

WESTFÄLISCHE WILHELMS-UNIVERSITÄT
MÜNSTER

**Untersuchung von
Modellreduktionsmethoden für
Parameter-abhängige dynamische
Systeme**

BACHELORARBEIT MATHEMATIK

Christopher Wiedey

betreut durch
Prof. Dr. Mario OHLBERGER
Dr. Felix Tobias SCHINDLER

11. Juni 2019

Plagiatserklärung des Studierenden

Hiermit versichere ich, dass die vorliegende Arbeit über *Untersuchungen von Modellreduktionsmethoden für Parameter-abhängige dynamische Systeme* selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

(Datum, Unterschrift)

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

(Datum, Unterschrift)

Inhaltsverzeichnis

1 Einleitung	3
2 Mathematische Grundlagen	4
2.1 Parametrisiertes dynamisches System	4
2.2 Projektion des dynamischen Systems	5
2.3 A-posteriori Fehlerabschätzung	11
3 Greedy-Algorithmen zur Konstruktion reduzierter Räume	14
4 Implementation des Verfahrens	18
4.1 Zeitintegration des dynamischen Systems	18
4.2 Zeitintegration der Fehlerabschätzung	21
4.3 Effiziente Lösungsverfahren für das reduzierte System	23
5 Numerische Experimente	29
5.1 Untersuchung der Problemstellung:	30
5.2 Deterministischer Fall	33
5.3 Allgemeiner Fall	35
5.4 Untersuchung der Fehlerabschätzung	38
5.5 CPU-Zeiten	39
5.6 Speicherbedarf	40
6 Fazit	54

1 Einleitung

Das numerische Lösen von Anfangswertproblemen ist ein wesentliches Mittel zur computergestützten Simulation dynamischer Systeme. Um eine der jeweiligen Anwendung angemessene Präzision sicherzustellen, benutzen die durchgeführten Verfahren äußerst hoch-dimensionale Matrizen. Im Kontext zeitabhängiger Probleme kann zudem oft nicht auf iterative Algorithmen verzichtet werden. Der Zeit- und Rechenaufwand solcher Simulationen ist daher im Allgemeinen recht hoch. Die Lage verkompliziert sich weiter, wenn wir parametrisierte Systeme betrachten. Der naive Ansatz, das hoch-dimensionale Problem für jeden gewünschten Parameter einzeln auszuwerten, ist in seiner Komplexität proportional zur Anzahl der Parameter. Eine effektive Methode, solche parametrisierten zeitabhängigen dynamischen Systeme approximativ zu berechnen, sind *reduzierte-Basis-Methoden*.

Ziel dieser Arbeit ist die Untersuchung des im Paper *Dynamical Model Reduction Method for Solving Parameter-Dependent Dynamical Systems*[BFN17] aus dem Jahre 2017 vorgestellten Verfahrens zur Reduktion parametrisierter zeitabhängiger dynamischer Systeme. Das von den Autoren benannte *T-Greedy-Verfahren* ist eine Methode zur Konstruktion reduzierter Basen. Um nicht für jeden Parameter das hoch-dimensionale Problem teuer auszuwerten, wird dessen Lösungsmannigfaltigkeit durch eine Folge zeitabhängiger Untervektorräume angenähert. Die Basen der Untervektorräume werden durch Auswertungen des unreduzierten Problems zu bestimmten Parametern mithilfe der Galerkin-Projektion konstruiert. Die Auswahl dieser Parameter geschieht durch ein spezifisches Greedy-Schema mit einer gegebenen a-posteriori Abschätzung des Approximationsfehlers. Die Berechnung der Lösung des reduzierten Systems erfolgt dann über eine Offline/Online-Zerlegung der dazu erforderlichen Berechnungen. Dafür werden möglichst alle im iterativen Zeitschrittverfahren benötigten Parameter-unabhängigen Größen vorher einmalig bestimmt (Offline-Phase), um später die Berechnungen für beliebige Parameter zeiteffizient durchführen zu können (Online-Phase).

Damit wir Aussagen über den T-Greedy-Algorithmus treffen zu können, möchten wir die mathematischen Ausführungen und numerischen Experimente aus [BFN17] nachvollziehen. Dazu gehen wir wie folgt vor:

Im ersten Teil der Arbeit legen wir die abstrakten mathematischen Grundlagen des Verfahrens dar. Wir definieren die dem Verfahren zugrunde liegende Struktur des *dynamischen Systems* anhand von *Anfangswertproblemen* sowie die Reduktion dieser auf eine niedrige Dimension mittels *Galerkin-Projektion*. Hieraus leiten wir einige, sich im Verlauf der Arbeit als hilfreich erweisende Aussagen über diese Konstruktionen her. Als Voraussetzung für die Durchführbarkeit des Algorithmus präsentieren wir Sätze zur *Existenz und Eindeutigkeit von Lösungen der Systeme*.

Der zweite Teil behandelt die in [BFN17] verwendete *a-posteriori Schätzung* für den Fehler der niedrig-dimensionalen Approximation. Wir stellen die dort geschilderte Herleitung einer oberen Schranke für die Abweichung der Näherungslösung dar. Die dabei eine zentrale Rolle spielende *lokale logarithmische Lipschitzkonstante* wird definiert und Wege, sie für lineare wie auch nichtlineare Operatoren konkret zu bestimmen, werden bewiesen.

Im dritten Abschnitt widmen wir uns den beiden *Greedy-Schemen* aus [BFN17, 3], *T-Greedy* und *POD-Greedy*. Wir erklären die Funktionsweise beider Verfahren, vergleichen sie miteinander und gehen dabei insbesondere auf Komplexität und Speicherbedarf ein.

Im vierten Teil untersuchen wir zunächst die in [BFN17, 4.1] eingeführten iterativen Verfahren zur Lösung der reduzierten Anfangswertprobleme. Wir erklären die Verfahren vor dem Hintergrund der Konstruktionen aus den vorigen Abschnitten und versuchen die Richtigkeit der Offline/Online-Zerlegung aus [BFN17, 4.2] - insbesondere der Berechnung des Residuums für die Fehlerschätzung - rigoros zu beweisen.

In Teil fünf erläutern wir zuerst den Aufbau des Experiments aus [BFN17, 5.1]. Insbesondere beleuchten wir das Schema zur Diskretisierung des gegebenen dynamischen Systems vor dem Hintergrund der Stabilität verwendeter Zeitschrittverfahren und Existenz auf diese Weise erhaltener Lösungen. Danach geben wir die Ergebnisse der Berechnungen durch unsere Implementation der beiden Verfahren, POD-Greedy und T-Greedy, wieder. Wir vergleichen diese Ergebnisse mit denen aus [BFN17] und präsentieren weitere gesammelte Daten zum Ablauf der Greedy-Verfahren, zur Dauer der Berechnungen und zum Speicheraufwand der produzierten Objekte.

Im letzten Teil ziehen wir ein Fazit über die in dieser Arbeit erhaltenen Ergebnisse und vergleichen diese mit den Aussagen aus [BFN17].

2 Mathematische Grundlagen

2.1 Parametrisiertes dynamisches System

Motiviert durch die Definition [BFN17, (1)] möchten wir ein parametrisiertes dynamisches System ebenso als System von Anfangswertproblemen mit einem endlich-dimensionalen reellen Vektorraum als Zustandsraum betrachten.

Definition 2.1.1 (Dynamisches System). Es sei $I = (0, T) \subset \mathbb{R}$ ein Intervall mit $T \in \mathbb{R}_{\geq 0}$, $\Xi \subset \mathbb{R}^s$ für $s \in \mathbb{N}$ die Menge möglicher Parameter und $X = \mathbb{R}^d$ der Zustandsraum mit Dimension $d \in \mathbb{N}$. Wir wollen fortan X als Hilbertraum mit kanonischem Skalarprodukt $\langle x, y \rangle_X$ und induzierter Norm $\|x\|_X = \sqrt{\langle x, x \rangle_X}$ für alle $x, y \in X$ betrachten. Weiterhin sei $f: X \times I \times \Xi \rightarrow X$ die Übergangsfunktion, auch Fluss genannt, und $u^0: \Xi \rightarrow \mathbb{R}^d$ die vektorwertige Abbildung für die allgemein Parameter-abhängigen Anfangswerte.

(i) Zusammen mit einem System aus d Anfangswertproblemen

$$\begin{cases} u'(t, \xi) &= f(u(t, \xi), t, \xi), \\ u(0, \xi) &= u^0(\xi) \end{cases} \quad (1)$$

nennen wir (I, Ξ, X, f, u^0) ein *Parameter-abhängiges dynamisches System*.

(ii) Wir nennen $u(t, \xi) \in X$ *Lösung des dynamischen Systems* für $t \in I$ und $\xi \in \Xi$, falls $u(t, \xi)$ das Anfangswertproblem

$$\begin{cases} u'(t, \xi) &= f(u(t, \xi), t, \xi), \\ u(0, \xi) &= u^0(\xi) \end{cases}$$

für t und ξ löst.

Bemerkung 2.1.2. Die folgende Existenz und Eindeutigkeitsaussage für Systeme aus Anfangswertproblemen ist eine leicht modifizierte, inhaltlich äquivalente Version des Satzes [Wal00, §10.VI] und auch als *Satz von Picard-Lindelöf* bekannt. Die Definition der Lipschitzbedingung wurde dabei aus [Wal00, §10.IV] übernommen. Ein Beweis dieser Aussage findet sich ebenfalls dort.

Satz 2.1.3 (Existenz- und Eindeutigkeit von Systemen aus Anfangswertproblemen). *Die Funktion $f \in C(I \times X)$ genüge in $I \times X$ einer lokalen Lipschitzbedingung. Für alle $u \in X$ existiere also eine Umgebung $D' \subset I \times X$ mit $(t, u) \in D'$ für ein $t \in I$ und eine Konstante $L > 0$, sodass für alle $(t, u') \in D'$ gilt:*

$$\|f(t, u) - f(t, u')\|_X \leq L \|u - u'\|_X \quad (2)$$

Dann hat, sofern $(0, y^0) \in I \times X$, das System aus Anfangswertproblemen

$$\begin{cases} y'(t) = f(t, y) \\ y(0) = y^0 \end{cases} \quad (3)$$

genau eine Lösung y , die sich bis zum Rand von $I \times X$ fortsetzen lässt.

Beweis. Siehe [Wal00, §10.VII]. □

Bemerkung 2.1.4. Die folgende Aussage ist der Transfer der Existenz- und Eindeutigkeitsaussage 2.1.3 für Systeme von Anfangswertproblemen auf die Definition 2.1.1 des dynamischen Systems und dessen Lösungen.

Korollar 2.1.5 (Existenz und Eindeutigkeit der Lösungen des dynamischen Systems). *Es sei (I, Ξ, X, f, u^0) ein dynamisches System wie in Definition 2.1.1. Der Fluss f erfülle die Bedingungen an den Satz 2.1.3, er sei also für ein festes $\xi \in \Xi$ stetig in $I \times X$ und erfülle die lokale Lipschitzbedingung (2). Dann besitzt das dynamische System (I, Ξ, X, f, u^0) für ξ und alle $t \in I$ genau eine Lösung $u(t, \xi) \in X$.*

Beweis. Nach Definition der Lösung des dynamischen Systems in 2.1.1 als Lösung von Systemen parametrisierter Anfangswertprobleme existiert genau dann eine eindeutige Lösung von (I, Ξ, X, f, u^0) für alle $t \in I$ und Parameter ξ , falls die Bedingungen an den Satz 2.1.3 erfüllt sind. Dies folgt direkt aus den Voraussetzungen dieses Satzes. □

2.2 Projektion des dynamischen Systems

Wir wollen hier das in [BFN17, 2.1] vorgestellte Verfahren zur Reduktion des hoch-dimensionalen dynamischen System erläutern. Hierzu wird dessen Zustandsraum $X = \mathbb{R}^d$ auf Unterräume $X_r \subseteq X$ projiziert, sodass der Approximationsfehler orthogonal zur Approximation steht.¹ Eine solches Vorgehen wird auch *Galerkin-Approximation* genannt.²

Bemerkung 2.2.1. Die folgende Definition der Projektionsmatrizen stammt aus [BFN17, 2.1.]. Die aus den Konstruktionen folgenden Eigenschaften werden dort zwar genannt, jedoch nicht formal gezeigt. Dies tun wir mit den Sätzen 2.2.3 und 2.2.5. Zu einem gegebenen Unterraum X_r von X wollen wir von nun an voraussetzen, dass $\dim X_r = r$ gilt.

¹Diese Eigenschaft charakterisiert die Galerkin-Methode. Vgl. [Qua15, (3.25)].

²Für eine explizite, tiefergehende Betrachtung dieses Verfahrens, siehe z.B. [Tho06], [ABCM02].

Definition 2.2.2 (Projektionen). Sei $(X, \langle \cdot, \cdot \rangle_X)$ ein reeller Hilbertraum mit kanonischem Skalarprodukt und $\{v_1, \dots, v_r\}$ Orthonormalbasis des Untervektorsraums X_r . Sei $V_r \in \mathbb{R}^{d \times r}$ die Matrix mit Spalten $\{v_1, \dots, v_r\}$ und $\mathbb{1}_d$ die Einheitsmatrix in $\mathbb{R}^{d \times d}$.

- (i) Wir definieren die *Projektion* von X auf X_r als

$$\Pi_{X_r} : X \rightarrow X, u \mapsto V_r V_r^T u.$$

- (ii) Wir definieren die zu Π_{X_r} *orthogonale Projektion* mit

$$\Pi_{X_r^\perp} := \mathbb{1}_d - \Pi_{X_r}.$$

Es folgen einige Aussagen über die Projektion, die im weiteren Verlauf vielfach angewandt werden.

Lemma 2.2.3. Für die orthogonale Projektion $\Pi_{X_r^\perp}$ und die Matrix V_r wie in Definition 2.2.2 gilt:

- (i) V_r, V_r^T sind zueinander biorthonormal, das heißt $V_r^T V_r = \mathbb{1}_r$.
- (ii) $\Pi_{X_r^\perp}$ ist idempotent, also $\Pi_{X_r^\perp} \Pi_{X_r^\perp} = \Pi_{X_r^\perp}$.
- (iii) $\Pi_{X_r^\perp}$ ist symmetrisch, also $\Pi_{X_r^\perp}^T = \Pi_{X_r^\perp}$ und somit auch selbstadjungiert.

