



Technische Universität München

Fakultät für Mathematik



Bachelor-Arbeit

Titel

Markus Stachl

Aufgabensteller: ...

Betreuer: ...

Abgabetermin: ...

Ich erkläre hiermit, dass ich die Bachelor-Arbeit selbständige und nur mit den angegebenen Hilfsmitteln angefertigt habe.

Garching, den

Summary

Bei einer in deutscher Sprache verfassten Arbeit muss eine Zusammenfassung in englischer Sprache vorangestellt werden. Dafür ist hier Platz.

Inhaltsverzeichnis

1 Einführung	1
2 Einführung und Motivation	1
2.1 Optimale Kontrolle	1
2.2 Berechnung der Wertefunktion	1
2.3 Beispiel: das invertierte Pendel	3
3 Niedring-Rang-Approximation einer Matrix	4
3.1 Singulärwertzerlegung (SVD)	4
3.1.1 Definition	4
3.1.2 Fehlerapproximation	5
3.2 CX-Zerlegung	6
3.3 CUR-Zerlegung	6
4 Anwendung der CUR-Zerlegung	8
4.1 Beispiel: das invertierte Pendel	9
5 Nach der CUR-Zerlegung	10
6 Ausblick und Diskussion	11

1 Einführung

Eventuell noch einfügen: NNCUR, NNCX

2 Einführung und Motivation

2.1 Optimale Kontrolle

Gegeben sei das Problem der optimalen Stabilisierung eines (instabilen) diskreten Systems

$$x_{k+1} = f(x_k, u_k), \quad k = 0 \dots n \quad (1)$$

um einen Fixpunkt $0 \in X$, also $f(0, 0) = 0$. Der Fixpunkt kann o.B.d.A gewählt werden, ein anderer Fixpunkt kann gegebenenfalls in den Nullpunkt geshiftet werden. Die x_k sind aus dem Zustandsphasenraum $X \subset \mathbb{R}^n$, die u_k aus der Feedback-Menge $U \subsetneq \mathbb{R}^m$. X und U sind jeweils kompakte Teilmengen der Grundräume. Die Kosten pro Schritt sind dabei gegeben durch eine (stetige) Kostenfunktion $g : X \times U \rightarrow [0, \infty)$ mit $g(x, u) \geq 0 \ \forall x \in X, u \in U$ und $g(x, u) = 0 \Leftrightarrow x = 0, u = 0$.

Das Feedback u soll nun so gewählt werden, dass die entstehenden Kosten minimiert werden und der Fixpunkt 0 zu einem asymptotisch stabilen Fixpunkt wird. Anders ausgedrückt: ausgehend von einem Punkt $x \in X$ soll ein Kontrollvektor $u = (u_1, u_2 \dots)$ gewählt werden, sodass die Trajektorie

$$x_0(x, u) = x, \quad x_{k+1} = f(x_k(x, u), u_k), \quad k = 0, 1, \dots$$

für alle k in X verbleibt und $x_k(x, u) \rightarrow 0$ für $k \rightarrow \infty$ (vgl. (9)). Alle Punkte x , für die eine Stabilisierung möglich ist werden in der Menge

$$S = \{x \in X \mid \exists u \in U^\mathbb{N} : x_k(x, u) \rightarrow 0\}$$

zusammengefasst.

Somit ergibt sich für die gesamten Kosten zur Stabilisierung eines Punktes $x \in X$ entlang einer Trajektorie

$$J(x, u) = \sum_{k=0}^{\infty} q(x_k(x, u), u_k) \quad \in [0, \infty] \quad (2)$$

Ist x nicht stabilisierbar, d.h. es existiert kein $u \in U^\mathbb{N}$ sodass $x_k(x, u) \rightarrow 0$, so ist $J(x, u) = \infty$ und $x \in X \setminus S$.

Gesucht sind nun die minimalen Kosten zur Stabilisierung (vgl. (10)) eines Punktes x , die Kostenfunktion

$$V(x) = \inf_{u \in U^\mathbb{N}} J(x, u) \quad (3)$$

2.2 Berechnung der Wertefunktion

Im Folgenden soll die zuvor definierte optimale Wertefunktion 3 numerisch berechnet werden. Hierzu wird das *Optimalitätsprinzip nach Bellman* (vgl. (3)) zu Hilfe genommen:

$$V(x) = \inf_{u \in U^\mathbb{N}} \{q(x, u) + V(f(x, u))\} \quad (4)$$

wobei sich die optimale Steuerungssequenz ergibt aus

$$u(x) = \operatorname{argmin}_{u \in U^{\mathbb{N}}} \{q(x, u) + V(f(x, u))\} \quad (5)$$

Zur numerischen Lösung dieses Optimierungsproblems wird ein Ansatz aus der Graphentheorie gewählt:

Die (kompakte) Menge X wird partitioniert, d.h. es entsteht eine Menge \mathcal{P} von Teilmengen $P_i \subset X, i = 1, \dots, l$, wobei $\bigcup_{i=1}^l P_i = X$ und $m(P_i \cap P_j) = 0$ für $i \neq j$ (m bezeichnet hier das Lebesgue-Maß).

