

2.4. Eigenwertprobleme

Rückblick auf die Theorie

Nipp/Stoffer 145-170

Sei $A \in \mathbb{R}^{n \times n}$ eine quadratische Matrix
 $\underline{x} \neq 0, \underline{x} \in \mathbb{R}^n$ ein nicht verschwindender Vekt.

Eigenwertproblem:

(1) $A \underline{x} = \lambda \cdot \underline{x}$

Zur Matrix A suchen wir einen Vektor $\underline{x} \neq 0$,
der bei der Abbildung $\underline{x} \mapsto A\underline{x}$ nur gestreckt
wird (mit Faktor λ).

\underline{x} heisst Eigenvektor zum Eigenwert λ

(i) Charakteristisches Polynom

(1) $\Rightarrow (\lambda I - A) \underline{x} = 0$

$I = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$

Einheitsmatrix

$\Rightarrow p(\lambda) := \det(\lambda I - A) = 0$

wegen $\underline{x} \neq 0$

$p(\lambda)$ heisst "charakteristisches Polynom von A":

$p(\lambda) = \lambda^n - \lambda^{n-1} \cdot \text{tr}(A) + \dots + (-1)^n \det(A)$
Spur, $\sum_{k=1}^n a_{kk}$ Determinante

Im allgemeinen: $\lambda_k \in \mathbb{C}, \lambda_k$ komplex

Die lineare Algebra offeriert folgende Technik:

- $p(\lambda)$ bestimmen Matlab:
 $c = \text{poly}(A)$
liefert Koeffizienten von p
- Nullstellen $\lambda = \text{roots}(c)$
liefert Nullst. von p
- $\lambda_1, \dots, \lambda_n$ von p bestimmen

Dies ist praktisch **UNBRAUCHBAR!**

Grund: Die Nullstellen λ_k des char. Pol. $p(\lambda)$ sind durch seine Koeffizienten unscharf bestimmt. Der Weg über $p(\lambda)$ bringt einen Informationsverlust.

Beispiel. $A = \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{pmatrix} \Rightarrow p(\lambda) = \lambda^2 - 2\lambda + 1 - \varepsilon^2$
 $p(\lambda) = 0 \Rightarrow \lambda_{1,2} = 1 \pm \varepsilon$
 Sei (in Matlab) $\varepsilon = 5 \cdot 10^{-9}$
 $\Rightarrow \tilde{p}(\lambda) = \lambda^2 - 2\lambda + 1$, da $1 - \varepsilon^2 = 1$ (in Matlab)
 $\Rightarrow \tilde{\lambda}_1 = \tilde{\lambda}_2 = 1$

Exakt gilt aber

$$\lambda_1 = 1.0000000050000000$$

$$\lambda_2 = 0.9999999950000000$$

Beide Zahlen wären gut darstellbar!

Bemerkung: Eigenwerte können zusammenfallen, mehrfach auftreten

→ Algebraische Vielfachheit

(ii) Diagonalform

Lemma: Zu verschiedenen Eigenwerten gehören linear unabhängige Eigenvektoren

Betrachte A mit lauter verschiedenen Eigenwerten $\lambda_1, \lambda_2, \dots, \lambda_n$. Dann existieren n lin. unabhängige EV $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n$:

$$A \underline{x}_k = \underline{x}_k \cdot \lambda_k, \quad k = 1, \dots, n$$

Alle n Vektorgleichungen nebeneinander schreiben:

$$A \begin{bmatrix} \vdots & \vdots & \vdots \\ \underline{x}_1 & \underline{x}_2 & \dots & \underline{x}_n \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \underline{x}_1 \lambda_1 & \dots & \underline{x}_n \lambda_n \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \underline{x}_1 & \dots & \underline{x}_n \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_n \end{bmatrix}$$

$T =$ Matrix mit

EV. als Kolonnen, **regulär!**

$D = \text{diag}(\lambda_k)$

SATZ: Falls $A \in \mathbb{R}^{n \times n}$ n linear unabhängige Eigenvektoren \underline{x}_k hat, gilt

$$\boxed{AT = TD}$$

Dabei ist $T = (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n)$, $D = \text{diag}(\lambda_k)$.