Beweis. Wir benutzen $\Pi_{X_r^\perp} = \mathbb{1}_d - V_r V_r^T$ aus Definition 2.2.2 und folgern:

- (i) Sei $u \in X$ beliebig, $1 \leq i \leq r$. Nach Konstruktion sind die i -te Zeile von V_r^T und die i -te Spalte von V_r gleich dem Basisvektor v_i und $\{v_1, \dots, v_r\}$ ist eine Orthonormalbasis. Es folgt daher

$$(V_r^T V_r u)_i = \sum_{j=1}^r (V_r^T V_r)_{ij} u_j = \sum_{j=1}^r \langle v_i, v_j \rangle_X u_j = \langle v_i, v_i \rangle_X u_i.$$

Es wurde gezeigt, dass für alle $1 \leq i \leq r$ der Eintrag u_i durch die Matrix wieder auf u_i abgebildet wird. Daher muss $V_r^T V_r$ die Einheitsmatrix sein.

- (ii) Wir setzen die Definition für $\Pi_{X_r^\perp}$ ein (vgl. Definition 2.2.2) und benutzen Aussage (i) dieses Lemmas.

$$\begin{aligned} \Pi_{X_r^\perp} \Pi_{X_r^\perp} &= (\mathbb{1}_d - V_r V_r^T)(\mathbb{1}_d - V_r V_r^T) = \mathbb{1}_d - 2V_r V_r^T + V_r V_r^T V_r V_r^T \\ &= \mathbb{1}_d - 2\Pi_{X_r} + \Pi_{X_r} = \mathbb{1}_d - \Pi_{X_r} \\ &= \Pi_{X_r^\perp} \end{aligned}$$

- (iii) Wie in (ii) setzen wir die entsprechende Definition ein. Zusätzlich wenden Rechenregeln für transponierte Matrizen an.³

$$\begin{aligned} \Pi_{X_r^\perp}^T &= (\mathbb{1}_d - (V_r V_r^T))^T = \mathbb{1}_d^T - (V_r V_r^T)^T \\ &= \mathbb{1}_d - V_r^T V_r^T = \mathbb{1}_d - \Pi_{X_r} \\ &= \Pi_{X_r^\perp} \end{aligned}$$

Da $\Pi_{X_r^\perp} \in \mathbb{R}^{d \times d}$, ist $\Pi_{X_r^\perp}^T$ die adjungierte Matrix. Daher ist $\Pi_{X_r^\perp}$ selbstadjungiert.

³Benutzt wurden [Bos10, 3.2.6, 3.2.7 und 3.2.8].

□

Bemerkung 2.2.4. Die Aussagen (ii) und (iii) aus Lemma 2.2.3 gelten ebenso für Π_{X_r} . Der Beweis erfolgt analog.

Lemma 2.2.5. Sei $\{v_1, \dots, v_r\}$ Orthonormalbasis von X_r .

(i) Die Projektion Π_{X_r} bildet alle $u \in X$ auf X_r ab.

Insbesondere gilt $\Pi_{X_r}v = v$ für alle $v \in \{v_1, \dots, v_r\}$ der Orthonormalbasis von X_r .

(ii) Die orthogonale Projektion $\Pi_{X_r^\perp}$ bildet alle $u \in X$ auf X_r^\perp ab.⁴.

Beweis. Sei $u \in X$ beliebig.

(i) Nach Konstruktion in Definition 2.2.2 gilt

$$\begin{aligned}\Pi_{X_r}u &= V_r V_r^T u = V_r \left(\sum_{j=1}^r v_{1,j} u_j, \dots, \sum_{j=1}^r v_{r,j} u_j \right)^T \\ &= \left(\sum_{i=1}^r \sum_{j=1}^r v_{i,1} v_{i,j} u_j, \dots, \sum_{i=1}^r \sum_{j=1}^r v_{i,r} v_{i,j} u_j \right)^T \\ &= \sum_{i=1}^r v_i \sum_{j=1}^r v_{i,j} u_j.\end{aligned}$$

Als Linearkombination der Basisvektoren $\{v_1, \dots, v_r\}$ von X_r ist $\Pi_{X_r}u$ in X_r .

Sei nun $b = v_k$ für ein $1 \leq k \leq d$. Wir setzen die vorige Gleichung fort. Es gilt

$$\begin{aligned}\sum_{i=1}^r v_i \sum_{j=1}^r v_{i,j} b_j &= \sum_{i=1}^r v_i \langle v_i, b \rangle_X = v_k \langle v_k, b \rangle_X = b \langle v_k, v_k \rangle_X \\ &= b.\end{aligned}$$

(ii) Wir wollen zeigen, dass $\Pi_{X_r^\perp}u$ zu $\Pi_{X_r}u$ orthogonal ist. Hierzu benutzen wir die Aussagen aus dem Lemma 2.2.3 und Bemerkung 2.2.4. Für das Skalarprodukt der beiden Vektoren gilt dann

$$\begin{aligned}\langle \Pi_{X_r}u, \Pi_{X_r^\perp}u \rangle_X &= \langle \Pi_{X_r}u, u \rangle_X - \langle \Pi_{X_r}u, \Pi_{X_r^\perp}u \rangle_X \\ &= \langle \Pi_{X_r}u, u \rangle_X - \langle \Pi_{X_r}^T \Pi_{X_r^\perp}u, u \rangle_X \\ &= \langle \Pi_{X_r}u, u \rangle_X - \langle \Pi_{X_r}u, u \rangle_X = 0.\end{aligned}$$

Das Bild von Π_{X_r} ist nach (i) in X_r und somit orthogonal zum Bild von $\Pi_{X_r^\perp}$, woraus die Behauptung folgt.

□

⁴Mit X_r^\perp ist das orthogonale Komplement von X_r gemeint. Siehe z.B. [Bos10, 7.2.8.].

Lemma 2.2.6. Sei $M \in \mathbb{R}^{n \times n}$ symmetrisch für $n \in \mathbb{N}$ und $\lambda_{\max}(M)$ der größte Eigenwert von M . Dann gilt für alle $x \in \mathbb{R}^n \setminus \{0\}$ mit dem kanonischen Skalarprodukt $\langle \cdot, \cdot \rangle$ und induzierter Norm $\|\cdot\|$

$$\langle x, Mx \rangle \leq \lambda_{\max}(M) \|x\|. \quad (4)$$

Beweis. Da M symmetrisch ist, existiert eine orthogonale Matrix $S \in \mathbb{R}^{n \times n}$ sodass $SMS^T = D$ für eine Diagonalmatrix $D \in \mathbb{R}^{n \times n}$.⁵ Dann folgt für $x \in \mathbb{R}^n \setminus \{0\}$ beliebig unter Zuhilfenahme der Eigenschaften orthogonaler Matrizen

$$\begin{aligned} \langle x, Mx \rangle &= \langle x, SDS^T x \rangle = \langle S^T x, DS^T x \rangle \\ &= \sum_{i=1}^n (S^T x)_i^2 D_{ii} \leq \max_{1 \leq i \leq n} D_{ii} \langle S^T x, S^T x \rangle = \lambda_{\max}(M) \langle x, SS^T x \rangle \\ &= \lambda_{\max}(M) \|x\|. \end{aligned}$$

□

Die im folgenden Teil dieses Abschnittes vorkommenden Untervektorräume von X möchten wir zeitabhängig definieren. Daher benutzen wir folgende Notation aus [BFN17, 1.]:

Notation 2.2.7. Für eine Menge $(X_{r,t})_{t \in I}$ an Untervektorräumen von X mit Orthonormalbasen $(\{v_{1,t}, \dots, v_{r,t}\})_{t \in I}$ und $t \in I$ schreiben wir

$$X_r(t) := (X_{r,t})_t \quad (5)$$

und

$$\{v_1(t), \dots, v_r(t)\} := (\{v_{1,t}, \dots, v_{r,t}\})_t. \quad (6)$$

Die diesen Basen entsprechenden Konstruktionen aus Definition 2.2.2 bezeichnen wir analog mit $V_r(t)$, $\Pi_{X_r}(t)$ und $\Pi_{X_r^\perp}(t)$.

Bemerkung 2.2.8. Die folgende Definition entspricht dem in [BFN17, (6)] definierten System aus Anfangswertproblemen. Es stellt eine Projektion des hochdimensionalen Systems auf die endliche Summe orthogonaler Unterräume dar.

Definition 2.2.9 (Projiziertes Dynamisches System). Es sei (I, Ξ, X, f, u^0) ein dynamisches System nach Definition 2.1.1 und es seien für alle $t \in I$ $X_r(t)$ Untervektorräume von X und $\Pi_{X_r}(t)$ die Projektion von X auf X_r wie in Definition 2.2.2. Wir nennen $(I, \Xi, X_r, \Pi_{X_r}, f, \Pi_{X_r}(0)u^0)$ ein *projiziertes dynamisches System* für ein System aus Anfangswertproblemen

$$\begin{cases} \Pi_{X_r}(t)u'_r(t, \xi) &= \Pi_{X_r}(t)f(u_r, t, \xi) \\ u_r(0, \xi) &= \Pi_{X_r}(0)u^0(\xi) \end{cases}. \quad (7)$$

Zu einer Lösung $u \in X$ nennen wir $u_r = \Pi_{X_r}u \in X_r$ die *projizierte Lösung*.

Bemerkung 2.2.10. $(I, \Xi, X_r, \Pi_{X_r}f, \Pi_{X_r}(0)u_r^0)$, aus (I, Ξ, X_r, f, u^0) konstruiert, ist ein dynamisches System im Sinne der Definition 2.1.1.

⁵Siehe Diagonalisierbarkeit reeller Matrizen, [Bos10, 7.6.6.].

Korollar 2.2.11 (Existenz der Lösungen des projizierten System). Sei (I, Ξ, X, f, u^0) ein dynamisches System wie in Definition 2.1.1, welches die Bedingungen an Satz 2.1.3 erfüllt. Der Fluss $\Pi_{X_r} f$ des reduzierten Systems sei gleichmäßig beschränkt in I . Dann existiert auch für beliebige, feste $\xi \in \Xi$ eine eindeutige Lösung $u_r(\cdot, \xi) \in X_r$ des projizierten dynamischen Systems $(I, \Xi, X_r, \Pi_{X_r} f, \Pi_{X_r}(0)u_r^0)$.

Beweis. Nach Voraussetzung ist der Fluss f von (I, Ξ, X, f, u^0) stetig in $I \times X$ und somit auch in $I \times X_r$. f erfüllt die Lipschitzbedingung (2) für alle $t \in I$, $u, v \in X$ und ein beliebiges, aber festes $\xi \in \Xi$.

Es bleibt zu zeigen, dass der projizierte Fluss $\Pi_{X_r} f$ stetig in $I \times X_r$ ist und die Lipschitzbedingung erfüllt. Die Projektion ist nach Voraussetzung beschränkt und linear, somit stetig in X_r . Folglich ist die Konkatenation $\Pi_{X_r} f$ stetig in $I \times X$.

Weiterhin gilt wegen der Beschränktheit von Π_{X_r} zu einer Konstante $C > 0$ und aufgrund der Lipschitzstetigkeit von f zu einer Konstante $L > 0$ für $f(\cdot, \cdot, \xi)$, beliebiges $\xi \in \Xi$ und alle $t \in I$ die Abschätzung

$$\begin{aligned}\|\Pi_{X_r}(t)f(t, u, \xi) - \Pi_{X_r}(t)f(t, u', \xi)\| &= \|\Pi_{X_r}(t)(f(t, u, \xi) - f(t, u', \xi))\| \\ &\leq C\|f(t, u, \xi) - f(t, u', \xi)\| \\ &\leq CL\|u - u'\|.\end{aligned}$$

Der Fluss $\Pi_{X_r} f$ erfüllt somit die Lipschitzbedingung. \square

Bemerkung 2.2.12. Im Falle einer von Parametern unabhängigen Anfangsbedingung u_r^0 soll das projizierte dynamische System die Anfangsbedingung erhalten, also

$$\Pi_{X_r}(0)u_r^0 = u_r^0. \quad (8)$$

Wir können keine $r > 1$ linear unabhängigen Vektoren aus nur einer Anfangsbedingung erhalten. Um (8) gerecht zu werden, reicht es jedoch aus, allein mit einem normalisierten Anfangswert als Basisvektor diese Projektion gemäß Definition 2.2.2 zu konstruieren. Damit dies nicht zu einem Widerspruch zur Bedingung $\dim X_r = r$ führt, müssen wir voraussetzen, dass $\dim X_r(t) = r$ für $t > 0$ gilt.

Bemerkung 2.2.13. Im Folgenden wird anhand des projizierten Systems wie in Definition 2.2.9 das reduzierte System aus [BFN17, (8)], [BFN17, (9)] definiert. Im Gegensatz zum projizierten System sind dessen Anfangsbedingung, Fluss und Lösungen nicht in \mathbb{R}^d , sondern in \mathbb{R}^r . Dies ermöglicht es uns später, Lösungen des projizierten Systems effizient auf kleinen Arrays zu berechnen.

Definition 2.2.14. Für $\xi \in \Xi$, $u_r(t, \xi)$ in $X_r(t)$ und $t \in I$ definieren wir mit

$$\alpha_r(t, \xi) := V_r^T(t)u_r(t, \xi) \in \mathbb{R}^r \quad (9)$$

und entsprechend für die Anfangsbedingung

$$\alpha_r(0, \xi) := \alpha_r^0 := V_r^T(0)u_r(0, \xi) = V_r^T(0)u^0(\xi) \in \mathbb{R}^r \quad (10)$$

die reduzierte Lösung von u_r . Für den Fluss f eines dynamischen Systems definieren wir durch

$$f_r(\alpha_r, t, \xi) := V_r^T(t)f(V_r(t)\alpha_r, t, \xi) - V_r^T(t)V_r'(t)\alpha_r \quad (11)$$

den reduzierten Fluss. Hierbei bezeichnet $V'_r(t)$ die komponentenweise nach der Zeit differenzierte Matrix $V_r(t)$.