Auf dieser Partition wird nun ein gerichteter Graph

$$G_{\mathcal{P}} = (\mathcal{P}, E_{\mathcal{P}}), \quad E_{\mathcal{P}} = \{(P_i, P_j) \in \mathcal{P} \times \mathcal{P} \mid f(P_i, U) \cap P_j \neq \emptyset\}$$

definiert. Hierbei sind die Gewichte der Kanten $e = (P_i, P_j)$ (vgl. (10)) alle nichtnegativ und gegeben durch

$$w(e) = \min_{x \in P_i, u \in U} \{q(x, u) \mid f(x, u) \in P_j\}.$$

Bild 1 zeigt schematisch solch einen Graphen ausgehend von einem Punkt x .

Bild des Graphen

Anstatt der exakten Wertefunktion $V(x)$ (4) wird nun die approximative Wertefunktion

$$V_{\mathcal{P}}(x) = \min\{w(p(x)) \mid \text{Pfad } p(x) = (e_1, \dots, e_m), e_k \in E_{\mathcal{P}} \text{ verbindet } x \text{ mit dem Ursprung}\}$$

berechnet.

Um, ausgehend von dem bestimmten Graphen, den kürzesten Weg und damit die bestmögliche Stabilisierung bezüglich der Kosten zwischen einem Punkt x und einem Fixpunkt 0 zu bestimmen, können Standard-*shortest-path*-Algorithmen wie beispielsweise der Algorithmus von Dijkstra (vgl. (4)) verwendet werden.

Somit kann für jedes Element der Partition \mathcal{P} eine optimale Stabilisierung, d.h. bezüglich der geringsten Kosten, berechnet werden.

Konvergenz: Für alle Punkte $x \in S$, für die eine Stabilisierung möglich ist, und für den Partitionsdurchmesser $\operatorname{diam}(\mathcal{P}) := \max_i \{\operatorname{diam}(P_i)\} \rightarrow 0$ gilt:

$$V_{\mathcal{P}}(x) \rightarrow V(x)$$

Für einen Beweis siehe (10).

Schreibt man den resultierenden Boxplot in Matrixform, erhält man Folgendes:

Da diese Matrizen je nach Parameter ziemlich groß werden können, sollen sie approximiert werden. Im nächsten Kapitel werden diverse Approximationsmethoden vorgestellt.