Folgerungen: $A = T D T^{-1}$ } Ähnlichkeits-
 $D = T^{-1} A T$ } transformation

Matlab: $[T, D] = \text{eig}(A)$ liefert T, D
 $\text{eig}(A)$ liefert Eigenwerte als Kolonnenvek

(iii) Eigenräume

Seien $\underline{x}_1, \underline{x}_2$ Eigenvektoren zum selben Ew. λ :

$$\begin{array}{l|l} A \underline{x}_1 = \lambda \underline{x}_1 & \cdot \alpha_1 \\ A \underline{x}_2 = \lambda \underline{x}_2 & \cdot \alpha_2 \end{array}$$

$\Rightarrow A(\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2) = \lambda(\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2)$

d.h. $\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2$ ist auch EV. zu λ , für alle α_1, α_2

Fälle: - $\underline{x}_1, \underline{x}_2$ sind linear abhängig, $\underline{x}_2 = \gamma \underline{x}_1$

$\Rightarrow \{\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2\} = \{c \underline{x}_1\}$ bilden
1-dimensionalen Eigenraum

- $\underline{x}_1, \underline{x}_2$ sind linear unabhängig

$\Rightarrow \{\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2\}$ bilden 2-dimensionalen Eigenraum

Zusammenfassung

- Eigenvektoren sind nur (höchstens) auf einen freien Faktor bestimmt \rightarrow z.B. Normierung $\|\cdot\|=1$
- Alle Eigenvektoren zum Eigenwert λ bilden einen linearen Raum: den Eigenraum zu λ
- Ein einfacher Eigenwert (algebraische Vielfachheit = 1) hat immer einen 1-dimens. Eigenraum
- Definition: Die Dimension des Eigenraumes zu einem mehrfachen Eigenwert heißt die geometrische Vielfachheit des Eigenwertes (=Anzahl der linear unabhängigen Eigenvektoren).

SATZ: geometrische Vielf. \leq algebraische Vielf.

Beispiele

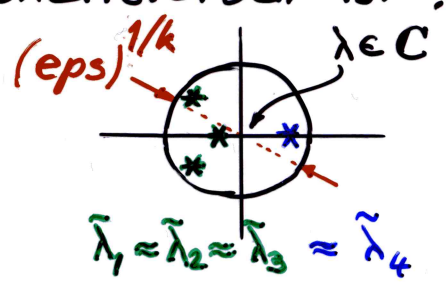
- $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\lambda_1 = \lambda_2 = 1$, alg. V. = 2
 $A - \lambda I$ hat Rang 0 \Rightarrow 2 lin. unabh. Ev, geom V. = 2
- $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $\lambda_1 = \lambda_2 = 1$, alg. V. = 2
 $A - \lambda I = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ hat Rang 1
 \Rightarrow es existiert nur 1 Eigenv. $\Delta_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, geom. V. = 1

SATZ: Eine Matrix ist genau dann diagonalisierbar, wenn für jeden Eigenwert gilt: geom. V. = algebr. V.

Folgerung: $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ ist nicht diagonalisierbar: es existiert kein T mit $A = TDT^{-1}$
↑
diagonal

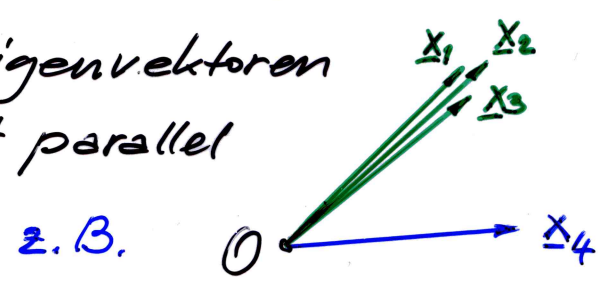
Matlab: Welches ist die Wirkung des Befehls $[T, D] = \text{eig}(A)$, falls A nicht diagonalisierbar ist?

- ein k -facher EW. λ wird in k einfache EW. aufgespalten:
 $|\tilde{\lambda}_i - \lambda| \leq \text{const} (\text{eps})^{1/k}$



Damit wird A "diagonalisierbar gemacht".