Bemerkung 2.2.15. Damit wir die Existenz der Matrix $V'_r(t)$ annehmen dürfen, müssen wir voraussetzen, dass die Lösung des Systems aus Anfangswertproblemen u aus Satz 2.1.3 differenzierbar in I ist. Diese Eigenschaft setzt sich dann nämlich auf die zeitabhängigen Basisvektoren fort. Nach [Chi99, 1.184.] ist dies sichergestellt und es gilt sogar $u \in C^1(I)$, sofern auch der Fluss stetig differenzierbar ist.

Lemma 2.2.16. *Für u_r Lösung von (7) ist α_r aus (9) Lösung des Systems aus Anfangswertproblemen*

$$\begin{cases} \alpha'_r(t, \xi) = f_r(\alpha_r(t, \xi), t, \xi), \\ \alpha_r(0, \xi) = \alpha_r^0(\xi) \end{cases} . \quad (12)$$

Beweis. Sei $u_r(t, \xi)$ eine Lösung des projizierten Systems wie in Definition 2.2.9 und $\alpha_r(t, \xi) = V_r^T(t)u_r(t, \xi)$. Für $t = 0$ folgt die Aussage direkt aus der Definition des Anfangswerts (10). Durch beidseitiges Addieren von und anschließendes Umstellen nach $u'_r(t, \xi)$ in der ersten Gleichung von (7) erhalten wir

$$\begin{aligned} u'_r(t, \xi) &= \Pi_{X_r}(t)f(u(t, \xi), t, \xi) + u'_r(t, \xi) - \Pi_{X_r}(t)u'_r(t, \xi) \\ &= \Pi_{X_r}(t)f(u(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)u'_r(t, \xi) \\ &= \Pi_{X_r}(t)f(u(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)V'_r(t)\alpha(t, \xi) + \Pi_{X_r^\perp}(t)V_r(t)\alpha'(t, \xi). \end{aligned}$$

Aus der Definition 2.2.2 und 2.2.3(i) folgt, dass

$$\Pi_{X_r^\perp}(t)V(t) = V_r(t) - V_r^T(t)V_r(t)V_r(t) = V_r(t) - V_r(t) = 0.$$

Somit gilt

$$u'_r(t, \xi) = \Pi_{X_r}(t)f(u_r(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)V'_r(t)\alpha(t, \xi).$$

Einsetzen von Definition 2.2.14 für α_r und (11) für f_r liefert das reduzierte r -dimensionale dynamische System $(I, \Xi, \mathbb{R}^r, f_r, \alpha_r^0)$ mit System aus Anfangswertproblemen

$$\begin{cases} \alpha'_r(t, \xi) = f_r(\alpha_r(t, \xi), t, \xi), \\ \alpha_r(0, \xi) = \alpha_r^0(\xi) \end{cases} .$$

□

Lemma 2.2.17. *Der lineare Operator $V_r^T(t)$ ist für alle $t \in I$ beschränkt bezüglich der kanonischen Norm $\|\cdot\|_r$ auf \mathbb{R}^r .*

Beweis. Sei $u \in X_r$. Multiplikation mit $V_r^T(t)$ entspricht der Abbildung

$$u \mapsto (\langle v_1, u \rangle_X, \dots, \langle v_r, u \rangle_X)^T.$$

Für die Norm $\|\cdot\|_r$ erhalten wir mit Cauchy-Schwarz

$$\begin{aligned} \|(\langle v_1, u \rangle_X, \dots, \langle v_r, u \rangle_X)^T\|_r^2 &= \sum_{i=1}^r \langle v_i, u \rangle_X^2 \leq \sum_{i=1}^r \|v_i\|_X^2 \|u\|_X^2 \\ &\leq r \|u\|_X^2 < \infty. \end{aligned}$$

□

Korollar 2.2.18 (Existenz und Eindeutigkeit des reduzierten Systems). *Es sei (I, Ξ, X, f, u^0) ein dynamisches System, welches die Bedingungen an Satz 2.1.3 erfüllt mit Lipschitzkonstante $L > 0$. Sei $V_r(t)$ komponentenweise stetig differenzierbar sowie $V'_r(t)$ stetig und gleichmäßig beschränkt, es existiert also für alle $\alpha \in \mathbb{R}^r$ eine Konstante $C > 0$ mit $\sup_{t \in I} \|V'_r(t)\alpha\|_r \leq C\|\alpha\|_r$. Hierbei ist $\|\cdot\|_r$ die kanonische Norm auf \mathbb{R}^r . Dann existiert eine eindeutige Lösung für $(I, \Xi, \mathbb{R}^r, f_r, \alpha_r^0)$ wie in Definition 2.2.14.*

Beweis. Wir wollen zeigen, dass f_r den Bedingungen an Satz 2.1.3 genügt. Nach Voraussetzung sind $V_r(t), V'_r(t)$ komponentenweise stetig und bilden damit auch Elemente aus \mathbb{R}^r durch Matrixmultiplikation stetig ab. f_r ist somit als Differenz und Konkatenation von in $I \times \mathbb{R}^r$ stetigen Abbildungen ebenso stetig.

Seien nun $\xi \in \Xi$ fest, $t \in I$ und $\alpha_r(\xi), \beta_r(\xi) \in \mathbb{R}^r$ mit $\alpha_r(\xi) = V_r^T(t)u_r(\xi)$ und $\beta_r(\xi) = V_r^T(t)v_r(\xi)$ für $u_r(\xi), v_r(\xi) \in X_r$. Wir wollen für diesen Teil in der Darstellung auf die Abhängigkeit von ξ verzichten. Es gilt zu zeigen, dass f_r die Lipschitzbedingung (2) für die Norm $\|\cdot\|_r$ erfüllt. Wir folgern mithilfe von Lemma 2.2.17

$$\begin{aligned} & \|f_r(t, \alpha_r) - f_r(t, \beta_r)\|_r \\ &= \|V_r^T(t)(f(u_r, t) - f(v_r, t)) - V_r^T(t)V'_r(t)(\alpha_r - \beta_r)\|_r \\ &= \|V_r^T(t)(f(u_r, t) - f(v_r, t)) - V_r^T(t)V'_r(t)V_r^T(t)(u_r - v_r)\|_r \\ &\leq r\|f(u_r, t) - f(v_r, t) - V'_r(t)V_r^T(t)(u_r - v_r)\|_X \\ &\leq r\|f(u_r, t) - f(v_r, t)\|_X + r\|V'_r(t)V_r^T(t)(u_r - v_r)\|_X \\ &\leq Lr\|u_r - v_r\|_X + Cr\|V_r^T(t)(u_r - v_r)\|_r \\ &\leq Lr\|u_r - v_r\|_X + Cr^2\|u_r - v_r\|_X = (Lr + Cr^2)\|u_r - v_r\|_X. \end{aligned}$$

f_r erfüllt die Lipschitzbedingung und das reduzierte System $(I, \Xi, \mathbb{R}^r, f_r, \alpha_r^0)$ besitzt dann nach Korollar 2.2.11 eine eindeutige Lösung. \square

2.3 A-posteriori Fehlerabschätzung

Zum Durchführen der Greedy-Schemen aus Abschnitt 3 benötigen wir eine obere Schranke für den exakten Fehler $\|e_r\|_X = \|u - u_r\|_X$. Sämtliche in diesem Abschnitt aufgeführten Aussagen und Definitionen sind aus [BFN17, 2.2] übernommen und teilweise etwas tiefer ausgeführt.

Lemma 2.3.1. ⁶ *Sei $u(t, \xi)$ Lösung eines dynamischen System aus Definition 2.1.1 und $u_r(t, \xi)$ Lösung eines reduzierten Systemes wie in Definition 2.2.9. Die Zeitableitung $e'_r(t, \xi)$ der Differenz $u(t, \xi) - u_r(t, \xi)$ entspricht dann*

$$\begin{aligned} & f(u(t, \xi), t, \xi) - f(u_r(t, \xi), t, \xi) \\ &+ \Pi_{X_r^\perp}(t)f(u_r(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)V'_r(t)V_r^T(t)u_r(t, \xi) \end{aligned} \tag{13}$$

und es gilt $e_r(0, \xi) = \Pi_{X_r^\perp}(0)u^0(\xi)$.

⁶Vgl. [BFN17, (10)]

Beweis. Wir setzen die Definitionen 2.1.1 und 2.2.9 in $u(t, \xi) - u_r(t, \xi)$ ein und erhalten

$$\begin{aligned}
e'_r(t, \xi) &= u'(t, \xi) - u'_r(t, \xi) \\
&= f(u(t, \xi), t, \xi) - \Pi_{X_r}(t)f(u_r(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)V'_r(t)\alpha(t, \xi) \\
&= f(u(t, \xi), t, \xi) - (1 - \Pi_{X_r^\perp}(t))f(u_r(t, \xi), t, \xi) \\
&\quad + \Pi_{X_r^\perp}(t)V'_r(t)V_r^T(t)u_r(t, \xi) \\
&= f(u(t, \xi), t, \xi) - f(u_r(t, \xi), t, \xi) + \Pi_{X_r^\perp}(t)f(u_r(t, \xi), t, \xi) \\
&\quad + \Pi_{X_r^\perp}(t)V'_r(t)V_r^T(t)u_r(t, \xi).
\end{aligned} \tag{14}$$

Der Fehler zur Zeit $t = 0$ ergibt sich aus den Konstruktionen aus Definition 2.1.1 und Definition 2.2.9 durch

$$\begin{aligned}
e(0, \xi) &= u(0, \xi) - u_r(0, \xi) = u(0, \xi) - \Pi_{X_r}(0)u(0, \xi) \\
&= \Pi_{X_r^\perp}(0)u(0, \xi) = \Pi_{X_r^\perp}(0)u^0(\xi).
\end{aligned}$$

□

Definition 2.3.2 (Lokale logarithmische Lipschitzkonstante).⁷ Der Fluss f eines dynamischen Systems erfülle die Lipschitzbedingung (2). Dann ist die *lokale logarithmische Lipschitzkonstante* von f an der Stelle $v \in X$ definiert durch

$$L_X[f](v) = \sup_{u \in X, u \neq v} \frac{\langle u - v, f(u) - f(v) \rangle_X}{\|u - v\|_X^2}. \tag{15}$$

Lemma 2.3.3 (Berechnung der Lipschitzkonstante für einen affinen Fluss).⁸ Ist f von der Form $f(u) = Au + g$ mit $A \in \mathbb{R}^{d \times d}$ invertierbar und $g \in \mathbb{R}^d$, so gilt $L_X[f](v) = \lambda_{\max}\left(\frac{A+A^T}{2}\right)$, wobei $\lambda_{\max}(M)$ der größte Eigenwert einer Matrix $M \in \mathbb{R}^{d \times d}$ ist.

Beweis. Anhand der Definition der lokalen logarithmischen Lipschitzkonstante (siehe Definition 2.3.2) folgern wir

$$\begin{aligned}
L_X[f](v) &= \sup_{u \in X, u \neq v} \frac{\langle u - v, Au + g - (Av + g) \rangle_X}{\|u - v\|_X^2} \\
&= \sup_{u \in X, u \neq v} \frac{\langle u - v, A(u - v) \rangle_X}{\|u - v\|_X^2} = \sup_{u \in X, u \neq 0} \frac{\langle u, Au \rangle_X}{\|u\|_X^2} \\
&= \sup_{u \in X, u \neq 0} \frac{1}{2} \frac{\langle u, Au \rangle + \langle A^T u, u \rangle}{\|u\|_X^2} = \sup_{u \in X, u \neq 0} \frac{1}{2} \frac{\langle u, Au \rangle + \langle u, A^T u \rangle}{\|u\|_X^2} \\
&= \sup_{u \in X, u \neq 0} \frac{1}{2} \frac{\langle u, (A + A^T)u \rangle}{\|u\|_X^2} = \lambda_{\max}\left(\frac{A + A^T}{2}\right).
\end{aligned}$$

Die Gleichheit der letzten beiden Terme folgt direkt aus dem Lemma 2.2.6, da $A + A^T$ symmetrisch ist. □

⁷Siehe [BFN17, Definition 2.2.]

⁸Vgl. [BFN17, (11)]

Das folgende Lemma hilft uns, die lokale logarithmische Lipschitzkonstante für einen nichtlinearen Fluss zu approximieren. Es ist die leicht abgeänderte Version [BFN17, Lemma 2.3] der Aussage [WSH14, Lemma 2.6]. Die Beweisidee haben wir ebenso aus [BFN17, Lemma 2.3] übernommen und etwas tiefer ausgeführt.

Lemma 2.3.4 (Comparison Lemma). *Seien $T > 0$ und $u, \alpha, \beta: [0, T] \rightarrow \mathbb{R}$ integrierbare Funktionen. Unter der Annahme, dass u differenzierbar ist mit $u' \leq \beta u + \alpha$, gilt, dass $u(t) \leq v(t)$, wobei $v(t)$ Lösung der Differentialgleichung $v' = \beta v + \alpha$ mit Anfangsbedingung $v(0) = u(0)$ ist.*

Beweis. Setze $\gamma(t) = \int_0^t \beta(\tau)d\tau$. Dann ist

$$v(t) = e^{\gamma(t)} \left(\int_0^t e^{-\gamma(\tau)} \alpha(\tau)d\tau + v(0) \right)$$

Lösung von $v' = \beta v + \alpha$. Mit der Voraussetzung $\alpha \geq u' - \beta u$ folgern wir

$$\begin{aligned} v(t) &\geq e^{\gamma(t)} \left(\int_0^t e^{-\gamma(\tau)} (u'(\tau) - \beta(\tau)u(\tau))d\tau + v(0) \right) \\ &= e^{\gamma(t)} \left(\int_0^t e^{-\gamma(\tau)} u'(\tau) - e^{-\gamma(\tau)} \gamma'(\tau)u(\tau)d\tau + v(0) \right) \\ &= e^{\gamma(t)} \left(\int_0^t (e^{-\gamma(\tau)} u(\tau))' d\tau + v(0) \right) \\ &= e^{\gamma(t)} e^{-\gamma(t)} u(t) - e^{\gamma(t)} u(0) + e^{\gamma(t)} u(0) \\ &= u(t) \end{aligned}$$

□

Die folgende Aussage erlaubt es uns den Approximationsfehler $\|e_r\|_X$ abzuschätzen, ohne dass wir eine exakte Lösung kennen müssen. Satz und Beweis finden sich in [BFN17, Proposition 2.4.]. Die einzelnen Beweisschritte sind hier unter Zuhilfenahme voriger Aussagen genauer erklärt.