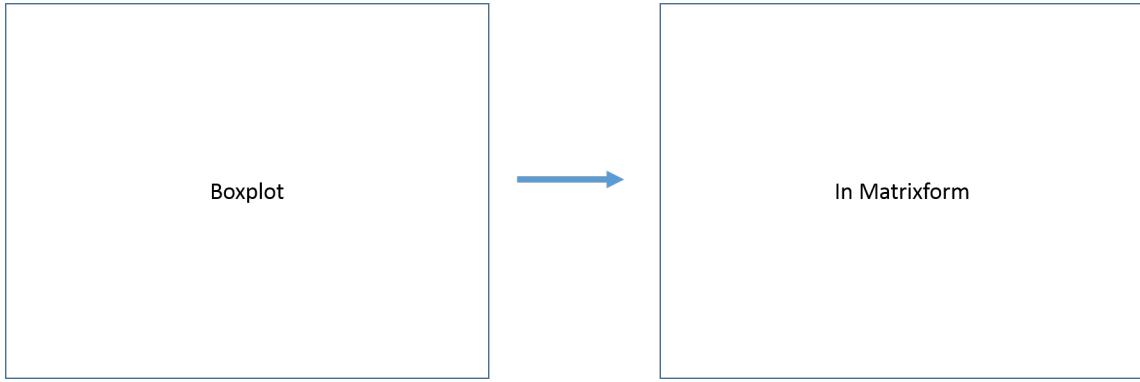


Abbildung 1: Umformung des Boxplots in Matrixform

2.3 Beispiel: das invertierte Pendel

Als Beispiel dient das Modell des invertierten Pendels auf einem Wagen. Bei diesem Modell wird ein starres invertiertes Pendel rotationsfähig auf einem Wagen montiert. Beide Körper, Pendelmasse und Wagen, werden als konzentrierte Punktmassen angenommen. Zur Balancierung des Pendels können Kräfte u auf den Wagen angewandt werden um ihn in der Horizontalen zu stabilisieren. Die Dynamik des Wagens wird in diesem Modell vernachlässigt, stattdessen konzentriert man sich auf den Auslenkungswinkel φ des Pendels von der Vertikalen und auf die Winkelgeschwindigkeit $\dot{\varphi}$ im Zustandsraum $(\varphi, \dot{\varphi}) \in \mathbb{R}^2$. Das Pendel befindet sich im höchsten Punkt in einer instabilen Ruhelage. Verlässt es diese Ruhelage um einen Winkel φ , so muss dem Pendel wieder Energie hinzugefügt werden um es wieder in den Zustand der instabilen Ruhelage zurückgeführt werden kann. Das mathematische System zu dieser Zustandsrückführung definiert sich wie folgt (vgl. (9)):

$$\dot{x}_1 = x_2 \quad (6)$$

$$\dot{x}_2 = \frac{\frac{g}{l} \sin(x_1) - \frac{1}{2} m_r x_2^2 \sin(2x_1) - \frac{m_r}{ml} \cos(x_1) u}{\frac{4}{3} - m_r \cos^2(x_1)} \quad (7)$$

wobei $x_1 = \varphi$, $x_2 = \dot{\varphi}$, $m_r = \frac{m}{m+M}$ das Massenverhältniss zwischen Pendelmasse und Wagenmasse, $m = 2\text{kg}$, $M = 8\text{kg}$, $l = 0.5\text{m}$ Länge des Pendels und $g = 9.8\frac{\text{m}}{\text{s}^2}$ die Gravitationskonstante.

Die schrittweisen Stabilisierungskosten des inversen Pendels ergeben sich aus

$$q(x, u) = \frac{1}{2}(0.1x_1^2 + 0.05x_2^2 + 0.01u^2). \quad (8)$$

Berechnet man auf Basis dieses mathematischen Modells nun die Wertefunktion V_P wie in 2.2 definiert, so ergibt sich folgendes Bild:

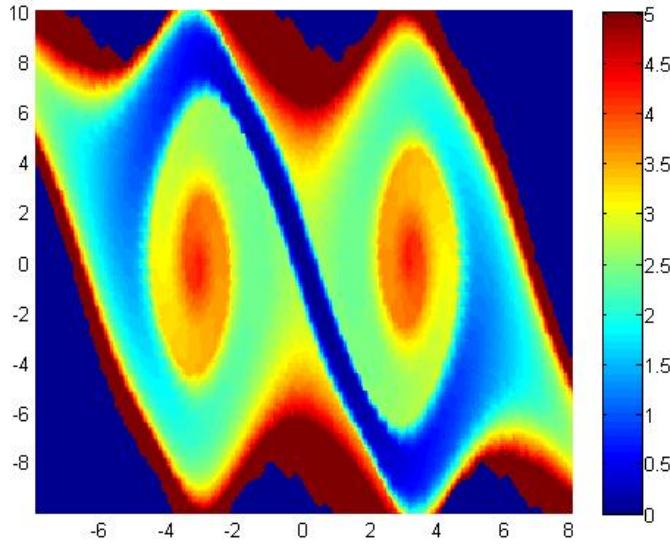


Abbildung 2: Approximierte Wertefunktion V_P bei Simulationstiefe 16

3 Niedring-Rang-Approximation einer Matrix

Je nach Tiefe der Simulation bzw. der Dimension des zugrunde liegenden Raumes (vgl. **Fluch der Dimension** nach Bellman (1)) können gigantische Matrizen entstehen. Da die gängigen Rechner bei der Speicherung solcher Matrizen schnell an ihre Grenzen stoßen, soll nun für die Wertematrix V eine Niedrig-Rang- Approximation der Form

$$V(x) \approx \sum_{i=1}^k \sigma_i u_i(x) v_i^T(x) \quad (9)$$

gefunden werden, wobei $\sigma_i \in \mathbb{R}$ Konstanten und u_i, v_i Vektoren im \mathbb{R}^n sind. Im Folgenden werden einige Näherungen vorgestellt.

3.1 Singulärwertzerlegung (SVD)

3.1.1 Definition

Die klassische Singulärwertzerlegung ist eine Erweiterung der Diagonalisierung von reellen quadratischen Matrizen auf komplexe $\mathbb{R}^{m \times n}$ -Matrizen und gerade in der numerischen Mathematik von großem Nutzen. Sie ist folgendermaßen definiert.