- Die zugehörigen Eigenvektoren werden (teilweise) fast parallel



(iv) Symmetrische Matrizen, $A = A^T$

- Es gilt:
- Alle Eigenwerte λ_k sind **reell**
 - $A = A^T$ ist immer **diagonalisierbar**, auch bei mehrfachen Ew.
 - T kann **orthogonal** gewählt werden, $T^{-1} = T^T$

Bemerkung:

- Dasselbe gilt für komplexe Matrizen $A \in \mathbb{C}^{n \times n}$, falls $A = (\bar{A})^T =: A^H$ (hermitesch)
- T ist dann "unitär", d.h. $T^{-1} = T^H$
- Matlab: $A' := (\bar{A})^T$

2.5. Numerik der Eigenwertprobleme

Forget about the characteristic polynomial!

Aufgabe

- (a) Bestimmung einzelner Ev. und EW., vor allem des **absolut grössten** oder des **abs. kleinsten** Eigenwertes

- (b) Bestimmung aller Eigenwerte und Eigenvektoren

Technik

Vektoriteration

Startvektor \underline{x}_0 ;

$$\underline{x}_{k+1} = A \underline{x}_k$$

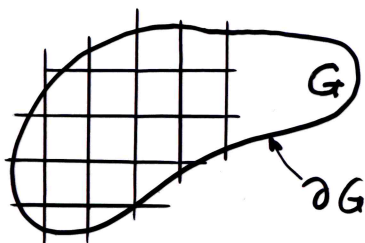
Matrixzerlegung

$$T^{-1}AT = D$$

a) Vektoriteration

Typisches Problem: Schwingungen eines Kontinuums in diskreter Approximation

z. B. Membran



kontinuierlich:

$$\begin{aligned}\Delta u &= \lambda u \text{ in } G \\ u &= 0 \text{ auf } \partial G\end{aligned}$$

Diskrete Approximation:

Sei $\underline{x} \in \mathbb{R}^n$ der Vektor der Auslenkungen in den inneren Gitterpunkten:

$$\begin{aligned}A \underline{x} &= \lambda \underline{x} \\ A &\in \mathbb{R}^{n \times n}\end{aligned}$$

Spezielle Situation:

- A gross, z.B. $n \sim 1000$ oder mehr
- A "schwach besetzt" (*sparse*), d.h. pro Zeile nur wenige Elemente $\neq 0$
- $A = A^T$ (A *symmetrisch*) gilt häufig
- Nur wenige Eigenwerte mit Eigenvektoren am unteren Ende des Spektrums gesucht (grosse Ew. sind "garbage"!).

Algorithmus: Wähle (fast) belieb. Startvektor \underline{x}_0

Iteration $\underline{x}_{k+1} = A \underline{x}_k, \quad k = 0, 1, \dots$

Resultat: Folge $\{\underline{x}_k\}$ konvergiert der Richtung nach gegen Ew. zum abs. grössten Ew.

Praktisch: Normierung nach jedem Schritt:

$$\underline{x}_{k+1} = \frac{A \underline{x}_k}{\|A \underline{x}_k\|}, \quad k = 0, 1, \dots$$

Unter schwachen Voraussetzungen gilt:

$$\lambda_{\max} = \lim_{k \rightarrow \infty} \frac{\underline{x}_k^T A \underline{x}_k}{\underline{x}_k^T \underline{x}_k} \quad (\text{Rayleigh-Quotient})$$

$$\underline{v} = \lim_{k \rightarrow \infty} \underline{x}_k \quad \text{Eigenvektor zu } \lambda_{\max}$$

Bemerkungen:

- Berechnung von λ_{\max} allein billiger als alle λ 's
- Man muss A nicht speichern!
Es genügt, die Abbildung $\underline{x} \mapsto A \underline{x}$ zu programmieren ("Operatorprinzip").
Damit Ausnutzung der Sparsity möglich!
- Wir brauchen aber λ_{\min} !