Proposition 2.3.5. *Die Norm des Fehlers $\|e_r(t, \xi)\|_X$ erfüllt die Ungleichung*

$$\|e_r(t, \xi)\|_X \leq \Delta_r(t, \xi) \tag{16}$$

für alle $t \geq 0$. Hierbei ist $\Delta_r(t, \xi)$ Lösung des Anfangswertproblems

$$\begin{cases} \Delta'_r(t, \xi) &= L_X[f](u_r(t, \xi))\Delta_r(t, \xi) + \|r(t, \xi)\|_X \\ \Delta_r(0, \xi) &= \|e_r(0, \xi)\|_X \end{cases} \tag{17}$$

mit

$$r(t, \xi) = \Pi_{X_r^\perp}(t) \left(V_r'(t) V_r^T(t) u_r(t, \xi) - f(u_r(t, \xi), t, \xi) \right) \tag{18}$$

und

$$e_r(0, \xi) = \Pi_{X_r^\perp}(0) u^0(\xi).$$

L_X ist hier die lokale logarithmische Lipschitzkonstante aus Definition 2.3.2.

Beweis. Da wir den Fehler stets für feste Parameter abschätzen wollen, verzichten wir für diesen Beweis in der Darstellung auf die Abhängigkeit von ξ . Mit der Kettenregel erhalten wir

$$\frac{1}{2} \frac{d}{dt} \|e_r(t)\|_X^2 = \langle e_r(t), e'_r(t) \rangle_X.$$

Einsetzen von $e_r = u - u_r$ und (14) ergibt

$$\left\langle u(t) - u_r(t), (f(u(t), t) - f(u_r(t), t)) + r(t) \right\rangle_X. \quad (19)$$

Wir setzen die Definition 2.3.2 der lokalen logarithmischen Lipschitzkonstante ein und erhalten

$$L_X[f](u_r(t)) \|e_r(t)\|_X^2 + \langle e_r(t), r(t) \rangle_X. \quad (20)$$

Anwenden von Cauchy-Schwarz liefert dann

$$L_X[f](u_r(t)) \|e_r(t)\|_X^2 + \|e_r(t)\|_X \|r(t)\|_X. \quad (21)$$

Wir bemerken $\frac{1}{2} \frac{d}{dt} \|e_r(t)\|_X^2 = \|e_r(t)\|_X \frac{d}{dt} \|e_r(t)\|_X$ und folgern daraus

$$\frac{d}{dt} \|e_r(t)\|_X \leq L_X[f](u_r(t)) \|e_r(t)\|_X + \|r(t)\|_X. \quad (22)$$

Wir wenden Lemma 2.3.4 mit $u' = \frac{d}{dt} \|e_r(t)\|_X$ und $v' = \Delta'_r(t, \xi)$ an:

Wegen (17) gilt die Bedingung $v(0) = u(0)$, aus (22) erhalten wir die Abschätzung für u' mit $\alpha = \|r(t)\|_X$ und $\beta = L_X[f](u_r(t))$ sowie die Gleichung für v' aus (17). Mit der Aussage von Lemma 2.3.4 folgt dann die behauptete Fehlerschätzung (16). \square

3 Greedy-Algorithmen zur Konstruktion reduzierter Räume

In diesem Abschnitt wollen wir die beiden Greedy-Schemen aus [BFN17, 3.] zur Konstruktion reduzierter Basen vorstellen. Von nun an bezeichne Ξ_{train} eine beliebige endliche Teilmenge von Ξ .

Definition 3.0.1. ⁹ Es bezeichne $\|\cdot\|_{(0,T),2}$ die natürliche Norm in $L^2([0, T])$. Wir definieren eine a-posteriori Abschätzung der $L^2(0, T)$ -Norm des Fehlers $\|e_r(\cdot, \xi)\|_X$ durch

$$\Delta_r^{(0,T)}(\xi) := \|\Delta_r(\cdot, \xi)\|_{(0,T),2} = \left(\int_0^T \Delta_r(t, \xi)^2 dt \right)^{\frac{1}{2}}. \quad (23)$$

Bemerkung 3.0.2. (17) liefert uns eine Fehlerabschätzung zu jedem Zeitpunkt $t \in I$. Anstatt die Auswahl des nächsten Parameters ξ^{r+1} im r -ten Reduktions schritt von einer globalen Norm abhängig zu machen, können wir den Parameter auch für jeden Zeitpunkt $t \in I$ separat wählen.

⁹Vgl. entsprechende Definition in [BFN17, 3.].

Jedoch müsste dann die Lösung des hoch-dimensionalen Problems für jeden dieser Parameter $(\xi^r(t))_{t \in I}$ berechnet werden. Im worst-case müssten also bereits im ersten Reduktionsschritt alle hoch-dimensionalen Lösungen teuer berechnet werden. Eine nach (25) definierte reduzierte Basis bildet nur die hochdimensionale Lösung $u(t, \xi^r(t))$ für alle $t \in I$ sicher und annähernd exakt ab. Dies ist im Allgemeinen eine schlechtere Approximation als die reduzierte Basis aus (24) für einen festen Parameter liefert. Siehe dazu die Resultate in Abschnitt 5.2.

Bemerkung 3.0.3. Zum Vergleichen zweier Fehlerabschätzungen in der Norm $\|\cdot\|_{(0,T),2}$ können wir auf die Einbeziehung des Intervalls $(0, T)$ verzichten. Für eine diskrete endliche Zeitdomäne $\{t^0, \dots, t^K\} \subset (0, T)$ mit $K \in \mathbb{N}$ werten wir die Norm $\|\cdot\|_{(0,T),2}$ daher als die euklidische Norm $\|\cdot\|_2$ über \mathbb{R}^K aus.

Bemerkung 3.0.4. Statt der $L^2(0, T)$ -Norm können wir auch andere Normen wie die $L^\infty(0, T)$ -Norm oder eine gewichtete $L^2(0, T)$ -Norm benutzen um den Fehler $\Delta_r^{(0,T)}(\xi)$ zu definieren. Siehe auch [BFN17, Remark 3.1].

Definition 3.0.5. Sei X_r ein Unterraum von X . Wir möchten herausfinden, welcher Parameter aus Ξ_{train} die a-posteriori Fehlerschätzung (17) maximiert. Die Abschätzung (16) liefert uns eine obere Schranke für den exakten Approximationsfehler $e_r(t, \xi)$ für alle $t \in I$, $\xi \in \Xi$. Die Folge reduzierter Räume definieren wir wie in [BFN17, 3.1.] mit

$$X_{r+1}(t) = X_r(t) + \text{span}\{u(t, \xi^{r+1})\} \quad (24)$$

für alle $t \in I$, wobei $\xi^{r+1} = \arg \max_{\xi \in \Xi} \Delta_r^{(0,T)}(t, \xi)$. Der Vektorraum $\text{span}\{u(t, \xi)\}$ ist hier die lineare Hülle von $\{u(t, \xi)\} \subset X$.

Bemerkung 3.0.6. Über die Konstruktion (24) können wir zwar eine Basis $\{u(t, \xi^1), \dots, u(t, \xi^{r+1})\}$ für X_{r+1} erhalten, diese ist aber im Allgemeinen nicht orthogonal. Durch die Abbildungen $\Pi_{X_{s-1}^\perp}(t)$ für $1 \leq s \leq r$ können wir aus den Snapshots $u(t, \xi^s)$ jedoch über

$$v_s(t) = \frac{\Pi_{X_{s-1}^\perp}(t)u(t, \xi^s)}{\|\Pi_{X_{s-1}^\perp}(t)u(t, \xi^s)\|_X} \quad (25)$$

eine Orthonormalbasis $\{v_1(t), \dots, v_r(t)\}$ berechnen.¹⁰ Aufgrund der Eigenschaft (ii) aus Lemma 2.2.5 der orthogonalen Projektion ist der Vektor $v_s(t)$ dann orthogonal zu allen $u \in X_{s-1}$. So stellen wir theoretisch sicher, dass zusätzliche Basiserweiterungen die Ergebnisse für bereits berücksichtigte Parameter nicht beeinflussen.¹¹

Bemerkung 3.0.7. (i) Es stellt sich heraus, dass Resultate zur algebraischen wie exponentiellen Konvergenz des Greedy-Verfahrens (siehe [BCD⁺11, 3.1, 3.2]) auch die entsprechende Konvergenz des POD-Greedy-Verfahrens impliziert (siehe [Haa13, 4.1, 4.2]). Ähnliche Resultate für den T-Greedy sind nicht bekannt.

¹⁰Wir setzen hierfür $\Pi_{X_0^\perp}(t) = \mathbb{1}_d$

¹¹In der konkreten Implementation können wir keine exakte Orthogonalität garantieren. Vielmehr arbeiten wir mit einer oberen Schranke für akzeptable Werte der Skalarprodukte.

- (ii) Theoretische Aussagen zum Verhalten von Greedy-Verfahren sind zum Beispiel in [Tem11] zu finden.

Definition 3.0.8 (POD-Greedy). Das POD-Greedy Verfahren, ursprünglich in [HO08] vorgestellt, ist ein gängiges Verfahren zur Reduktion Parameter- und zeitabhängiger dynamischer Systeme. Neben einem effizienten Zeitschrittverfahren für die Lösungen u_r des reduzierten Systems bedarf das Greedy-Schema eines a-posteriori Fehlerschätzers Δ_r ¹². Für den ersten Parameter ξ^0 , dessen hoch-dimensionale Lösung $(u(t^k, \xi^0))_{k=0}^K$ für $k \in \mathbb{N}$ Zeitschritte vorliegt, bestimmen wir die $l \in \mathbb{N}$ Indizes von $(u(t^k, \xi^0))_{k=0}^K$ mit den größten Singulärwerten. Die korrespondierenden Einträge der Lösung nennen wir *POD-Modes*. Wir erhalten mit den l POD-Modes somit die in gewisser Weise ausschlaggebendsten Komponenten des Approximationsfehlers. Im Falle $l = 1$ konstruieren wir eine Basis gemäß 25.

```

input : Dynamisches System  $(I, \Xi_{train}, X, f, u^0)$ , geforderte Genauigkeit
         $\varepsilon$ , Testparameter  $\Xi_{train}$ 
output: Approximierter Zustandsraum  $X_r(t)$ 

begin
    set:  $r \leftarrow 0$  and  $X_r \leftarrow \{0\}$ 
    repeat
        foreach  $\xi \in \Xi_{train}$  do
            | calculate:  $u_r(\cdot, \xi)$  and  $\Delta_r(\cdot, \xi)$ 
        end
        find:  $\xi^{r+1} \in \arg \max_{\xi \in \Xi_{train}} \Delta_r^{(0,T)}(\xi)$ 
        if  $\Delta_r^{(0,T)}(\xi^{r+1}) < \varepsilon$  then
            | return  $X_r(t)$ 
        end
        calculate:  $u(\cdot, \xi^{r+1})$  and  $s_r(\cdot, \xi^{r+1}) = \Pi_{X_r^\perp} u(\cdot, \xi^{r+1})$ 
                  //Basiserweiterung
        set:  $X_{r+l} \leftarrow X_r + \text{span}\{POD_l(s_r(\cdot, \xi^{r+1}))\}$ 
        set:  $r \leftarrow r + l$ 
    end
end

```

Algorithmus 2: POD-Greedy-Algorithmus nach [BFN17, Algorithm 2].

Es wurde die dort angemerkt Abbruchbedingung $\Delta_r^{(0,T)}(\xi^{r+1}) < \varepsilon$ hinzugefügt. Zusätzlich wurde der Pseudocode übersichtlicher gestaltet.

Bemerkung 3.0.9. In [BFN17, 3.2] wurde die langsame Approximation für den Fall $l = 1$ kurz angemerkt. Da in dem Paper keine weiteren Aussagen zur Anzahl der verwendeten POD-Modes in den numerischen Experimenten gemacht wurde, gehen wir für die eigene Umsetzung des Greedy-Verfahrens stets von $l = 1$ aus.

Bemerkung 3.0.10. Sollte es mehr als einen Parameter geben, der die Fehlerabschätzung maximiert, so wählen wir nur einen davon aus. Dabei sollte jede Wahl zielführend sein. In den durchgeführten Experimenten in Abschnitt 5 unterliegen alle Parameter einer totalen Ordnung. Wir wählen im Zweifel immer den kleinsten Parameter.

¹²In den Experimenten verwenden wir den Fehlerschätzer (17) für den Spezialfall zeitunabhängiger reduzierte Räume $X_r(t) = X_r(s) \forall s, t \in I$. Siehe auch Bemerkung 4.3.4.

Definition 3.0.11 (T-Greedy).

```

input : Dynamisches System  $(I, \Xi_{train}, X, f, u^0)$ , geforderte Genauigkeit
         $\varepsilon$ , Testparameter  $\Xi_{train}$ 
output: Approximierter Zustandsraum  $X_r(t)$ 

begin
    | set:  $r \leftarrow 0$  and  $X_r \leftarrow \{0\}$ 
    | repeat
    |   | foreach  $\xi \in \Xi_{train}$  do
    |   |   | calculate:  $u_r(\cdot, \xi)$  and  $\Delta_r(\cdot, \xi)$ 
    |   |   | end
    |   | find:  $\xi^{r+1} \in \arg \max_{\xi \in \Xi_{train}} \Delta_r^{(0,T)}(\xi)$ 
    |   | if  $\Delta_r^{(0,T)}(\xi^{r+1}) < \varepsilon$  then
    |   |   | return  $X_r(t)$ 
    |   |   | end
    |   | calculate:  $t \mapsto u(t, \xi^{r+1})$ 
    |   | set:  $X_{r+1}(t) \leftarrow X_r(t) + \text{span}\{u(t, \xi^{r+1})\}$  //Basiserweiterung
    |   | set:  $r \leftarrow r + 1$ 
    |   | end
    | end

```

Algorithmus 4: T-Greedy-Algorithmus nach [BFN17, Algorithm 1.]. Es wurde die dort angemerkt Abbruchbedingung $\Delta_r^{(0,T)}(\xi^{r+1}) < \varepsilon$ hinzugefügt. Zusätzlich wurde der Pseudocode übersichtlicher gestaltet.

