

Übungen zu
POD für linear-quadratische Optimalsteuerung

<http://www.math.uni-konstanz.de/numerik/personen/beermann/teaching>

Blatt 2

Abgabe: Dienstag, 21. November in der Vorlesung

Aufgabe 2.1 (Theorie)

(5 Punkte)

Seien $m, n \in \mathbb{N}$ beliebig. Die Frobeniusnorm $\|\cdot\|_F$ auf dem Vektorraum der reellen $(m \times n)$ -Matrizen ist gegeben durch

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

für alle $A \in \mathbb{R}^{m \times n}$.

Seien $Y \in \mathbb{R}^{m \times n}$ eine Rang- d Matrix, $\ell \leq d$ und $\Psi = (\psi_1, \dots, \psi_\ell)$ eine POD-Basis zu Y bzgl. des Standard-Skalarproduktes des \mathbb{R}^m .

Zeigen Sie, dass die POD-Basis eine optimale Approximation bzgl. der Frobeniusnorm liefert, d.h. dass für jedes beliebige Orthonormalsystem $\Phi = (\phi_1, \dots, \phi_\ell)$ des \mathbb{R}^m gilt

$$\|Y - \Psi(\Psi^T Y)\|_F \leq \|Y - \Phi(\Phi^T Y)\|_F.$$

Aufgabe 2.2 (Programmierteil)

(10 Punkte)

Gegeben seien das Ortsintervall $\Omega \subset \mathbb{R}$, das Zeitintervall $\Theta \subset \mathbb{R}$ sowie die Parametermenge $U \subset \mathbb{R}$. Nach Diskretisierung des Orts- und des Zeitintervalls in n_x , bzw. n_t Punkte verstehen wir eine parameterabhängige Funktion $y : \Omega \times \Theta \times U \rightarrow \mathbb{R}$ für jedes $u \in U$ als Matrix $Y(u) = [y^1(u), \dots, y^{n_t}(u)] \in \mathbb{R}^{n_x \times n_t}$. Eine POD-Basis der Matrix $Y(u)$ vom Rang ℓ wird mit $\Psi(u) = (\psi_1(u), \dots, \psi_\ell(u)) \in \mathbb{R}^{n_x \times \ell}$ bezeichnet, um die Abhängigkeit vom Parameter u zu verdeutlichen. Ein wiederkehrendes Thema in Optimaler Steuerung ist es, eine solche POD-Basis an einem konkreten Parameter zu erstellen und davon auszugehen, dass diese Basis auch für andere Parameter gute Approximationen liefert. In dieser Aufgabe soll der dadurch gemachte Fehler untersucht werden.

Sei dafür die Funktion

$$y(x, t, u) := \cos(u(t + x)), \quad \Omega = (0, 2\pi), \quad \Theta = (0, 2\pi), \quad U = [0, 2]$$

gegeben.

1. Erstellung einer POD-Basis:

Diskretisieren Sie das Ortsintervall Ω und das Zeitintervall Θ in $n_x = 200$ bzw. $n_t = 100$ Punkte. Verwenden Sie die in Aufgabe 1.1 erstellte Funktion zur Berechnung einer POD-Basis $\Psi(1)$ der Matrix $Y(1)$ vom Rang $\ell = 2$.

2. Fehlerbetrachtung bei variierendem Parameter:

Diskretisieren Sie das Intervall U in $n_u = 20$ Punkte und berechnen Sie jeweils die POD-Approximationen von $y(\cdot, \cdot, u_i)$ ($i = 1, \dots, n_u$) bzgl. der POD-Basis $\Psi(1)$.

- Betrachten Sie grafische Veranschaulichungen der POD-Approximationen. Was beobachten Sie bei der POD-Approximation von $y(\cdot, \cdot, 2)$? Erklären Sie das Verhalten mathematisch.
- Stellen Sie den Approximationsfehler in Abhängigkeit des Parameters u in einem Diagramm dar.

3. Intelligentes Clustering durch u -lokale Basen:

Diskretisieren Sie das Intervall U in n_u Punkte. Überlegen Sie sich und implementieren Sie einen klugen Algorithmus, der Punkte $u_{p_1}, \dots, u_{p_k} \in U$ und dazugehörige paarweise disjunkte Teilmengen $U_{p_1}, \dots, U_{p_k} \subset U$ findet, so dass folgende Eigenschaften erfüllt sind:

- (a) Es gilt $u_{p_i} \in U_{p_i}$ für alle $i = 1, \dots, k$.
- (b) $U = U_{p_1} \cup \dots \cup U_{p_k}$.
- (c) Für jede Teilmenge U_{p_i} gilt: Für alle $u \in U_{p_i}$ ist der Approximationsfehler der POD-Approximation der Funktion $y(\cdot, \cdot, u)$ bzgl. der POD-Basis $\Psi(u_{p_i})$ vom Rang $\ell = 2$ kleiner als eine vorgegebene Fehlerschranke ε .

Testen Sie den Algorithmus für $n_u = 20, 40$ und $\varepsilon = 2, 3$.

Aufgabe 2.3 (Programmierteil)

(5 Punkte)

Sie wissen aus der Vorlesung, dass der Wert

$$\mathcal{E}(\ell) := \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^d \lambda_i} \in [0, 1]$$

- auch *Energie* genannt - einen Indikator für die in einer POD-Basis enthaltene relative Informationsmenge enthält. Erweitern Sie Ihren Code aus Aufgabe 1.1., der bisher noch eine POD-Basis mit fester Länge $\ell \in \mathbb{N}$ generiert, um die Option, diese Länge über die Energie zu bestimmen: Unter Vorgabe einer Schranke $\varepsilon \in [0, 1)$ soll diese Variante das kleinste $\ell \in \mathbb{N}$ wählen, sodass $\mathcal{E}(\ell) \geq 1 - \varepsilon$ gilt.

Testen Sie Ihren Code mit den Funktionen y_1, y_2 und y_3 aus Aufgabe 1.1. Tragen Sie für diese außerdem die Eigenvektoren $\lambda_1, \dots, \lambda_{20}$ in einem logarithmischen Plot auf. Erklären Sie Ihre Beobachtungen.