Rückwärtsiteration:

$$\underline{x}_k = A^{-1} \underline{x}_{k+1} = A \setminus \underline{x}_{k+1} \quad \text{lin. Gleichsystem lösen!}$$

für $k = -1, -2, \dots$

$$\left(\text{bzw. mit Normierung } \underline{x}_k = \frac{A^{-1} \underline{x}_{k+1}}{\|A^{-1} \underline{x}_{k+1}\|} \right)$$

konvergiert für $k \rightarrow -\infty$ gegen den Eigenv. zum abs. kleinsten EW, λ_{\min} , denn es gilt

$$(\lambda \text{ ist EW. von } A) \Rightarrow \left(\frac{1}{\lambda} \text{ ist EW. von } A^{-1} \right).$$

zum selben E'vektor!

Konvergenztheorie der Vektoriteration

Annahme: $A \in \mathbb{R}^{n \times n}$ habe n linear unabh. Eigenvektoren \underline{v}_j (z. B. erfüllt, falls alle Eigenwerte λ_j verschieden):

$$A \underline{v}_j = \lambda_j \underline{v}_j \quad (j=1, \dots, n)$$

Startvektor $\underline{x}_0 = \sum_{j=1}^n c_j \underline{v}_j$

(Zerlegung von \underline{x}_0 in der Basis \underline{v}_j)

Iteration: $\underline{x}_1 = A \underline{x}_0 = \sum_{j=1}^n c_j A \underline{v}_j = \sum_{j=1}^n c_j \lambda_j \underline{v}_j$

$$\underline{x}_2 = A \underline{x}_1 = \sum_{j=1}^n c_j \lambda_j (A \underline{v}_j) = \sum_{j=1}^n c_j \lambda_j^2 \underline{v}_j$$

(1) $\underline{x}_k = \sum_{j=1}^n c_j \lambda_j^k \underline{v}_j, \quad k=0, 1, \dots$

Voraussetzung "nur ein dominanter Eigenwert"

Sei $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$

Dann folgt aus (1):

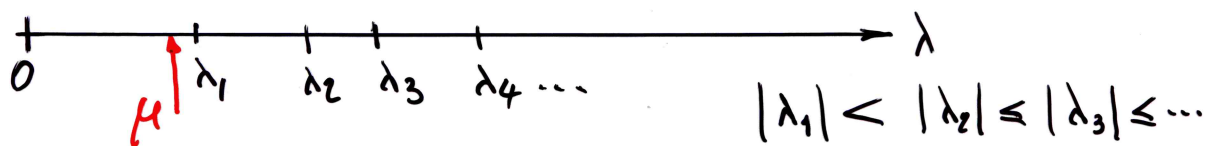
(2) $\underline{x}_k = \lambda_1^k \left[c_1 \underline{v}_1 + \underbrace{\left(\frac{\lambda_2}{\lambda_1} \right)^k c_2 \underline{v}_2 + \left(\frac{\lambda_3}{\lambda_1} \right)^k c_3 \underline{v}_3 + \dots}_{\rightarrow 0 \text{ für } k \rightarrow \infty} \right]$

SATZ: Sei $A \in \mathbb{R}^{n \times n}$; Eigenwerte $|\lambda_1| > |\lambda_j| \forall j > 1$; $\underline{x}_0 \in \mathbb{R}^n$ habe eine nichtverschwindende Komponente $c_1 \neq 0$ in Richtung des 1. Ev \underline{v}_1 . Dann konv. die Folge $\underline{x}_k = A \underline{x}_{k-1}$ der Richtung nach gegen \underline{v}_1 .

Bemerkungen

(i) Falls $c_1 = 0$, konvergiert \underline{x}_k praktisch dennoch gegen \underline{v}_1 . **Rundungsfehler helfen!**

(ii) Konvergenzbeschleunigung bei inverser V'It.