Bemerkung 3.0.12. In den Algorithmen 1 und 2 ist eine geforderte Genauigkeit ε als Abbruchbedingung des Verfahrens angegeben. Dies ist eine für konkrete Anwendungen dieser Routinen angemessene Wahl, da man dort eher an solch einem maximalen Approximationsfehler über Parameter in Ξ_{train} interessiert ist. In den Experimenten in [BFN17, 5.1.] und entsprechend auch hier in Abschnitt 5 brechen wir die Basiskonstruktion hingegen nach einer festen Anzahl an Basiserweiterungen ab, um die Qualität der Verfahren direkt vergleichen zu können.

Bemerkung 3.0.13. Wir möchten hier das originäre Basiserweiterungsverfahren aus [BFN17] mit der Umsetzung des POD-Greedy-Verfahrens vergleichen.

POD-Greedy und T-Greedy unterscheiden sich vor allem in der Konstruktion der Basis (siehe Algorithmen 12 Kommentar). In beiden Fällen wird die reduzierte Basis um einen orthogonalen Vektor erweitert. Der große Unterschied der beiden Verfahren ist der Umgang mit der Zeitabhängigkeit des Problems. Während beim POD-Greedy $1 \leq l \leq \dim(X)$ einzelne Basisvektoren aus der Singulärwertzerlegung der Zeittrajektorie einer Lösung hervorgehen, wird beim T-Greedy jede Basis $X_r(t)$ für $t > 0$ nur um eine Dimension vergrößert. Außerdem spart sich der T-Greedy-Algorithmus die Singulärwertzerlegung.

Da der T-Greedy-Algorithmus für alle $K \in \mathbb{N}$ Zeitschritte je eine Basis benötigt, ist der Speicheraufwand für eine r -dimensionale reduzierte Basis gleich $d \cdot r \cdot K$ Fließkommazahlen¹³, wobei $d \in \mathbb{N}$ die Dimension des vollen Problems ist. Hingegen muss der POD-Greedy nur $d \cdot r$ Fließkommazahlen abspeichern.

¹³Im Falle einer Parameter-unabhängigen Anfangsbedingung nur $d \cdot r \cdot (K - 1)$ Fließkommazahlen.

Um das hoch-dimensionale Problem mit der reduzierten Basis effizient zu approximieren, wird ein günstiges Zeitschrittverfahren benötigt. Das Lösungsverfahren niedriger Dimension¹⁴ kann für beide Algorithmen verwendet werden. Für zeitunabhängige Basismatrizen vereinfacht sich das Verfahren, da in diesem Falle $\frac{\delta V}{\delta t} = 0$ gilt. Ob die reduzierte Basis durch mehrere POD-Modes aus Lösungen für verschiedene oder nur für einen Parameter generiert wurde, ist für den Ablauf das Zeitschrittverfahren unerheblich. Trotz Verwendung von K Basen im T-Greedy sollte der Rechenaufwand des Lösungsverfahrens in beiden Fällen gleich sein, sofern man von den Kosten für die Berechnung der zeitabhängigen Größen absieht.

Auch die in (42) definierte iterativ berechenbare Fehlerabschätzung $\Delta_r(t, \xi)$ kann für beide Algorithmen benutzt werden. Im POD-Greedy sind die Basen nicht zeitabhängig. Das Berechnung des Residuums (18) inklusive Speicherung benötigter Offlinematrizen gestaltet sich dann weniger aufwändig (siehe Bemerkung 4.3.4).

4 Implementation des Verfahrens

Nach Reduktion des dynamischen Systems liegt dieses als System gewöhnlicher Differentialgleichungen vor. Auch die Fehlerabschätzung Δ_r ist als Lösung eines Anfangswertproblems definiert. Zur Berechnung dieser Werte werden in [BFN17, 4.] numerische Verfahren vorgeschlagen, welche wir hier darstellen und untersuchen möchten.

Um die Struktur des Flusses $f: X \times (0, T) \times \Xi \rightarrow X$ des dynamischen Systems berücksichtigen zu können, teilen wir diesen wie in [BFN17, 4.1.] in einen linearen Teil $A(t, \xi) \in \mathbb{R}^{d \times d}$, einen nichtlinearen Teil $h(\cdot, t, \xi): X \rightarrow \mathbb{R}^d$ und den Vektor $g(t, \xi) \in \mathbb{R}^d$ auf:

$$f(u, t, \xi) = A(t, \xi)u + h(u, t, \xi) + g(t, \xi) \quad (26)$$

Wir gehen von einer gleichmäßigen Diskretisierung der Zeitdomäne des instationären Problems aus. Für die zeitabhängigen Größen benutzen wir folgende Schreibweise:

Notation 4.0.1. Sei $\mathbb{T} = \{t^k\}_{k=0}^K$ die Zeitdiskretisierung des Intervalls $[0, T]$ gleichmäßiger Schrittweite mit $K \in \mathbb{N}$ Schritten. Wir schreiben $t^k := k \cdot \delta t$, wobei $\delta t := \frac{T}{K}$. Für eine Abbildung a aus $[0, T]$ definieren wir wie in [BFN17, 4.1.]

$$a^k := a(t^k)$$

und

$$\delta a^k := a^{k+1} - a^k.$$

4.1 Zeitintegration des dynamischen Systems

Es bezeichne X_r^k den reduzierten Zustandsraum zur Zeit t^k mit Orthonormalbasis $\{v_1^k, \dots, v_r^k\}$, also $X_r^k = X_r(t^k)$. Mit der Matrix $V_r^k = [v_1^k, \dots, v_r^k]$ erhalten

¹⁴siehe [BFN17, 4.2.1.] und Lemma 4.1.4

wir aus der Definition 2.2.2 die Projektionsmatrizen $\Pi_{X_r^k} = V_r^{k^T} V_r^k$ auf X_r^k und $\Pi_{X_r^{k+1}} = \mathbb{1}_d - \Pi_{X_r^k}$. Es ergibt sich eine Approximation u_r^k zur Zeit t^k , indem wir das diskrete dynamische System (1) durch Linksmultiplikation von $\Pi_{X_r^{k+1}}$ auf X_r^{k+1} projizieren. Vergleiche hierzu auch Definition 2.2.9.

Definition 4.1.1 (Reduzierte Operatoren). Für die Operatoren A, g aus (26) definieren wir anhand der Basismatrizen aus Definition 2.2.2 mit

$$A_r^k(\xi) := V_r^{k^T} A^k(\xi) V_r^k \in \mathbb{R}^{r \times r} \quad (27)$$

und

$$g_r^k(\xi) := V_r^{k^T} g^k(\xi) \in \mathbb{R}^r \quad (28)$$

die *reduzierten Operatoren*.

Bemerkung 4.1.2. Das EIM-Verfahren (kurz für *Empirical Interpolation Method*) ist ein Greedy-Verfahren zur Approximation eines nichtlinearen Operators (siehe [BMS13, 4.]). Wir schildern hier kurz die Verwendung dieser Methode in [BFN17, 4.2.1.]. Da die in Abschnitt 5 durchgeföhrten Experimente keine solche Nichtlinearitäten beinhalten, möchten wir von einer tiefen Untersuchung dieses Verfahrens absehen.

Definition 4.1.3 (EIM-Approximation des nichtlinearen Operators). Sei $h: X \times I \times \Xi \rightarrow X$ der nichtlineare Anteil eines Flusses wie in (26). Zu Snapshots $(u(t, \xi^i))_{i=0}^r$ der hoch-dimensionalen Lösung eines dynamischen Systems und einer gegebenen Genauigkeit $\varepsilon > 0$ berechnen wir eine sogenannte EIM-Basis $H_m := \{h_1, \dots, h_m\} \subset X$:

Mittels eines dem POD-Greedy Schema (siehe Algorithmus 1) ähnlichen Verfahrens werden der Indexmenge $\mathcal{I}_m = \{i_1, \dots, i_m\} \subset \{1, \dots, d\}$ solange Elemente hinzugefügt, bis der Fehler der Approximation

$$\max_{\substack{0 \leq k \leq K \\ 1 \leq i \leq r}} \|\tilde{h}(u_r(t^k, \xi^i), t^k, \xi^i) - h(u_r(t^k, \xi^i), t^k, \xi^i)\|_X$$

die geforderte Genauigkeit ε unterschreitet. Die *EIM-Approximation* \tilde{h} von h ist dann definiert durch

$$\tilde{h}(u_r(t, \xi), t, \xi) := U_m P_m^T h(u_r(t, \xi), t, \xi), \quad (29)$$

wobei

$$U_m := H_m (H_m P_m^T)^{-1}, \quad (30)$$

und

$$P_m \in \mathbb{R}^{d \times m} \quad (31)$$

die Matrix mit Spalten \mathcal{I}_m der Einheitsmatrix $\mathbb{1}_d$ ist.

Lemma 4.1.4 (Niedrigdimensionales Zeitschrittverfahren). Für alle $0 \leq k \leq K - 1$ und beliebiges $\xi \in \Xi$ sei $u_r^{k+1}(\xi) \in X_r^k$ Lösung des projektionsdynamischen Systems (7) zum reduzierten Anfangswert $u_r^0(\xi) \in X_r^0$ und für alle $0 < k \leq K$. Für die reduzierte Lösung $\alpha_r^k(\xi) \in \mathbb{R}^r$ gelte

$$\begin{aligned} & (\mathbb{1}_r - \delta t A_r^{k+1}(\xi)) \alpha_r^{k+1}(\xi) \\ &= \alpha_r^k(\xi) + \delta t V_r^{k+1T} \left(h^k(V_r^{k+1} a_r^{k+1}(\xi), \xi) + g^k(\xi) - \frac{\delta V_r^k}{\delta t} \alpha_r^k(\xi) \right) \end{aligned} \quad (32)$$

sowie

$$a_r^0(\xi) = V_r^{0T} u_r^0(\xi). \quad (33)$$

Dann stimmt dieses $\alpha_r^k(\xi)$ mit Definition 2.2.14 überein, es gilt also $u_r^k(\xi) = V_r^k \alpha_r^k(\xi)$ für alle $0 \leq k \leq K$.

Beweis. Wir zeigen die Aussage für beliebiges $k + 1 = 0, \dots, K$. Einsetzen von (9) ergibt

$$\begin{aligned} \alpha_r^{k+1}(\xi) &= V_r^{k+1T} u_r^{k+1}(\xi) \\ &= V_r^{k+1T} u_r^k(\xi) + \delta t V_r^{k+1T} (A^{k+1}(\xi) u_r^{k+1}(\xi) + h^k(u_r^{k+1}(\xi), \xi) + g^k(\xi)) \\ &= V_r^{k+1T} V_r^k \alpha_r^k(\xi) + \delta t V_r^{k+1T} A^{k+1}(\xi) V_r^{k+1} u_r^{k+1}(\xi) \\ &\quad + \delta t V_r^{k+1T} (h^k(u_r^{k+1}(\xi), \xi) + g^k(\xi)). \end{aligned}$$

Wir subtrahieren den Term $A^{k+1}(\xi) u_r^{k+1}(\xi)$ auf beiden Seiten und benutzen Definition 4.1.1.

$$\begin{aligned} & (\mathbb{1}_r - \delta t A_r^{k+1}(\xi)) \alpha_r^{k+1}(\xi) \\ &= \delta t V_r^{k+1T} \left(\frac{V_r^k \alpha_r^k(\xi)}{\delta t} + h^k(u_r^{k+1}(\xi), \xi) + g^k(\xi) \right) \\ &= \delta t V_r^{k+1T} \left(\frac{V_r^k \alpha_r^k(\xi) + V_r^{k+1} \alpha_r^k(\xi) - V_r^{k+1} \alpha_r^k(\xi)}{\delta t} \right. \\ &\quad \left. + h^k(u_r^{k+1}(\xi), \xi) + g^k(\xi) \right) \\ &= V_r^{k+1T} V_r^{k+1} \alpha_r^k(\xi) + \delta t V_r^{k+1T} \left(h^k(u_r^{k+1}(\xi), \xi) + g^k(\xi) - \frac{\delta V_r^k}{\delta t} \alpha_r^k(\xi) \right) \\ &= \alpha_r^k(\xi) + \delta t V_r^{k+1T} \left(h^k(V_r^{k+1} a_r^{k+1}(\xi), \xi) + g^k(\xi) - \frac{\delta V_r^k}{\delta t} \alpha_r^k(\xi) \right) \end{aligned}$$

□

Bemerkung 4.1.5. Wie in [BFN17, 4.2.1.] angemerkt, können die Matrizen $V_r^{k+1T} U_m \in \mathbb{R}^{r \times m}$ in jedem Schritt der Basisreduktion einmalig berechnet werden, um im Zeitschrittverfahren (32) diese auf den Vektor $P_m^T h(u_r(t^k, \xi), t^k, \xi) \in \mathbb{R}^m$ für $0 \leq k \leq K$ und beliebige $\xi \in \Xi$ anzuwenden. Wir erhalten dadurch eine EIM-Approximation wie in (29). Zur effizienten Berechnung des Residuums der Fehlerabschätzung (41) via Offline-/Online-Zerlegung benötigen wir dann zusätzlich die Matrizen U_m und P_m (siehe Definition 4.3.1).

