lokal maxpunkt
- $\nabla f(a) = 0$ & $H(a)$ indefinit
  $\Rightarrow$ sadelpunkt
- $\nabla f(a) = 0$ & $H(a)$ semidefinit
  $\Rightarrow$ måste utveckla längre.

<u>Exempel</u> $f(x) = f(x_1, x_2) = x_1^4 + x_1^2 + 2x_2^2 - 2x_1x_2$

Taylorutvecklar i $x = a = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$$\nabla f(x) = \begin{pmatrix} 4x_1^3 + 2x_1 - 2x_2 \\ 4x_2 - 2x_1 \end{pmatrix} \overset{x=a}{=} \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$H(x) = \begin{pmatrix} 12x_1^2 + 2 & -2 \\ -2 & 4 \end{pmatrix} \overset{x=a}{=} \begin{pmatrix} 14 & -2 \\ -2 & 4 \end{pmatrix}$$

kring a

$$T_2(x) = 2 + \begin{pmatrix} 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} x_1 - 1 & x_2 - 1 \end{pmatrix} \begin{pmatrix} 14 & -2 \\ -2 & 4 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 1 \end{pmatrix}$$

$$= 7x_1^2 + 2x_2^2 - 2x_1x_2 - 8x_1 + 3$$

Optimering utan bivillkor

=> Lös $\nabla f(x)=0$

Kontrollera minimalitet

Optimering med likhetsbivillkor

Ett alternativ: Lagranges multiplikatormetod

Inför Lagrangefunktionen

$$L(x,\lambda) = f(x) + \underbrace{\lambda^T g(x)}_{\text{"straffterm"}}, \qquad \lambda \in \mathbb{R}^m \leftarrow \text{Lagrange-multiplikatorer}$$

Sök min m.a.p. x och max m.a.p. λ

=> sök extrempunkt (sadelpunkt) till $L(x,\lambda)$

$$\nabla L(x,\lambda) = 0 \iff \begin{cases} \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0 \\ g(x) = 0 \end{cases}$$

Lös med metoder för icke-linjära ekvationssystem (t.ex. Newtons metod)

Generella resultat

1) $s \in \mathbb{R}^n$ är en descentriktning om $\nabla f(x)^T s < 0$
2) Om x^* lokalt min. så finns ingen tillåten descentriktning i x^*
3) Om x^* lokalt min. i det inre av tillåtna området är $\nabla f(x^*) = 0$.

Endim. optimering.

Viktigt då $n = 1$ eller vid linjesökning

Utan bivillkor: $\min_{x \in \mathbb{R}} f(x)$ där $f : \mathbb{R} \to \mathbb{R}$

Ibland $\min_{x \in [a,b]} f(x)$, typiskt $[a,b] = [0,1]$

Newtons metod

Kom ihåg: Newton för $g(x) = 0$:

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Vill minimera $f(x)$ $\Rightarrow$ sätt $g(x) = f'(x)$.

Ger Newtons metod för minimering.

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Konvergensegenskaper som för rotlösning

Nackdelar:

- Hittar extrempunkter, d.v.s. kan hitta max istället för min. Måste kontrollera resultatet.
- Kräver $f'(x)$ och $f''(x)$ i alla iterationer.

Modifierad Newton

Approximera $f''(x_k)$ med $f''(x_0)$ eller approximation av $f''(x_0)$.

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_0)}$$

Sämre konvergens men behöver ej $f''(x_k)\ \forall k$

Sekantmetoden

Som Newton men approximera

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

d.v.s. sekanten av $f'(x)$ i x_{k-1} och x_k

Som sekantmetod för rotlösning men med $f'(x)$

$$x_{k+1} = x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}$$

Slipper $f''(x_k)$ men behöver två startvärden.

Lana 200526

Optimering forts.

Repetition

Endim optimering utan bivillkor, d.v.s. sök $\min_x f(x)$ då $x \in [a,b]$.

Newton, Modifierad Newton, sekantmetoden kräver $f'(x)$. Diffkvot ej alltid bra. Idag: Gyllene snitt-metoden och polynomapproximation kräver bara $f(x)$ och inte $f'(x)$.

Gyllenesnitt-metoden

Fungerar unimodala funktioner

Def $f: [a,b] \to \mathbb{R}$ är UNIMODAL om det finns en minpunkt x^* och f är strängt avtagande för $x \in [a, x^*]$ och strängt växande för $x \in [x^*, b]$.

Unimodal

Unimodal

Ej unimodal

Gyllenesnitt metoden

Börja med två inre punkter $a < x_1 < x_2 < b$

$$\boxed{f(x_1) < f(x_2) \Rightarrow x^* \in [a, x_2]}$$

ty om $x^* \in (x_2, b]$ ger unimodalitet att f är strängt avtagande på $[a, x_2]$. Motsäger att $f(x_1) < f(x_2)$. P.S.S.

$$\boxed{f(x_2) < f(x_1) \Rightarrow x^* \in [x_1, b]}$$

Vi får nytt intervall att undersöka.
Kan behålla inre punkten för att spara funktionsberäkningar.