Definition 3.1 Sei A eine komplexe $m \times n$ -Matrix von Rang r . Die Singulärwertzerlegung von A ist gegeben durch (vgl. (3))

$$A = U \Sigma V^*$$

wobei

- U eine unitäre $m \times m$ -Matrix ist,
- V^* die Adjungierte einer $n \times n$ -Matrix V ist und

- Σ eine reelle $m \times n$ -Diagonalmatrix der Form

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & \vdots & & \\ & & & & 0 & \dots \\ & & & & \dots & 0 & \dots \\ & & & & & & \vdots \end{pmatrix}$$

Die Einträge $\sigma_i > 0$, $i = 1 \dots r$, $\sigma_1 > \dots > \sigma_r$ heißen Singulärwerte von A . Die Spalten von U , u_i , heißen linke Singulärvektoren, die Spalten von V , v_i , heißen rechte Singulärvektoren.

Bemerkung 1 A kann auch geschrieben werden als die Summe von Rang-1-Matrizen:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

Die σ_i sind dabei die Singulärwerte von A und u_i und v_i die Zeilen von U bzw. V .

Die Berechnungskosten, die für das Ausführen der Matrixzerlegung benötigt werden, betragen $\mathcal{O}(\min(mn^2, m^2n))$ (vgl. (11)).

Im Folgenden wollen wir die Wertematrix V so approximieren, dass der Fehler

$$e(D) = \|V - D\|_\xi$$

bezüglich einer Norm ξ möglichst klein wird für Matrizen D von Rang $\text{rank}(D) \leq k$. $\xi = 2$ bezeichnet hierbei die 2-Norm ($\|A\|_2 = \left(\sum_{i,j} |a_{ij}|^2\right)^{\frac{1}{2}}$), $\xi = \max$ bezeichnet die Maximumsnorm ($\|A\|_{\max} = \max_{i,j} |a_{ij}|$) und $\xi = F$ verwendet die Frobeniusnorm ($\|A\|_F = (\sigma_1^2 + \dots + \sigma_r^2)^{\frac{1}{2}}$).

3.1.2 Fehlerapproximation

Die erste Frage, die auftaucht, ist die nach der Existenz einer "besten" Rang-k-Approximation.

Satz 3.2 (Eckart-Young) Sei eine Matrix V von Rang r gegeben durch 3.1 und $k < r$. Die beste Rang-k-Approximation von V , also eine Matrix, die

$$\min_{\text{rank}(B) \leq k} \|V - B\|_2$$

erfüllt, ist gegeben durch

$$V_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

und

$$\|V - V_k\|_2 = \sigma_{k+1}$$

wobei σ_{k+1} der $(k+1)$ -te Singulärwert von V ist. Dies impliziert, dass für Singulärwertzerlegungen von Rang $k \geq r$ gilt: $\|V - V_k\|_2 = 0$.

Beweis 1 Für einen Beweis siehe (8).

Die Kosten der Rang-k-Singulärwertzerlegung betragen $\mathcal{O}(k(m + n))$ (6).

Nachteil: Es werden keine konkreten Einträge der ursprünglichen Matrix V verwendet, weswegen die Singulärwertzerlegung ohne Erweiterungen für unsere Zwecke nicht verwendbar ist.

In den nächsten beiden Abschnitten werden zwei Verfahren vorgestellt, welche Spalten und/oder Zeilen der Eingabematrix verwenden.

3.2 CX-Zerlegung

Die erste approximative Matrix-Zerlegung, welche konkrete Einträge der ursprünglichen Matrix verwendet, ist die CX-Zerlegung (vgl. (7)).

Definition 3.3 Sei A eine $m \times n$ -Matrix. Sei $C \in \mathbb{R}^{m \times c}$ eine Matrix bestehend aus c Spalten der Matrix A (in der Regel ist $c \ll n$). Dann ist die approximative Matrix $A' = CX$ für jede beliebige Matrix $X \in \mathbb{R}^{c \times n}$ eine spaltenbasierte Approximation von A , beziehungsweise eine CX-Zerlegung von A .

3.3 CUR-Zerlegung

Bei der CUR-Zerlegung werden im Gegensatz zur CX-Zerlegung Spalten und Zeilen der zu approximierenden Matrix verwendet. Die Matrix C enthält dabei Spalten von V und Zeilen werden in R gefasst. Die Matrix U wird dann so gewählt, dass der Fehler zwischen V und CUR bezüglich einer Norm minimal wird.