4.2 Zeitintegration der Fehlerabschätzung

Für die Berechnung der Fehlerschätzung aus Proposition 2.3.5 benötigen wir ein Verfahren zur Bestimmung der Lipschitzkonstante aus Definition 2.3.2. Zuerst stellen wir fest, dass $\Delta_r(t, \xi)$ weiterhin eine obere Schranke für den exakten Fehler $\|e_r(t, \xi)\|$ ist, wenn wir die Konstante nach oben abschätzen (vgl. (17)). Zusammen mit (26) erhalten wir folgende Abschätzung für $L_X[f]$:

Lemma 4.2.1. *Es sei $L_X[f](u)$ die lokale logarithmische Lipschitzkonstante für den Fluss f eines dynamischen Systems, welcher der Voraussetzung (26) genügt und $u \in X$. Dann gilt*

$$L_X[f](u) \leq L_X[A] + L_X[h](u) \quad (34)$$

Beweis. Sei $u \in X$. Wir setzen die Definition der lokalen logarithmischen Lipschitzkonstante aus 2.3.2 ein und folgern:

$$\begin{aligned} L_X[f](u) &= \sup_{u \in X, u \neq v} \frac{\langle u - v, f(u) - f(v) \rangle_X}{\|u - v\|_X^2} \\ &= \sup_{u \in X, u \neq v} \frac{\langle u - v, Au - Av + h(u) - h(v) \rangle}{\|u - v\|_X^2} \\ &= \sup_{u \in X, u \neq v} \frac{\langle u - v, Au - Av \rangle_X + \langle u - v, h(u) - h(v) \rangle_X}{\|u - v\|_X^2} \\ &\leq \sup_{u \in X, u \neq v} \frac{\langle u - v, A(u - v) \rangle_X}{\|u - v\|_X^2} + \sup_{u \in X, u \neq v} \frac{\langle u - v, h(u) - h(v) \rangle_X}{\|u - v\|_X^2} \\ &= L_X[A] + L_X[h](u) \end{aligned}$$

□

Definition 4.2.2. Für alle $t \in I$ seien $A(t, \xi) \in \mathbb{R}^{d \times d}$ und $g(t, \xi) \in \mathbb{R}^d$. Dann sagen wir, dass A, g eine *zeitabhängige affine Repräsentation* besitzen, falls es $Q_A, Q_g \in \mathbb{N}_0$, $A^i: I \rightarrow \mathbb{R}^{d \times d}$, $g^j: I \rightarrow \mathbb{R}^d$ und Funktionen $\theta_A^i, \theta_g^j: I \times \Xi \rightarrow \mathbb{R}$ mit $0 \leq i \leq Q_A$, $0 \leq j \leq Q_g$ gibt, sodass

$$A(t, \xi) = \sum_{i=1}^{Q_A} \theta_A^i(t, \xi) A^i(t) \quad (35)$$

und

$$g(t, \xi) = \sum_{i=1}^{Q_g} \theta_g^i(t, \xi) g^i(t). \quad (36)$$

Definition 4.2.3. Sei $A: I \times \Xi \rightarrow \mathbb{R}^{d \times d}$ mit zeitabhängiger affiner Repräsentation $A(t, \xi) = \sum_{i=1}^{Q_A} \theta_A^i(t, \xi) A^i(t)$ wie in Definition 4.2.2. Sei weiterhin $\tilde{h}^k: X \times I \times \Xi$ eine Approximation des nichtlinearen Flusses (siehe Definition 4.1.3). Dann definieren wir die Abschätzung der *lokalen logarithmischen Lipschitzkonstante* L_X aus Definition 2.3.2 zu A und h für alle $u \in X$ jeweils durch

$$\tilde{L}[A(t, \xi)] := \sum_{i=1}^{Q_A} |\theta_A^i(t, \xi)| |L_X[A^i(t)]| \quad (37)$$

und

$$\tilde{L}[h](u(t, \xi), t, \xi) := \sum_{i=1}^r \gamma^i(\xi) L_X[\nabla h](u(t, \xi^i), t, \xi^i). \quad (38)$$

Hierbei ist ∇h der Gradient von h und

$$\gamma^i(\xi) := \begin{cases} 1 & \text{falls } d(\xi, \xi^i) = \min_{1 \leq j \leq r} d(\xi^i, \xi^j) \\ 0 & \text{sonst} \end{cases}$$

zu einer Metrik $d(\cdot, \cdot)$ auf Ξ .

Bemerkung 4.2.4. Da der Rechenaufwand zur Bestimmung des größten Eigenwertes von $A(t, \xi) \in \mathbb{R}^{d \times d}$ abhängig von d ist, möchten wir diese Berechnung ungern für jeden Parameter erneut vornehmen. Die Operatoren aus Definition 4.2.2 sind Parameter-unabhängig. Wir müssen $L_X[A^i(t)]$ aus Definition 4.2.3 für alle $1 \leq i \leq Q_A$ nur einmalig auswerten. Dann können wir $L_X[A(t, \xi)]$ für ein $\xi \in \Xi$ abschätzen, indem wir die Linearkombination (37) berechnen.

Bemerkung 4.2.5. In [BFN17, 4.2.2.] wird darauf hingewiesen, dass die Konstruktion (38) keine echte obere Schranke für $L_X[h]$ darstellt. Die Konstante wird durch die Taylorreihe erster Ordnung für Operatoren approximiert.¹⁵ Laut [BFN17, 4.2.2.] ist dies für den in [BFN17, 5.3.] durchgeführten Test eine zufriedenstellende Approximation. Da wir dieses Experiment in Abschnitt 5 nicht nachstellen, werden wir diese Aussage auch nicht untersuchen.

Die folgende Aussage erlaubt es uns, die Approximationen $\tilde{L}[A], \tilde{L}[h]$ aus Definition 4.2.3 statt den exakten Lipschitzkonstanten in (34) zu verwenden.

Lemma 4.2.6. Für $A: I \times \Xi \rightarrow \mathbb{R}^{d \times d}$ mit zeitabhängiger affiner Repräsentation gilt für alle $t \in I$ die Ungleichung

$$L_X[A](t) \leq \tilde{L}_X[A](t). \quad (39)$$

Beweis. Sei $\lambda_{\max}\left(\frac{A(t, \xi) + A^T(t, \xi)}{2}\right)$ der größte Eigenwert von $\frac{1}{2}(A(t, \xi) + A^T(t, \xi))$ für $t \in I, \xi \in \Xi$. Wir setzen Definition 4.2.2 der zeitabhängigen affinen Repräsentation in Definition 2.3.3 von $L_X[A](t)$ aus ein und nutzen die Linearität des Eigenwerts:

$$\begin{aligned} & L_X[A(t, \xi)] \\ &= \lambda_{\max}\left(\frac{A(t, \xi) + A^T(t, \xi)}{2}\right) = \lambda_{\max}\left(\sum_{i=1}^{Q_A} \theta_A^i(t, \xi) \frac{A^i(t) + A^{i^T}(t)}{2}\right) \\ &\leq \sum_{i=1}^{Q_A} \lambda_{\max}\left(\theta_A^i(t, \xi) \frac{A^i(t) + A^{i^T}(t)}{2}\right) = \sum_{i=1}^{Q_A} \theta_A^i(t, \xi) \lambda_{\max}\left(\frac{A^i(t) + A^{i^T}(t)}{2}\right) \\ &= \sum_{i=1}^{Q_A} \theta_A^i(t, \xi) L_X[A^i(t)] = \tilde{L}_X[A(t, \xi)] \end{aligned}$$

□

¹⁵Siehe z.B. [HP57, Abschnitt 3.6], insbesondere [HP57, Theorem 3.16.2].

Wir definieren die Approximation $\tilde{\Delta}_r$ von Δ_r aus Proposition 2.3.5, indem wir die Lipschitzkonstanten durch die entsprechenden Approximationen austauschen. Für den Fall $h \equiv 0$ stellt $\tilde{\Delta}_r$ weiterhin eine obere Schranke zum normierten exakten Fehler $\|e_r\|_X$ dar.¹⁶

Definition 4.2.7. Sei $\tilde{\Delta}_r$ definiert durch

$$\begin{cases} \tilde{\Delta}'_r(t, \xi) &= \tilde{L}_X[A]\tilde{\Delta}_r(t, \xi) + \tilde{L}_X[h](u_r)\tilde{\Delta}_r(t, \xi) + \|\tilde{r}(t, \xi)\|_X \\ \tilde{\Delta}_r(0, \xi) &= \|e_r(0, \xi)\|_X \end{cases} \quad (40)$$

mit

$$\tilde{r}^k(\xi) = \Pi_{X_r^\perp}^{k+1} \left(\frac{\delta V_r^k}{\delta t} \alpha_r^k(\xi) - A^{k+1}(\xi) u_r^{k+1} - \tilde{h}^k(u_r^k(\xi), \xi) - g^k(\xi) \right). \quad (41)$$

Wir lösen dieses Anfangswertproblem dann durch folgendes Schema:

$$\tilde{\Delta}_r^{k+1}(\xi) = (1 - \delta t \tilde{L}_X[A^{k+1}(\xi)])^{-1} \left(\tilde{\Delta}_r^k(\xi) + \delta t \tilde{L}_X[h^k](u_r^k(\xi)) + \delta t \|\tilde{r}^k(\xi)\|_X \right) \quad (42)$$

4.3 Effiziente Lösungsverfahren für das reduzierte System

Mit Definition 4.2.7 haben wir ein iteratives Lösungsverfahren für das System aus Anfangswertproblemen (7). Mit (41) und Definition 4.2.3 können wir die Fehlerschätzungen aus Definition 4.2.7 und Definition 3.0.1 bestimmen. Diese Verfahren sind jedoch noch von Matrizen in $\mathbb{R}^{d \times d}$ abhängig. Daher wollen wir in diesem Abschnitt die in [BFN17][4.2] beschriebenen niedrig-dimensionalen Lösungsverfahren der Gleichungen (41) und (42) vorstellen und ihre Richtigkeit beweisen. Um die Lösungs- und Fehlerschätzverfahren auf den reduzierten Systemen schnell durchführen zu können, sollen möglichst viele Matrizen direkt nach der Basisreduktion im sogenannten *Offline-Schritt* berechnet werden.

Definition 4.3.1 (Offlinematrizen). Es seien $A \in \mathbb{R}^{d \times d}$, $g \in \mathbb{R}^d$ mit zeitabhängigen affinen Repräsentationen wie in Definition 4.2.2. Sei $(X_r(t^k))_{k=0}^K$ ein Folge von Vektorräumen mit den entsprechenden Konstruktionen wie in Definition 2.2.2 für alle Zeitschritte $0 \leq k \leq K \in \mathbb{N}$. Seien $U_m \in \mathbb{R}^{m \times d}$ und $P_m \in \mathbb{R}^{d \times m}$ die nach der EIM-Approximation aus Definition 4.1.3 errechneten Matrizen. Dann definieren wir nach [BFN17][4.2.2] die *Offlinematrizen*

$$\begin{aligned} K_{ij}^{1,k} &= V_r^{kT} A^{i,k} \Pi_{X_r^\perp}^k A^{j,k} V_r^k, & K_i^{2,k} &= \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1} A^{i,k+1} V_r^{k+1}, \\ K^{3,k} &= \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^{kT}}{\delta t}, & K^{4,k} &= \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1} U_m, \\ K_i^{5,k} &= V_r^{kT} A^{i,k} \Pi_{X_r^\perp}^k U_m \end{aligned} .$$

und die Vektoren

$$b_i^{1,k} = \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1} g^{i,k}, \quad b_{ij}^{2,k} = V_r^{k+1T} A^{i,k+1T} \Pi_{X_r^\perp}^{k+1} g^{j,k},$$

$$b_i^{3,k} = U_m^T \Pi_{X_r^\perp}^{k+1} g^{i,k}.$$

¹⁶Vgl. Bemerkung 4.2.5.

Definition 4.3.2 (Onlinematrizen). Zu den Offlinematrizen aus Definition 4.3.2 definieren wir die *Onlinematrizen* wie in [BFN17][4.2.2] durch die Matrizen

$$M_1^k(\xi) = \sum_{i,j=1}^{Q_A} \theta_A^{i,k} \theta_A^{j,k} K_{ij}^{1,k}, \quad M_2^k(\xi) = -2 \sum_{i=1}^{Q_A} \theta_A^{i,k+1}(\xi) K_i^{2,k},$$

$$M_3^k(\xi) = K^{3,k}, \quad M_4^k(\xi) = -2K^{4,k},$$

$$M_5^k(\xi) = 2 \sum_{i=1}^{Q_A} \theta_A^{i,k}(\xi) K_i^{5,k}, \quad M_6^k = U_m^T \Pi_{X_r^\perp}^k U_m,$$

die Vektoren

$$v_1^k(\xi) = P_m^T h^k(u_r^k(\xi), \xi), \quad v_2^k(\xi) = \sum_{i=1}^{Q_g} \theta_g^{i,k}(\xi) b_i^{3,k},$$

$$v_3^k(\xi) = -2 \sum_{i=1}^{Q_g} \theta_g^{i,k}(\xi) b_i^{1,k}, \quad v_4^k(\xi) = 2 \sum_{i=1}^{Q_A} \sum_{j=1}^{Q_g} \theta_A^{i,k+1}(\xi) \theta_g^{j,k}(\xi) b_{ij}^{2,k}$$

und den skalaren Wert

$$b^k = \sum_{i,j=1}^{Q_g} \theta_g^{i,k}(\xi) \theta_g^{j,k}(\xi) \langle \Pi_{X_r^\perp}^{k+1} g^{i,k}, \Pi_{X_r^\perp}^{k+1} g^{j,k} \rangle_X.$$

Es sei angemerkt, dass wir für b^k die Berechnungsvorschrift aus [BFN17, 4.2.2.] nicht exakt übernehmen konnten. Sollte die dort verwendete Darstellung korrekt sein (d.h. wir können sie für die Zerlegung des Residuums wie im folgenden Satz 4.3.3 verwenden), so stimmt sie auch mit der hier benutzten überein (siehe die Erläuterung zu (53)).