Exempel: Ett steg

$f(x_1) < f(x_2)$ ger

$f(x_2) < f(x_1)$ ger

Vi vill helst ha samma proportioner, d.v.s. skalfaktor g i båda fall.

Hur får vi samma skalfaktor

$$\begin{cases} g(b-a) = (b'-a') = x_2 - a \\ g(x_2-a) = (x_2'-a') = x_1 - a \end{cases}$$

$$\Rightarrow \begin{cases} x_1 - a = g(x_2-a) = g^2(b-a) \\ g(b-a) = b''-a'' = b - x_1 \quad \Rightarrow x_1 = b - g(b-a) \end{cases}$$

$$\Rightarrow g^2(b-a) = b-a-g(b-a)$$

$$\Rightarrow g^2 = 1-g$$

$$\Rightarrow g = \frac{\sqrt{5}-1}{2}$$ d.v.s. gyllene snittet

Så för varje iteration:

$x_1 = b - g(b-a)$

$x_2 = a + g(b-a)$

Ger att vi kan återanvända funktionsberäkningar och behålla propotioner.

Upprepa k gånger ger intervall med längd $g^k(b-a)$ som innehåller x^*.

Liknar intervallhalvering för rotsökning.

Polynomapproximation

Fungerar om f är konvex och har min i $[a,b]$ (medför unimodalitet på $[a,b]$).

- Börja med tre punkter, t.ex

$$x_1 = a \qquad x_2 = \frac{1}{2}(a+b) \qquad x_3 = b$$

Låt $p_2(x)$ vara interpolationspolynomet i x_1, x_2, x_3 av grad 2. Låt x_4 vara minpunkt till $p_2(x)$ (enkel att beräkna)

- Sortera $\{x_1, x_2, x_3, x_4\}$ ger punkter $y_1 < y_2 < y_3 < y_4$. Används som i gyllenesnittmetoden men $a = y_1$, $x_1 = y_2$, $x_2 = y_3$, $b = y_4$.

  OBS! Får ej samma proportioner
- Jämför $f(x_1)$ med $f(x_2) \Rightarrow$ nytt intervall med tre punkter, upprepa metoden. Avbryt om intervallet är kort eller om x_4 hamnar nära x_1, x_2 eller x_3.

Ger snabbare konvergens än gyllenesnittmetoden utan att kräva fler funktionsberäkningar (en per iteration)

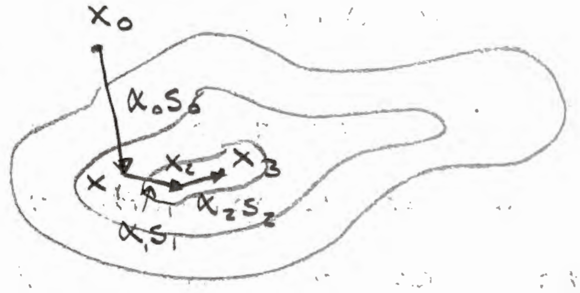
Flerdimensionell optimering utan bivillkor

Problem: $\min_{x \in \mathbb{R}^n} f(x) \quad f: \mathbb{R}^n \to \mathbb{R}$

Sökmetoder: Iterera $x_{k+1} = x_k + \alpha_k s_k$

där s_k sökriktning och α_k steglängd

Ex



Val av sökriktning

Steepest descent / Brantaste lutning-metoden

(SD) $\boxed{s_k = -\nabla f(x_k)}$

f avtar snabbast i riktning $-\nabla f(x_k)$, ortogonalt mot nivåkurvorna.

$s_k = -\nabla f(x_k)$ descentriktning om x_k ej minpunkt

ty $\nabla f(x_k)^T s_k = -\nabla f(x_k)^T \nabla f(x_k) = -\|\nabla f(x_k)\|^2 \le 0$

Likhet om $\nabla f(x_k) = 0$, d.v.s. minpunkt.

Avbryt när $\|\nabla f(x_k)\|$ är liten.

Newtons metod

Lös $\nabla f(x)=0$ med Newtons metod för att finna rötter till ekvationssystem.

d.v.s. lös $J(\nabla f(x_k))s_k = -(\nabla f(x_k))$

$$\Rightarrow \boxed{H(x_k)s_k = -\nabla f(x_k)}$$

$x_{k+1} = x_k + s_k$ från Newton antyder $\alpha_k = 1$ bra.

Kräver $H(x)$ men kan konvergera snabbt.