Fehlergesetz (2) für die inv. V'It.:

$$(3) \quad \underline{x}_k = \lambda_1^{-k} \left[c_1 \underline{v}_1 + \left(\frac{\lambda_1}{\lambda_2} \right)^k c_2 \underline{v}_2 + \mathcal{O} \left(\left(\frac{\lambda_1}{\lambda_3} \right)^k \right) \right]$$

denn:

	Matrix	Ew.	Ev.	Grund
	A	λ_k	\underline{v}_k	$A = T D T^{-1}$
	A^{-1}	λ_k^{-1}	\underline{v}_k	$A^{-1} = T D^{-1} T^{-1}$
	$A - \mu I$	$\lambda_k - \mu$	\underline{v}_k	$A - \mu I = T (D - \mu I) T^{-1}$

Verschiebungen: Sei $\lambda_1 > 0$;

wähle Verschiebung $\mu \in (0, \lambda_1)$, $\lambda_1 - \mu \ll \lambda_1$
und betrachte die Matrix

$$\tilde{A} := A - \mu I;$$

\tilde{A} hat die Eigenwerte $\tilde{\lambda}_k := \lambda_k - \mu$,
jedoch dieselben Eigenvektoren \underline{v}_k wie A .

Resultat:

Inverse Vektoriteration mit $\tilde{A} := A - \mu I$
konvergiert gegen den Eigenvektor \underline{v}_1 zum
kleinsten Ew. von A . **Lineare Konvergenz**
mit Konvergenzfaktor $\rho = \frac{\lambda_1 - \mu}{\lambda_2 - \mu}$

(iii) Simultane Vektoriteration

Idee: Iteration eines Unterraumes $U_0 \subset \mathbb{R}^n$, z.B. des Unt'r. aufg. dch. $\underline{x}_0, \underline{y}_0$:

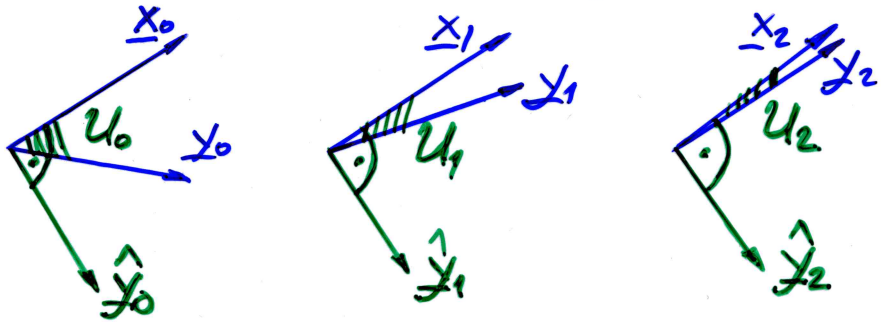
$$\underline{x}_0, \quad \underline{x}_{k+1} = A \underline{x}_k \quad (k=0,1,\dots)$$

$$\underline{y}_0, \quad \underline{y}_{k+1} = A \underline{y}_k \quad (k=0,1,\dots)$$

Hoffen: $\underline{x}_k \rightarrow \underline{v}_1, \quad \underline{y}_k \rightarrow \underline{v}_2$

Was passiert:

$\underline{x}_k, \underline{y}_k$
werden immer
mehr parallel!



Abhilfe: Nach jedem Iterationsschritt orthog. Basis $\hat{\underline{x}}_k, \hat{\underline{y}}_k$ in U_k konstruieren (mit QR-Zerlegung!) und nur diese weiter iterieren.

Resultat:

$$\text{Sei } A = A^T$$

\Rightarrow Eigenvektoren sind als paarweise orthogonal wählbar

Damit folgt:
$$\left. \begin{array}{l} \hat{\underline{x}}_k \rightarrow \underline{v}_1 \\ \hat{\underline{y}}_k \rightarrow \underline{v}_2 \end{array} \right\} \text{für } k \rightarrow \infty, \text{ Konvergenz nach Richtung}$$

b) Der QR-Algorithmus

Nipp/Stoffer
228 - 232

(41)

Der stärkste bekannte Algorithmus zur Berechnung aller EW. und EV. einer allgemeinen Matrix

Sei $A \in \mathbb{R}^{n \times n}$; $A_0 := A$

Erster
Iterations-
schritt:
 $A_0 \rightarrow A_1$

Zerlege $A_0 = Q_0 R_0 = \boxed{Q_0} \boxed{R_0}$, $Q_0^T Q_0 = I$
und bilde $A_1 = R_0 Q_0$