Satz 4.3.3 (Zerlegung des Residuums). *Sei $0 \leq k \leq K$ und $\tilde{r}^k(\xi)$ wie in (41) definiert. Dann gilt zusammen mit den Onlinematrizen aus Definition 4.3.2:*

$$\begin{aligned} & \|\tilde{r}^k(\xi)\|_X^2 \\ &= \langle \alpha_r^{k+1}(\xi), M_1^{k+1}(\xi) \alpha_r^{k+1}(\xi) \rangle_X + \langle \alpha_r^k(\xi), M_2^k(\xi) \alpha_r^{k+1}(\xi) \rangle_X \\ &+ \langle \alpha_r^k(\xi), M_3^k(\xi) \alpha_r^k(\xi) \rangle_X + \langle \alpha_r^k(\xi), M_4^k(\xi) v_1^k(\xi) \rangle_X \\ &+ \langle \alpha_r^{k+1}, M_5^{k+1}(\xi) v_1^k(\xi) \rangle_X + \langle v_1^k(\xi), M_6 v_1^k(\xi) \rangle_X \\ &+ \langle v_1^k(\xi), v_2^k(\xi) \rangle_X + \langle \alpha_r^k(\xi), v_3(\xi) \rangle_X + \langle \alpha_r^{k+1}(\xi), v_4^k(\xi) \rangle_X + b^k \end{aligned} \quad (43)$$

Beweis. Da der Parameter ξ in der Aussage nicht variiert, verzichten wir für den Beweis darauf. Nach Definition 2.1.1 ist die Norm $\|\cdot\|_X$ durch das Skalarprodukt $\langle \cdot, \cdot \rangle_X$ induziert. Es gilt

$$\|\tilde{r}^k\|_X^2 = \langle \tilde{r}^k, \tilde{r}^k \rangle. \quad (44)$$

Wir möchten im Folgenden eine Berechnungsvorschrift für $\|\tilde{r}^k\|_X^2$ erhalten. Zuerst teilen wir das Residuum in vier Teile auf, um sehr lange Terme zu vermeiden:

$$\tilde{r}^k = \rho_1^k \alpha_r^k - \rho_2^k u_r^{k+1} - \rho_3^k \tilde{h}^k(u_r^k, \xi) - \rho_3^k g^k \quad (45)$$

mit $\rho_1^k = \Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t}$, $\rho_2^k = \Pi_{X_r^\perp}^{k+1} A^{k+1}$, $\rho_3^k = \Pi_{X_r^\perp}^{k+1}$.
Einsetzen in (44) liefert

$$\begin{aligned} & \|\tilde{r}^k\|_X^2 \\ &= \langle \rho_1^k \alpha_r^k, \rho_1^k \alpha_r^k \rangle_X + \langle \rho_2^k u_r^{k+1}, \rho_2^k u_r^{k+1} \rangle_X \\ &+ \langle \rho_3^k \tilde{h}^k(u_r^k, \xi), \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X + \langle \rho_3^k g^k, \rho_3^k g^k \rangle_X \\ &- 2 \langle \rho_1^k \alpha_r^k, \rho_2^k u_r^{k+1} \rangle_X \\ &- 2 \langle \rho_1^k \alpha_r^k, \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X - 2 \langle \rho_1^k \alpha_r^k, \rho_3^k g^k \rangle_X \\ &+ 2 \langle \rho_2^k u_r^{k+1}, \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X \\ &+ 2 \langle \rho_2^k u_r^{k+1}, \rho_3^k g^k \rangle_X + 2 \langle \rho_3^k \tilde{h}^k(u_r^k, \xi), \rho_3^k g^k \rangle_X. \end{aligned}$$

Hierbei wurde benutzt, dass X ein \mathbb{R} -Vektorraum ist und somit $\langle x, y \rangle_X = \langle y, x \rangle_X$ für alle $x, y \in X$ gilt.

Wir möchten das linke Argument der Skalarprodukte unabhängig von ρ_1^k, ρ_2^k und ρ_3^k darstellen. Hierzu nutzen wir die Eigenschaft $\langle Ax, y \rangle = \langle x, A^T y \rangle$ der zu A adjungierten Abbildung.¹⁷ Den approximierten nichtlinearen Fluss \tilde{h}^k ersetzen wir gemäß (29). Wir erhalten die Gleichung:

$$\begin{aligned} & \|\tilde{r}^k\|_X^2 = \langle \tilde{r}^k, \tilde{r}^k \rangle_X \\ &= \langle \alpha_r^k, \rho_1^{kT} \rho_1^k \alpha_r^k \rangle_X + \langle u_r^{k+1}, \rho_2^{kT} \rho_2^k u_r^{k+1} \rangle_X \\ &+ \langle \tilde{h}^k(u_r^k, \xi), \rho_3^{kT} \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X + \langle g^k, \rho_3^{kT} \rho_3^k g^k \rangle_X \\ &- 2 \langle \alpha_r^k, \rho_1^{kT} \rho_2^k u_r^{k+1} \rangle_X \\ &- 2 \langle \alpha_r^k, \rho_1^{kT} \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X - 2 \langle \alpha_r^k, \rho_1^{kT} \rho_3^k g^k \rangle_X \\ &+ 2 \langle u_r^{k+1}, \rho_2^{kT} \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X \\ &+ 2 \langle u_r^{k+1}, \rho_2^{kT} \rho_3^k g^k \rangle_X + 2 \langle \tilde{h}^k(u_r^k, \xi), \rho_3^{kT} \rho_3^k g^k \rangle_X \end{aligned} \quad (46)$$

Wir setzen die Definitionen für $\rho_1^k, \rho_2^k, \rho_3^k$ ein und betrachten die einzelnen Summanden. Hierbei nutzen wir die Eigenschaften aus Lemma 2.2.3 für die orthogonale

¹⁷Für reelle Matrizen ist die adjungierte Matrix gleich der transponierten Matrix.

Projektion $\Pi_{X_r^\perp}^{k+1}$.

$$\begin{aligned}
& \langle \rho_2^k u_r^{k+1}, \rho_2^k u_r^{k+1} \rangle_X \\
&= \langle u_r^{k+1}, \rho_2^{kT} \rho_2^k u_r^{k+1} \rangle_X \\
&= \langle u_r^{k+1}, A^{k+1T} \Pi_{X_r^\perp}^{k+1T} \Pi_{X_r^\perp}^{k+1} A^{k+1} u_r^{k+1} \rangle_X \\
&= \langle V_r^{k+1} \alpha_r^{k+1}, A^{k+1T} \Pi_{X_r^\perp}^{k+1} A^{k+1} V_r^{k+1} \alpha_r^{k+1} \rangle_X \\
&= \langle \alpha_r^{k+1}, V_r^{k+1T} A^{k+1T} \Pi_{X_r^\perp}^{k+1} A^{k+1} V_r^{k+1} \alpha_r^{k+1} \rangle_X \\
&= \sum_{i,j=1}^{Q_A} \theta_A^{i,k+1} \theta_A^{j,k+1} \langle \alpha_r^{k+1}, V_r^{k+1T} A^{i,k+1T} \Pi_{X_r^\perp}^{k+1} A^{j,k+1} V_r^{k+1} \alpha_r^{k+1} \rangle_X \\
&= \sum_{i,j=1}^{Q_A} \theta_A^{i,k+1} \theta_A^{j,k+1} \langle \alpha_r^{k+1}, K_{ij}^{1,k+1} \alpha_r^{k+1} \rangle_X = \langle \alpha_r^{k+1}, M_1^{k+1} \alpha_r^{k+1} \rangle_X \quad (47)
\end{aligned}$$

$$\begin{aligned}
& -2 \langle \alpha_r^k, \rho_1^{kT} \rho_2^k u_r^{k+1} \rangle_X \\
&= -2 \langle \alpha_r^k, \frac{\delta V_r^k}{\delta t} \Pi_{X_r^\perp}^{k+1T} \Pi_{X_r^\perp}^{k+1} A^{k+1} u_r^{k+1} \rangle_X \\
&= -2 \sum_{i=1}^{Q_A} \theta_A^{i,k+1} \langle \alpha_r^k, \frac{\delta V_r^k}{\delta t} \Pi_{X_r^\perp}^{k+1} A^{i,k+1} V_r^{k+1} \alpha_r^{k+1} \rangle_X \\
&= -2 \sum_{i=1}^{Q_A} \theta_A^{i,k+1} \langle \alpha_r^k, K_i^{2,k} \alpha_r^{k+1} \rangle_X \\
&= \langle \alpha_r^k, M_2^k \alpha_r^{k+1} \rangle_X \quad (48)
\end{aligned}$$

$$\begin{aligned}
& \langle \alpha_r^k, \rho_1^{kT} \rho_1^k \alpha_r^k \rangle_X \\
&= \langle \alpha_r^k, \left(\Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t} \right)^T \left(\Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t} \right) \alpha_r^k \rangle_X \\
&= \langle \alpha_r^k, \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1T} \Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t} \alpha_r^k \rangle_X \\
&= \langle \alpha_r^k, \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1T} \frac{\delta V_r^k}{\delta t} \alpha_r^k \rangle_X \\
&= \langle \alpha_r^k, K^{3,k} \alpha_r^k \rangle_X = \langle \alpha_r^k, M_3^k \alpha_r^k \rangle_X \quad (49)
\end{aligned}$$

$$\begin{aligned}
& -2 \langle \alpha_r^k, \rho_1^{kT} \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X \\
&= -2 \langle \alpha_r^k, \left(\Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t} \right)^T \Pi_{X_r^\perp}^{k+1} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= -2 \langle \alpha_r^k, \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1T} \Pi_{X_r^\perp}^{k+1} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= -2 \langle \alpha_r^k, \frac{\delta V_r^{kT}}{\delta t} \Pi_{X_r^\perp}^{k+1T} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= -2 \langle \alpha_r^k, K^{4,k} v_1^k \rangle_X = \langle \alpha_r^k, M_4^k v_1^k \rangle_X \quad (50)
\end{aligned}$$

$$\begin{aligned}
& 2\langle u_r^{k+1}, \rho_2^k \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X \\
&= 2\langle V_r^{k+1} \alpha_r^{k+1}, \left(\Pi_{X_r^\perp}^{k+1} A^{k+1} \right)^T \Pi_{X_r^\perp}^{k+1} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= 2\langle \alpha_r^{k+1}, V_r^{k+1 T} A^{k+1 T} \Pi_{X_r^\perp}^{k+1 T} \Pi_{X_r^\perp}^{k+1} U_m v_1^k \rangle_X \\
&= 2 \sum_{i=1}^{Q_A} \theta_A^{i,k+1} \langle \alpha_r^{k+1}, V_r^{k+1 T} A^{i,k+1 T} \Pi_{X_r^\perp}^{k+1} U_m v_1^k \rangle_X \\
&= 2 \sum_{i=1}^{Q_A} \theta_A^{i,k+1} \langle \alpha_r^{k+1}, K_i^{5,k+1} v_1^k \rangle_X = \langle \alpha_r^{k+1}, M_5^{k+1} v_1^k \rangle_X
\end{aligned} \tag{51}$$

$$\begin{aligned}
& \langle \tilde{h}^k(u_r^k, \xi), \rho_3^k \tilde{h}^k(u_r^k, \xi) \rangle_X \\
&= \langle U_m P_m^T h^k(u_r, \xi), \Pi_{X_r^\perp}^{k+1 T} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= \langle P_m^T h^k(u_r, \xi), U_m^T \Pi_{X_r^\perp}^{k+1} U_m P_m^T h^k(u_r, \xi) \rangle_X \\
&= \langle v_1^k, M_6^k v_1^k \rangle_X
\end{aligned} \tag{52}$$

Die folgende Gleichung verursacht eine Abweichung zu der in [BFN17, 4.2.2.] aufgestellten Berechnungsvorschrift für b^k . Um diese Behauptung zu zeigen, ist es erforderlich, dass die Abbildung $\Pi_{X_r^\perp}^{k+1}$ die Norm von g^k für alle $0 \leq k \leq K$ erhält. Da wir keinen Grund gefunden haben dies anzunehmen, benutzen wir die hier gezeigte Darstellung.¹⁸ Sollte die Darstellung aus [BFN17, 4.2.2.] doch mit der Zerlegung des Residuums kompatibel sein, so ist unsere Darstellung nicht falsch, sondern kann nur weiter vereinfacht werden.

$$\begin{aligned}
& \langle \rho_3^k g^k, \rho_3^k g^k \rangle_X \\
&= \langle \Pi_{X_r^\perp}^{k+1} g^k, \Pi_{X_r^\perp}^{k+1} g^k \rangle_X = \sum_{i,j=1}^{Q_g} \theta_g^{i,k} \theta_g^{j,k} \langle \Pi_{X_r^\perp}^{k+1} g^{i,k}, \Pi_{X_r^\perp}^{k+1} g^{j,k} \rangle_X \\
&= b^k
\end{aligned} \tag{53}$$

$$\begin{aligned}
& 2\langle \tilde{h}^k(u_r^k, \xi), \rho_3^k \rho_3^k g^k \rangle_X \\
&= 2\langle U_m P_m^T h^k(u_r, \xi), \Pi_{X_r^\perp}^{k+1 T} \Pi_{X_r^\perp}^{k+1} g^k \rangle_X \\
&= 2 \sum_{i=1}^{Q_g} \theta_g^{i,k} \langle v_1^k, U_m^T \Pi_{X_r^\perp}^{k+1} g^{i,k} \rangle_X = 2 \sum_{i=1}^{Q_g} \theta_g^{i,k} \langle v_1^k, b_i^{3,k} \rangle_X \\
&= \langle v_1^k, v_2^k \rangle_X
\end{aligned} \tag{54}$$

¹⁸Dies hat keinen Einfluss auf die durchgeführten Experimente, da es sich bei den dortigen Problemen nur um homogene Differentialgleichungen handelt, also $g \equiv 0$. Die Größe b^k spielt nach Bemerkung 4.3.4 keine Rolle.