Die mathematische Definition der CUR-Zerlegung ist wie folgt:

Definition 3.4 (vgl. (7)) Gegeben sei eine Matrix $V \in \mathbb{R}^{m \times n}$. Sei $C \in \mathbb{R}^{m \times c}$ eine Matrix, welche aus c Spalten von V besteht, und $R \in \mathbb{R}^{r \times n}$ eine Matrix, deren r Zeilen aus Zeilen von V bestehen. Dann ist die Matrix $V' = CUR$ für jede beliebige Matrix $U \in \mathbb{R}^{c \times r}$ eine Spalten-Zeilen-basierte Approximation von V , beziehungsweise eine CUR-Zerlegung von V .

Im Folgenden wird ein Verfahren zur Berechnung der CUR-Zerlegung nach Mahoney und Drineas (11) vorgestellt.

Besonders kritisch bei der Berechnung der Zerlegung ist, welche Zeilen bzw. Spalten für die Approximation verwendet werden. Dafür wird ein Rating erzeugt (siehe (11)), welche den Einfluss einer Spalte/Zeile auf die restlichen Einträge der Matrix beschreibt.

Bestimmung des statistischen Einflusswertes einer Spalte/Zeile:

Aus 1 wissen wir, dass sich die j -te Spalte einer Zielmatrix V von Rang r darstellen lassen kann als

$$V^j = \sum_{\xi=1}^r (\sigma_\xi u^\xi) v_j^\xi$$

bzw. deren "beste" Rang-k-Approximation durch

$$V^j = \sum_{\xi=1}^k (\sigma_\xi u^\xi) v_j^\xi$$

wobei u^ξ der ξ -te linke Singulärvektor und v_j^ξ die j -te Koordinate des ξ -ten rechten Singulärvektors ist. Der **statistische Einflusswert** einer Spalte von V kann nun gemessen werden durch

$$\pi_j = \frac{1}{k} \sum_{\xi=1}^k (v_j^\xi)^2 \quad j \in [1, n] \quad (10)$$

Er gibt an, wie groß der Einfluss einer Spalte auf die durch die Singulärwertzerlegung definierte "beste" Rang-k-Approximation V_k ist.

Diese Einflusswerte bilden eine Wahrscheinlichkeitsverteilung auf den n Spalten und es gilt: $\sum_{j=1}^n \pi_j = 1$ und $\pi_j \geq 0$.

Das Selektieren der Spalten wird anschließend mithilfe des Algorithmus' COLUMNSELECT (vgl. (11)) durchgeführt:

- (i) **Input:** Matrix $V \in \mathbb{R}^{m \times n}$ und Fehlerparameter ϵ
- (ii) Berechne die ersten k rechten Singulärvektoren (v_1, \dots, v_k) von V und die statistischen Einflusswerte 10 der Spalten von V .
- (iii) Verwende die j -te Spalte von V mit Wahrscheinlichkeit $p_j = \min\{1, c\pi_j\}$, wobei $c = \mathcal{O}(k \log \frac{k}{\epsilon^2})$
- (iv) **Output:** Matrix C , welche aus den selektierten Spalten von A besteht.

Mithilfe dieses Algorithmus' entsteht eine Matrix C , die $c' \leq c$ der selektierten Spalten verwendet. Die Kosten zur Ausführung des Algorithmus' beruhen hauptsächlich auf der Berechnung der ersten k Singulärvektoren und sind daher $\mathcal{O}(mnk)$ (11).

Zur Berechnung der Vektoren C, U und R wird abschließend folgender Algorithmus ALGORITHMCUR (vgl (11)) verwendet:

- (i) Wende COLUMNSELECT auf A an mit $c = \mathcal{O}(k \log \frac{k}{\epsilon^2})$ und berechne somit die Matrix C
- (ii) Wende COLUMNSELECT auf A^T an mit $r = \mathcal{O}(k \log \frac{k}{\epsilon^2})$ und berechne somit die Matrix R
- (iii) Definiere die Matrix U als $U = C^+ A R^+$, wobei X^+ die Moore-Penrose-Inverse (12) von X ist

Die Wahrscheinlichkeit für die Auswahl einer der c Zeilen bzw. r Reihen erfolgt dabei proportional zur Einflussrate der entsprechenden Matrixeinträge.

Die mithilfe des Algorithmus' konstruierte approximative Matrix $V' = CUR$ erfüllt nun, dass

$$\|V - V'\|_F \leq (2 + \epsilon)\|V - V_k\|_F$$

mit hoher Wahrscheinlichkeit (m.h.W.) (vgl. (11)).