Wegen $R_0 = Q_0^{-1} A_0$ gilt: $A_1 = Q_0^{-1} A_0 Q_0$,

$\Rightarrow A_1$ ist ähnlich zu A_0

$\Rightarrow A_1$ hat dieselben Eigenwerte wie A_0

Naheliegender: Iterieren, wiederholen:

Algorithmus: Gegeben $A_0 = A \in \mathbb{R}^{n \times n}$

$$A_k = Q_k R_k; \quad A_{k+1} = R_k Q_k, \quad k = 0, 1, \dots$$

Bilde QR-Zerl. von A_k

Das Wunder:

SATZ: Falls A_0 lauter Eigenwerte von verschiedenem Betrag hat, konvergiert die Matrixfolge A_k gegen eine obere Dreiecksmatrix R .

Bemerkungen:

- (i) Eigenwerte von A erscheinen auf Diagonalen v. R .
- (ii) Bei betragsgleichen EW. entstehen "Kästchen" auf der Diagonalen, die nicht konvergieren.

- (iii) Für symmetrische Matrizen, $A = A^T$, führt auch das eng verwandte **Jacobi-Verfahren** zum Ziel. Dieses erlaubt einen einfachen **Konvergenzbeweis**, siehe Nipp/Stoffer, Seiten 222-228
- (iv) Eigenwertberechnung ist **ein unendlicher Prozess!**

2.6. Die Singulärwertzerlegung

The "fanciest" matrix decomposition!
 "Singular-value decomposition, SVD"

Sei $A \in \mathbb{R}^{m \times n}$ eine **beliebige** Rechtecksmatrix von **beliebigem** Rang $\rho \leq \ell := \min(m, n)$

Betrachte die Abbildung

(1) $\underline{x} \in \mathbb{R}^n \mapsto \underline{y} := A \underline{x} \in \mathbb{R}^m$

Führe in \mathbb{R}^n und \mathbb{R}^m neue **gedrehte** Koordinatensysteme ein:

(2)
$$\begin{aligned} \underline{x} &= V \underline{\xi} \in \mathbb{R}^n, & V^T V &= I_n \\ \underline{y} &= U \underline{\eta} \in \mathbb{R}^m, & U^T U &= I_m \end{aligned} \left. \vphantom{\begin{aligned} \underline{x} &= V \underline{\xi} \in \mathbb{R}^n, \\ \underline{y} &= U \underline{\eta} \in \mathbb{R}^m, \end{aligned}} \right\} \begin{array}{l} \text{orthog.} \\ \text{Matrizen} \end{array}$$

($\underline{\xi} \in \mathbb{R}^n, \underline{\eta} \in \mathbb{R}^m$ enthalten die Vektorkomponenten bez. der neuen Koordinatensysteme).

(1), (2) $\Rightarrow U \underline{\eta} = A V \underline{\xi}$

oder $\underline{\eta} = S \underline{\xi}$ mit $S := U^T A V$
 oder $A = U S V^T$

Ziel: Wahl von U, V so, dass S möglichst einfach (S in "Normalform").