Kan konvergera mot lokal maxpunkt istället, men om $H(x)$ positivt definit $\forall x \in \mathbb{R}^n$

($\Leftrightarrow f(x)$ konvex) så är $H^{-1}(x)$ positivt definit

(egenvärden $\lambda_i^{-1} > 0$)

$$\Rightarrow \nabla f(x_k)^T s_k = -\underbrace{\nabla f(x_k)^T H^{-1}(x_k)\nabla f(x_k)}_{\text{pos. def. kvadratisk form}} \leq 0$$

d.v.s. ger descentriktning för konvexa f.

Andragradspolynom löses på en iteration.

Kvasi-Newtonmetoder

Approximera $H(x_k)$ med B_k, t.ex $B_0 = H(x_0)$ och uppdatera med Broydens metod.

$$\left|\begin{aligned} B_k s_k &= -\nabla f(x_k) \\ x_{k+1} &= x_{k+1} + s_k \\ B_{k+1} &= B_k + \frac{1}{\|s_k\|_2^2}\left(\nabla f(x_{k+1}) - \nabla f(x_k) - B_k s_k\right)s_k^T \end{aligned}\right.$$

Andra val av B_k finns.

Som Newton: descentriktning om B_k är positivt definit

Konjugerande gradientmetoden

Steepest descent kan ge många korta "sicksack"-steg. Kan bli bättre om man tar med lite av föregående sökriktning.

(KG) $$\boxed{s_k = -\nabla f(x_k) + \beta_k s_{k-1}, \quad \beta_k = \frac{\|\nabla f(x_k)\|_2^2}{\|\nabla f(x_{k-1})\|_2^2}}$$

Första steget som SD (d.v.s. $\beta_0 = 0$)

Om f andragradspolynom blir $s_k^T H s_j = 0$ då $k \neq j$ om H pos. def.

$\Rightarrow$ ortogonala sökriktningar

$\Rightarrow$ max n iterationer

Mycket bättre än SD för andragradspolynom!

Då vi valt sökriktning kan vi välja steglängd. Vill ha $\min_{\alpha_k} g(\alpha_k)$, $g(\alpha_k) = f(x_k + \alpha_k s_k)$

$\Rightarrow$ Endim. linjesökning

Specialfall om f andragradspolynom:

$$g(\alpha_k) = f(x_k + \alpha_k s_k) =$$
$$= f(x_k) + \alpha_k \nabla f(x_k)^T s_k + \frac{1}{2}\alpha_k^2 \underset{\text{konstant}}{\underbrace{s_k^T H s_k}}$$

$g'(\alpha_k) = 0$ ger

$$0 = g'(\alpha_k) = \nabla f(x_k)^T s_k + \alpha s_k^T H s_k$$

$$\Rightarrow \boxed{\alpha_k = -\frac{\nabla f(x_k)^T s_k}{s_k^T H s_k}}$$

För Newtons metod är $\alpha_k = 1$ OK.

Vid konvergensproblem kan α_k väljas mindre i början (Dämpad Newton)

Exempel

minimera $f(x) = x_1^2 + 2x_1x_2^2 + x_2^3 - 4x_1 - 6x_2$ utan bivillkor med Newtons metod och exakt linjesökning (En iteration)

$$\nabla f(x) = \begin{pmatrix} 2x_1^2 + 2x_1 - 4 \\ 3x_2^2 + 4x_1x_2 - 6 \end{pmatrix} \quad H(x) = \begin{pmatrix} 2 & 4x_2 \\ 4x_2 & 4x_1 + 6x_2 \end{pmatrix}$$

Givet: $x^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

pos def!
Newton konvergerar i omgivningen.

$$\nabla f(x^{(0)}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad H(x^{(0)}) = \begin{pmatrix} 2 & 4 \\ 4 & 10 \end{pmatrix}$$

Newtons metod: $H(x^{(k)})\, s_{k+1} = -\nabla f(x^{(k)})$

$$\Rightarrow \left[\begin{array}{cc|c} 2 & 4 & 0 \\ 4 & 10 & -1 \end{array}\right] \Rightarrow s_1 = \begin{pmatrix} 1 \\ -1/2 \end{pmatrix}$$

Linjesökning:

$$g(\alpha) = f(x^{(0)} + \alpha s_1) = \ldots = \frac{3}{8}\alpha^3 + \frac{1}{4}\alpha^2 - \frac{1}{2}\alpha - 6$$

$$g'(\alpha) = 0 \Rightarrow \alpha_1 = \frac{2}{9}(\sqrt{10} - 1) \approx 0.4805$$

positiv rot, men α måste vara positiv för descentriktning

$$x^{(1)} = x^{(0)} + \alpha s_1 = \begin{pmatrix} 1 + \frac{2}{9}(\sqrt{10} - 1) \\ 1 - \frac{1}{9}(\sqrt{10} - 1) \end{pmatrix} \approx \begin{pmatrix} 1.4805 \\ 0.7597 \end{pmatrix}$$

$$f(x^{(1)}) \approx -6.1409 \quad //$$

ska egentligen fortsätta tills $\|\nabla f(x^{(k)})\|$ litet.