Die Rate $(2 + \epsilon)$ kann mit einer Variation des Algorithmus' noch weiter verbessert werden (7). Bei dieser Methode wird die Matrix R abhängig von den für die Matrix C gewählten Zeilen berechnet. Durch diese Abwandlung verringert sich der Fehler m.h.W. zu

$$\|V - CUR\|_F \leq (1 + \epsilon)\|V - V_k\|_F$$

Dies ist nur eine Möglichkeit eine approximative CUR -Zerlegung der Matrix V zu finden. Weitere werden in folgender Tabelle gezeigt:

	Anzahl Spalten c	Anzahl Reihen r	rank(U)	$\ V - CUR\ _F^2 \leq$
Drineas et al. 2003 (5)	$\frac{k}{\epsilon^2}$	$\frac{k}{\epsilon^2}$	k	$\ V - V_k\ _F^2 + \epsilon\ A\ _F^2$
Drineas et al. 2008 (7)	$\frac{k \log k}{\epsilon^2}$	$\frac{k \log k}{\epsilon^4}$	$\frac{k \log k}{\epsilon^2}$	$(1 + \epsilon)\ V - V_k\ _F^2$
Wang et al. 2013 (13)	$\frac{k}{\epsilon}$	$\frac{k}{\epsilon^2}$	$\frac{k}{\epsilon}$	$(1 + \epsilon)\ V - V_k\ _F^2$

Abbildung 3: Tabelle über verschiedene CUR-Approximationen

Vorteil: Zeilen und Spalten der ursprünglichen Matrix werden verwendet

Nachteil: Die approximierende Matrix ist nicht mehr von der Form 9.

4 Anwendung der CUR-Zerlegung

Die numerische Berechnung der approximativen Matrix $C' = CUR$ wird in R^1 durchgeführt. Dafür wird ein auf Drineas und Mahoney (11) basierender Algorithmus (2) verwendet.

Die ursprünglich von Drineas et al. (11) veröffentlichte Methode verwendet eine Auswahl an Zeilen und Spalten der Zielmatrix entsprechend der stochastischen Einflussrate. Dies impliziert einige numerische Probleme, da die resultierenden Fehler zwischen Zielmatrix und CUR -Zerlegung je nach ausgewählten Zeilen/ Spalten sehr stark variieren. Aus diesem Grund wurden Berechnungen mit unterschiedlichen Variationen dieser Methode durchgeführt.

Die besten Ergebnisse wurden mit der Methode *ortho.top.scores* erreicht. Bei dieser Variation des Algorithmus' werden die Spalten und Zeilen aufgrund der Linearkombination

¹R Version 3.1.1, Script ausgeführt mithilfe RStudio Version 0.98.1017

aus Einflusswert und "maximaler Orthogonalität" der Spalten bzw. Zeilen zueinander ausgewählt.

4.1 Beispiel: das invertierte Pendel

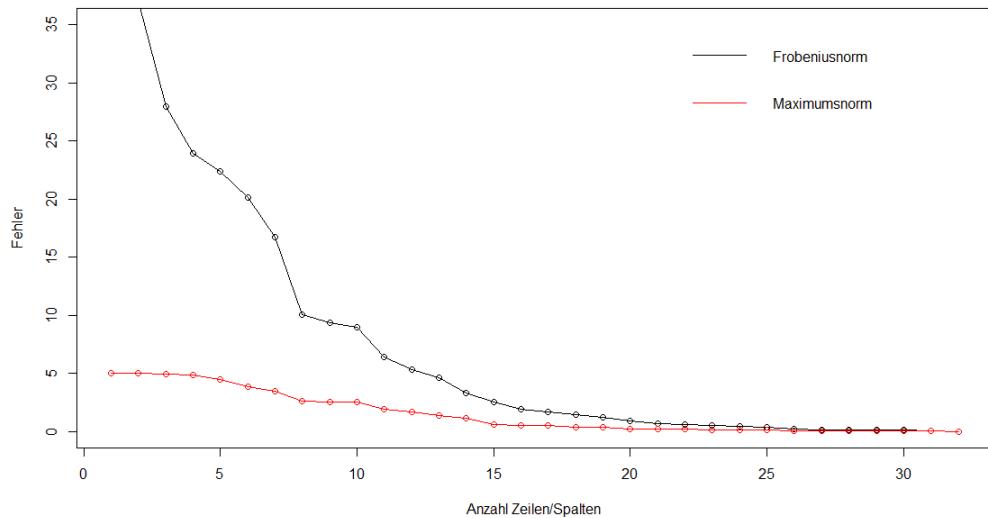


Abbildung 4: Plot der resultierenden Fehler bei einer 32x32-Matrix (schwarz) Frobeniusnorm (rot) Maximumsnorm