SATZ: Zu jeder Matrix A gibt es orthogonale Matrizen U, V , so dass

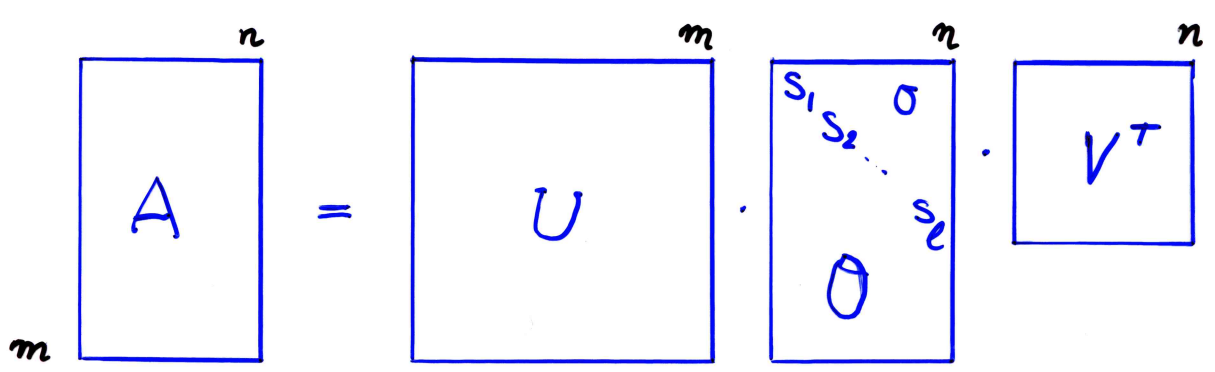
$$A = U S V^T ;$$

dabei ist S diagonal mit nichtnegativen Diagonalelementen

$$s_1 \geq s_2 \geq \dots \geq s_l \geq 0 ;$$

s_1, \dots, s_l heißen Singulärwerte von A .

Beispiel. $A \in \mathbb{R}^{m \times n}$, $m \geq n$, $l = n$



S , gleiche Dimensionen wie A

Bemerkungen

- (i) Guter, stabiler Algorithmus zur Erzeugung der SVD bekannt, ca. $10 m^3$ Operationen, nicht anfällig auf Rundungsfehler
- (ii) Matlab: $[U, S, V] = \text{svd}(A)$
- (iii) Zusammenhang mit symmetr. Ew-pr. $\Rightarrow U^T A = S V^T$

$A^T A = V (S^T S) V^T \Rightarrow s_k = \sqrt{\lambda_k}$ wobei $\lambda_k =$ Ew. von $A^T A$

Anwendungen der SVD

(i) Rang einer Matrix

Sei $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = \rho < \min(m, n)$

$$\Rightarrow S = \text{diag}(s_1, \dots, s_\rho, 0, \dots, 0)$$

Praktisch werden die s_k mit $k > \rho$ zwar klein, aber nicht Null; meist etwa

$$\frac{s_k}{s_1} \approx \text{eps} \quad \forall k > \rho$$

Maschinen-Epsilon, $\text{eps} = \min_{1+\epsilon > 1} \epsilon$ Der "numerische Rang" von A hängt also ab von der Rechengenauigkeit eps .

Wähle z.B.

$$\rho = \max_{s_k > \text{eps} \cdot s_1} (k)$$

(ii) Euklidische Norm und Kondition

(*) Sei $A \in \mathbb{R}^{n \times n}$, $A = USV^T$

$$S = \text{diag}(s_1, \dots, s_n)$$

Norm: $\|A\|_2 = \text{maximale Länge des Bildes eines Einheitsvektors}$

Aus (*) folgt sofort:

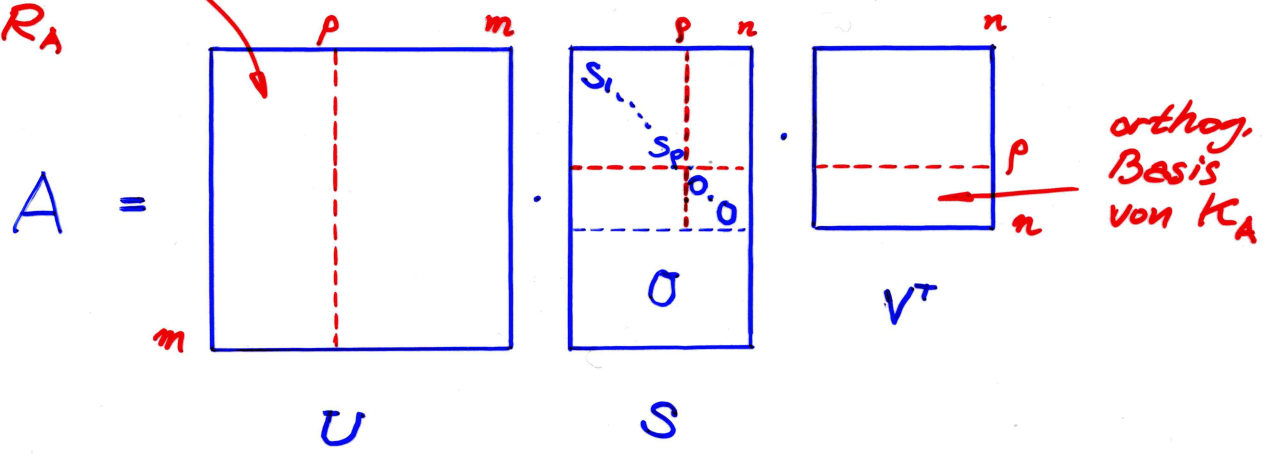
$$\|A\|_2 = s_1$$

Kondition: $\text{cond}_2(A) = s_1 / s_n$ Einsparungen: SVD ohne U, V erzeugen

(iii) Lineare Abbildungen: Kern & Bild

Sei $A \in \mathbb{R}^{m \times n}$, $m \geq n$
 $\text{rank}(A) = \rho < n$

orthog. Basis von R_A



orthog. Basis von K_A

Abbildung: $\underline{x} \mapsto \underline{y} = A \underline{x}$, $\underline{x} \in \mathbb{R}^n$, $\underline{y} \in \mathbb{R}^m$

Kern: $K_A := \{ \underline{x} \in \mathbb{R}^n \mid A \underline{x} = 0 \}$

$\dim K_A = n - \rho$

orthog. Basis: Kolonnen $\rho+1, \dots, n$ von V

Bild (Range): $R_A := \{ \underline{y} \in \mathbb{R}^m \mid \exists \underline{x} \in \mathbb{R}^n \text{ mit } \underline{y} = A \underline{x} \}$

$\dim R_A = \rho$

orthog. Basis: Kolonnen $1, \dots, \rho$ von U

(iv) Ausgleichsproblem mit nichtvollem Rang

Sei $A \in \mathbb{R}^{m \times n}$, $m \geq n$, $\text{rank}(A) = \rho < n$

Residuum:

Modellfunktionen linear abhängig!

$\underline{r} := A \underline{c} - \underline{f}$; \underline{c} so: $\| \underline{r} \|_2 \stackrel{!}{=} \min$

Sei SVD von A bekannt:

$$A = USV^T$$

$$S = \text{diag}(s_1, \dots, s_p, 0, \dots, 0)$$

Gedrehtes Residuum:

$$\tilde{r} := U^T r = S(\underbrace{V^T \underline{\xi}}_{y=?}) - \underbrace{U^T f}_g$$

komponentenweise:

$$\begin{array}{l}
 \tilde{r}_1 = s_1 y_1 - g_1 \\
 \vdots \\
 \tilde{r}_p = s_p y_p - g_p \\
 \tilde{r}_{p+1} = 0 \cdot y_{p+1} - g_{p+1} \\
 \vdots \\
 \tilde{r}_n = 0 \cdot y_n - g_n \\
 \hline
 \tilde{r}_{n+1} = -g_{n+1} \\
 \vdots \\
 \tilde{r}_m = -g_m
 \end{array}
 \left. \begin{array}{l}
 = 0 \\
 \vdots \\
 = 0 \\
 \\
 \\
 \\
 \\
 \end{array} \right\} \begin{array}{l}
 \|\tilde{r}\| = \min \\
 \leftarrow \\
 \\
 \\
 \text{fest}
 \end{array}$$

Lösung:

$$y_k = \begin{cases} g_k / s_k, & k = 1, 2, \dots, p \\ \text{beliebig}, & k = p+1, \dots, n \end{cases}$$

Es gibt ∞ viele Lösungen: $\underline{\xi} = V y$

Auswahl einer ausgezeichneten Lösung:

Wähle y_{p+1}, \dots, y_n so, dass $\|\underline{\xi}\| \stackrel{!}{=} \min$
(die Lösung minimaler Norm)

Wegen $\|\underline{\xi}\| = \|y\|$ resultiert die Minimum-Norm-Lösung mit $y_{p+1} = y_{p+2} = \dots = y_n = 0$.