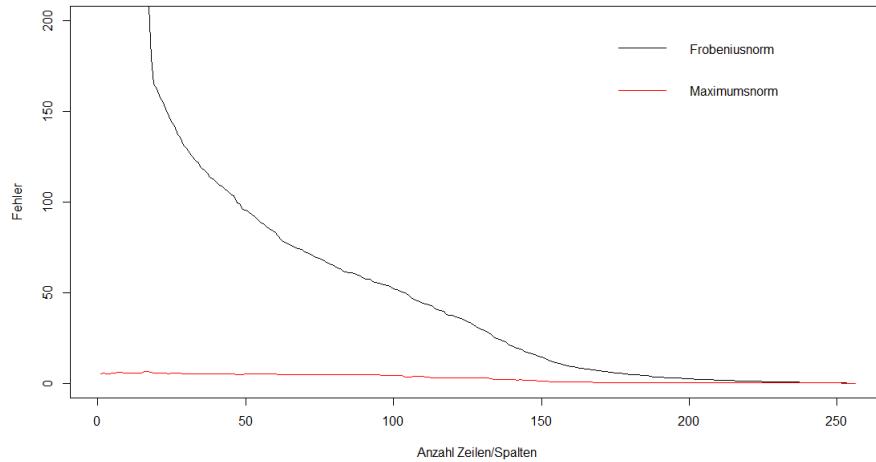


Abbildung 5: Plot der resultierenden Fehler bei einer 256x256-Matrix (schwarz) Frobeniusnorm (rot) Maximumsnorm

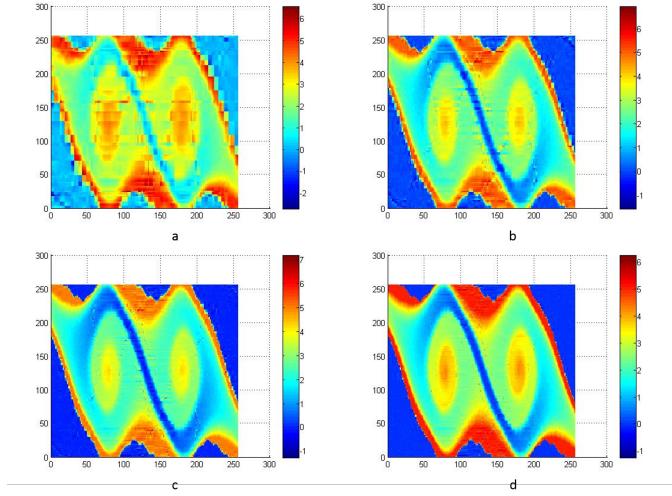


Abbildung 6: Plot der Wertefunktion bei einer Auswahl von 20 (a) 50 (b) 80 (c) und 120 (d) Spalten und Zeilen bei einer 256x256-Matrix

Für die numerische Betrachtung ist es natürlich von unbestreitbarem Interesse zu wissen, wo die größten Fehler vorkommen. Im Falle, dass die größten Fehler vor allem am Rand des stabilisierbaren Bereiches auftreten und im Inneren der Matrix gilt, dass $V_{CUR} \approx V_P$, so können die Fehler am Rand vernachlässigt und eine ungenauere Approximation verwendet werden.

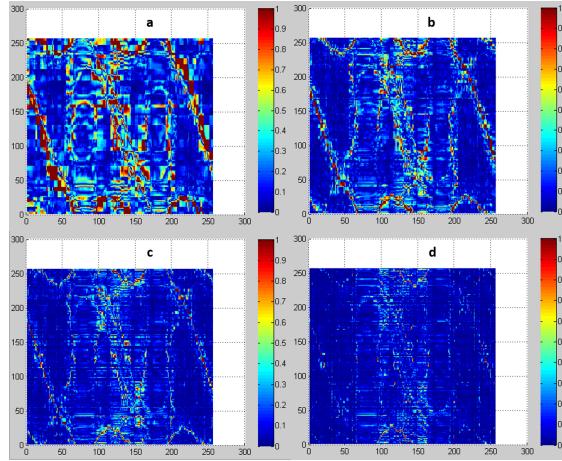


Abbildung 7: Lokalisierung der Stellen der großen Fehler der Wertefunktion bei einer Auswahl von 20 (a) 50 (b) 80 (c) und 120 (d) Spalten und Zeilen bei einer 256x256-Matrix. Zur besseren Übersichtlichkeit wurde die angezeigte Skala auf 0-1 eingeschränkt

5 Nach der CUR-Zerlegung

Für die exakte Wertefunktion V gilt nach Bellmann, dass

$$\begin{aligned} V(x_k) &= \inf_{u \in U^{\mathbb{N}}} \{q(x_k, u) + V(f(x_k, u))\} \\ &= q(x_k, u_k^*) + V(f(x_k, u_k^*)) \end{aligned}$$

mit dem minimierenden u_k^* , falls das Optimum angenommen wird, wobei $V(f(x_k, u)) =: V(x_{k+1})$. Da $q(x, u) > 0$ für $x \neq 0$ gilt also insbesondere, dass $V(x_k) > V(x_{k+1})$.

Mit dieser Eigenschaft wird die Wertefunktion zu einer **Lyapunov-Funktion** (siehe (14)), da gilt:

- (i) $V(x) \geq 0$
- (ii) $V(x) = 0 \Leftrightarrow x = 0$
- (iii) $V(x_k) > V(x_{k+1})$

Leider ergibt sich bei der approximativen CUR-Zerlegung \tilde{V} das Problem, dass die Bedingung (iii) nicht mehr notwendigerweise erfüllt ist.

Für den resultieren Fehler berechnet man dazu das **Residuum** der Bellman-Gleichung. Dieses ist gegeben durch

$$R(x) = \tilde{V}(x) - \inf_u \{q(x, u) + \tilde{V}(f(x, u))\}. \quad (11)$$

mit

$$\tilde{u}(x) = \operatorname{argmin}_{u \in U^N} \{q(x, u) + \tilde{V}(f(x, u))\} \quad (12)$$

Im Falle des Erfüllens der Bedingung (iii) ergibt sich

$$R(x) \geq 0 \quad \forall x \in X \quad (13)$$

da $q(x, u) \geq 0$ für alle $x \in X$.

Typischerweise ist diese Bedingung aber nicht für alle $x \in X$ erfüllt. Als alternative Formulierung soll deshalb die Menge $\bar{S} = \{x \in X | R(x) < 0\}$ bezüglich des Lebesgue-Maßes möglichst geringes Volumen besitzen.

6 Ausblick und Diskussion

Literatur

- [1] BELLMAN, Richard E.: *Adaptive Control Processes*. Princeton : Princeton University Press, 1961
- [2] BODOR, Andras ; SOLYMOSI, Norbert: *rCUR: an R package for CUR matrix decomposition*. 2012
- [3] DEUFLHARD, Peter ; HOHMANN, Andreas: *Numerische Mathematik 1: Eine algorithmisch orientierte Einführung*. 4. überarb. u. erw. Aufl. Berlin u.a. : de Gruyter, 2008 (de Gruyter Lehrbuch). <http://www.reference-global.com/action/showBook?doi=10.1515/9783110203554>. – ISBN 9783110203554
- [4] DIJKSTRA, E.W.: *A Note on Two Problems in Connection with Graphs*. 1959
- [5] DRINEAS, Petros ; KANNAN, R.: Pass Efficient Algorithms for Approximating Large Matrices. In: *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2003), S. 223–232
- [6] DRINEAS, Petros ; KANNAN, Ravi ; MAHONEY, Michael W.: Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. In: *Society for Industrial and Applied Mathematics (SIAM)* (2006), S. 184–206
- [7] DRINEAS, Petros ; MAHONEY, Michael W. ; MUTHUKRISHNAN, S.: Relative-Error CUR Matrix Decompositions. In: *SIAM journal on matrix analysis and applications* 30 (2009), Nr. 3, S. 844–881
- [8] GOLUB, Gene H. ; VAN LOAN, Charles F.: *Matrix computations*. 4. ed. Baltimore : Johns Hopkins Univ Press, 2013 (Johns Hopkins studies in the mathematical sciences). – ISBN 9781421408590
- [9] GRÜNE, Lars ; JUNGE, Oliver: A set oriented approach to optimal feedback stabilization. In: *Systems & control letters* 54 (2005), Nr. 2, S. 169
- [10] JUNGE, O. ; OSINGA, H. M.: A set oriented approach to global optimal control. In: *Control, optimisation and calculus of variations : COCV* 10 (2004) (2004), S. 259–270
- [11] MAHONEY, Michael W. ; DRINEAS, Petros: CUR matrix decompositions for improved data analysis. In: *PNAS* 2009 (2008), Nr. vol. 106, S. 697–702
- [12] PENROSE, Roger: A generalized inverse for matrices. In: *Proceedings of the Cambridge Philosophical Society* 51 (1955), S. 406–4013
- [13] WANG, Shusen ; ZANG, Zhihua: Improving CUR Matrix Decomposition and the Nyström Approximation via Adaptive Sampling. In: *Journal of Machine Learning Research* 2013 (2013), Nr. 14, S. 2729–2769
- [14] WIGGINS, Stephen: *Introduction to applied nonlinear dynamical systems and chaos*. New York : Springer-Verlag, 1990