$$\begin{aligned}
& -2\langle \alpha_r^k, \rho_1^k \rho_3^k g^k \rangle_X \\
& = -2\langle \alpha_r^k, \left(\Pi_{X_r^\perp}^{k+1} \frac{\delta V_r^k}{\delta t} \right)^T \Pi_{X_r^\perp}^{k+1} g^k \rangle_X \\
& = -2\langle \alpha_r^k, \frac{\delta V_r^k}{\delta t} \Pi_{X_r^\perp}^{k+1 T} \Pi_{X_r^\perp}^{k+1} g^k \rangle_X \\
& = -2 \sum_{i=1}^{Q_g} \theta_g^{i,k} \langle \alpha_r^k \frac{\delta V_r^k}{\delta t} \Pi_{X_r^\perp}^{k+1} g^{i,k} \rangle_X = -2 \sum_{i=1}^{Q_g} \theta_g^{i,k} \langle \alpha_r^k, b_i^{1,k} \rangle_X \\
& = \langle \alpha_r^k, v_3^k \rangle_X
\end{aligned} \tag{55}$$

$$\begin{aligned}
& 2\langle u_r^{k+1}, \rho_2^k \rho_3^k g^k \rangle_X \\
& = 2\langle V_r^{k+1} \alpha_r^{k+1}, \left(\Pi_{X_r^\perp}^{k+1} A^{k+1} \right)^T \Pi_{X_r^\perp}^{k+1} g^k \rangle_X \\
& = 2 \sum_{i=1}^{Q_A} \sum_{j=1}^{Q_g} \theta_A^{i,k+1} \theta_g^{j,k} \langle \alpha_r^{k+1}, V_r^{k+1 T} A^{i,k+1 T} \Pi_{X_r^\perp}^{k+1 T} \Pi_{X_r^\perp}^{k+1} g^{j,k} \rangle_X \\
& = 2 \sum_{i=1}^{Q_A} \sum_{j=1}^{Q_g} \theta_A^{i,k+1} \theta_g^{j,k} \langle \alpha_r^{k+1}, b_{ij}^{2,k} \rangle_X = \langle \alpha_r^{k+1}, v_4^k \rangle_X
\end{aligned} \tag{56}$$

Wir haben für alle Summanden in (46) die Gleichheit mit einem Term hergeleitet, der sich aus den Onlinematrizen aus Definition 4.3.2 zusammensetzt. Einsetzen der Ergebnisse von (47), (48), (49), (50), (51), (52) und (53), (54), (55), (56) in (46) ergibt genau die Gleichung (43). \square

Bemerkung 4.3.4. (i) Liegt wie in Abschnitt 5 ein homogenes Problem mit ausschließlich linearen Differentialoperatoren vor, so können wir auf die meisten Matrizen aus den Definitionen 4.3.1 und 4.3.2 verzichten. Gilt $g \equiv 0$ und $h \equiv 0$, so bleiben nur die Matrizen K^1, K^2, K^3 in der Offlinephase und demnach M_1, M_2, M_3 in der Onlinephase für jeden Zeitschritt.

(ii) Wir können die Offline- und Onlineberechnungen für die Fehlerabschätzung im T-Greedy wie auch im POD-Greedy verwenden. Im POD-Greedy verringert sich das Volumen zu speichernder Werte immens. Da die Basen der reduzierten Räume in diesem Fall nicht zeitabhängig sind, bleiben nur K^1, K^5, b^2 und b^3 in der Offlinephase sowie $M_1, M_5, M_6, v_1, v_2, v_4$ und b in der Onlinephase und das nur für einen Zeitschritt.

Bemerkung 4.3.5. Aufgrund der Ergebnisse in den deterministischen Tests (siehe Abschnitt 5.2) wurden sich einige Strategien zum Ausgleichen der als negativ berechneten Normen des Residuums überlegt. Diese stellen wir hier kurz mit dem zugehörigen Kürzel vor. Für die Berechnung nach (43) verwenden wir das Kürzel *fast*.

- *slow*: Berechnung des Residuums nach (41). Die oben beschriebenen Probleme tauchen dabei nicht auf, jedoch macht sich dieses Verfahren nicht die Offline-/Online-Zerlegung zunutze.

- *vol*: Unter der Annahme, dass die negativen Werte für (43) zufällig verteilt sind (z.B. durch Rundungsfehler), schätzen wir den richtigen Wert für (42) statistisch ab. Hierzu bestimmen wir die Rate negativer Werte $\omega(\xi) := \frac{\#\{k: \|\tilde{r}^k(\xi)\|_X^2 < 0\}}{K}$ und berechnen dann $\tilde{\Delta}_r^{(0,T)}$, wobei negative Werte für (43) durch 0 ersetzt werden.¹⁹ Dann bestimmen wir

$$\tilde{\Delta}_r^{\text{vol}}(\xi) := \frac{\tilde{\Delta}_r^{(0,T)}(\xi)}{1 - \omega(\xi)}.$$

- *abs*: Die auftretenden negativen Werte für (43) werden einfach normiert. Der Fehler wird dann mit (42) berechnet.

Bemerkung 4.3.6 (Numerisch stabile Berechnung des Residuums). Wie bereits angemerkt zeigt das Verfahren zur Berechnung des Residuums aus Satz 4.3.3 unerwartete Ergebnisse (siehe Abschnitt 5.2). Für eben solche Stabilitätsaspekte in der Offline-/Online-Zerlegungen wurde in [BEOR14, 4.2] ein numerisch stabiles Berechnungsverfahren vorgestellt. Jedoch wurde diese Zerlegung nur für zeitunabhängige Residuen gezeigt, nicht für solche wie in (46). Daher weisen wir an dieser Stelle nur auf die Existenz dieses stabilen Verfahrens hin.

5 Numerische Experimente

Es wurde versucht, die Tests aus [BFN17][5.1.] mit den dort angegebenen Spezifikationen möglichst genau nachzustellen um so die dort gezeigten Ergebnisse zu reproduzieren. Zusätzlich präsentieren wir einige weitere mit den Greedy-Verfahren aus dem System (57) errechnete Daten. Die Experimente in [BFN17] wurden mittels Matlab realisiert (siehe [BFN17][Ende Abschnitt 5.]). Wir möchten die Ergebnisse auf anderem Wege erhalten und verwenden hierzu Python in der Version 3.7.2 sowie einige quelloffene Bibliotheken. Sämtlicher eigener Code wurde auf Linux-Systemen getestet. Weitere Informationen zum Programm befinden sich im Anhang, ebenso der kommentierte und teilweise dokumentierte Quellcode.

Der Hauptteil der Implementation basiert auf pyMOR. Diese Bibliothek enthält Objekte und effiziente Algorithmen für die Reduktion parameterabhängiger Systeme. Insbesondere werden Routinen zur Realisierung von Greedy-Verfahren auf diskretisierten dynamischen Systemen bereitgestellt. Die numerischen Löser und Klassen zum Durchführen der RB-Verfahren der eigenen Implementation sind gegen die Schnittstellen dieser Funktion programmiert. Eine Ausführung über die mathematischen Hintergründe und technischen Erwägungen sowie eine umfangreiche Diskussion der Architektur und durchgeföhrter Tests findet sich in der Publikation der Entwickler (siehe [MRF16]).

Es sei noch angemerkt, dass uns weder der Matlab-Code der Experimente aus [BFN17], noch über die dort geschilderten Implementations- und Versuchsbeschreibungen hinausgehende Informationen zur genauen Realisierung dieser Tests vorliegen. Daher musste die Genauigkeit diverser numerischer Verfahren nach eigenem Ermessen gewählt werden. Dies umschließt unter anderem die Toleranz der Orthogonalisierungsverfahren und die Genauigkeit der Lösungsverfahren für lineare Gleichungssysteme.

¹⁹Der Ausdruck $\|\cdot\|_X^2 < 0$ ist natürlich im Sinne der Berechnung nach (43) zu verstehen, da für reelle Zahlen natürlich weder die Norm noch das Quadrat negativ sein können.

5.1 Untersuchung der Problemstellung:

Wir betrachten die lineare Transportgleichung

$$\frac{d}{dt}u(t, x, \xi) - a(\xi) \cdot \frac{d}{dx}u(t, x, \xi) = 0 \quad (57)$$

mit Geschwindigkeitsfunktion $a(\xi) = 1 + \frac{1}{2}\xi$ auf $\Omega = (0, 1) \subset \mathbb{R}$ und $I = (0, 0.2)$. Die Parametermenge Ξ_{train} bestehe aus gleichmäßig verteilten Werten auf dem Intervall $[-1, 1]$. Wir berechnen die Lösungen zu den Parameter-unabhängigen Anfangsbedingungen u_{cont}^0 und u_{disc}^0 , wobei

$$u_{cont}^0(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\left(\frac{x - 0.6}{0.05}\right)^2\right) \quad (58)$$

und

$$u_{disc}^0(x) = \mathbb{1}_{[0.1, 0.9]}(x) \cdot (\lfloor 3x \rfloor + \sin(10x))^2. \quad (59)$$

20

Wir diskretisieren die Domäne Ω mit $d = 2001$ Punkten auf dem Gitter $\{x_i\}_{i=0}^{d-1}$, wobei $x_i = i \cdot \delta x$ und $\delta x = \frac{1}{d-1}$. Dadurch erhalten wir ein System aus d Anfangswertproblemen der Form

$$\frac{d}{dt}u_i(t, \xi) + a(\xi) \frac{d}{dx}u_i(t, \xi) = 0 \quad (60)$$

mit Anfangswerten $u_i(0, \xi) = u(0, x_i, \xi)$ und für $1 \leq i \leq d$.

Zur Diskretisierung des Zeitintervalls benutzen wir wie auch [BFN17] ein Finite-Differenzen-Schema wie in [Krö97, 2.2.7] beschrieben. In [BFN17, 5.1.] wird das Verfahren aus Definition 5.1.1 auch *finite difference upwind scheme* genannt.

Definition 5.1.1 (Engquist-Osher-Schema). Sei

$$f(u, t, \xi) = a \cdot \frac{d}{dx}u(t, xi)$$

der Fluss eines dynamischen Systems wie Definition 1. Für $a < 0$ diskretisieren wir die Zeitentwicklung mit dem expliziten Schema (vgl. [Krö97, (2.3.55)])

$$u_i^{k+1}(\xi) = u_i^k(\xi) - \frac{\delta t}{\delta x}a(\xi)(u_{i+1}^k(\xi) - u_i^k(\xi)) \quad (61)$$

und dem impliziten Schema (vgl. [Krö97, (2.3.56)])

$$u_i^{k+1}(\xi) = u_i^k(\xi) - \frac{\delta t}{\delta x}a(\xi)(u_{i+1}^{k+1}(\xi) - u_i^{k+1}(\xi)) \quad (62)$$

zu der Anfangsbedingung $u^0 \in \mathbb{R}^d$.

²⁰ $\lfloor x \rfloor$ ist der abgerundete Wert von $x \in \mathbb{R}$. $\mathbb{1}_{[a,b]}$ ist die Indikatorfunktion auf dem Intervall $[a, b] \subset \mathbb{R}$.

Bemerkung 5.1.2. Durch vektorwertige Betrachtung der beiden Schemen (61) und (62) erhalten wir analog zu [BFN17, (28)]

$$u^{k+1}(\xi) = (I_d + \delta t C(\xi)) u^k(\xi) \quad (63)$$

und

$$u^{k+1}(\xi) = (I_d - \delta t C(\xi))^{-1} u^k(\xi) \quad (64)$$

mit diskretem Advektionsoperator

$$C(\xi) = \frac{a(\xi)}{\delta x} \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & & -1 \end{pmatrix} \in \mathbb{R}^{d \times d} \quad (65)$$

mit Einträgen ungleich 0 auf der Diagonalen und der ersten oberen Nebendiagonalen der Matrix. Wir fordern, dass der Eintrag in der ersten Spalte der letzten Zeile gleich 1 ist, um die in [BFN17, 5.1.] geforderte Periodizität der Lösung sicherzustellen.

Bemerkung 5.1.3. Konkret wählen wir die Zeitschrittweite $\delta t = \frac{1}{3} \delta x$. In [BFN17, 5.1] wird dafür $\delta t = 0.5 \frac{\delta x}{c_0 + c_1}$ angegeben. Die Konstanten c_0 und c_1 werden im gesamten Paper nicht definiert oder anders verwendet. Jedoch erhalten wir den Wert $\frac{1}{1200}$, wenn wir statt c_0, c_1 die Konstanten a_0, a_1 aus der Definition der Geschwindigkeit $a(\xi)$ einsetzen. Die Grafiken [BFN17, Fig. 5, Fig. 6] legen zusätzlich nahe, dass $K = 1200$ gelten muss.

Bemerkung 5.1.4. Wir können die beiden Zeitschrittverfahren aus Definition 5.1.1 zu

$$\frac{u_i^{k+1}(\xi) - u_i^k(\xi)}{\delta t} = a(\xi) \frac{u_i^k(\xi) - u_{i-1}^k(\xi)}{\delta x} \quad (66)$$

und

$$\frac{u_i^{k+1}(\xi) - u_i^k(\xi)}{\delta t} = a(\xi) \frac{u_i^{k+1}(\xi) - u_{i-1}^{k+1}(\xi)}{\delta x} \quad (67)$$

umschreiben.

Die rechten Seiten von (66) und (67) entsprechen jeweils dem Fluss f des diskreten dynamischen Systems der Dimension d .

Korollar 5.1.5. Das Problem (57) besitzt für alle $0 \leq k \leq K$ und $\xi \in \Xi$ eine eindeutige Lösung in X .

Beweis. Der Fluss f des diskreten dynamischen Systems voller Dimension ist gegeben durch die rechte Seite von (66).²¹ Wir wollen induktiv zeigen, dass dieser Fluss den Bedingungen an Satz 2.1.3 für alle $0 \leq k \leq K$ genügt. Seien $u, v \in X$ und $k = 0$:

²¹Der Beweis erfolgt für das implizite Verfahren analog. Wir zeigen die Aussage daher nur für das explizite Schema.

Dann gilt

$$\begin{aligned}\|f(u, t^0, \xi) - f(v, t^0, \xi)\|_X &= \|a(\xi) \frac{u_i^0 - v_i^0 - (u_{i-1}^0 - v_{i-1}^0)}{\delta x}\|_X \\ &\leq \|a(\xi) \frac{u_i^0 - v_i^0}{\delta x}\|_X + \|a(\xi) \frac{u_{i-1}^0 - v_{i-1}^0}{\delta x}\|_X \\ &\leq \frac{2a(\xi)}{\delta x} \|u - v\|_X\end{aligned}$$

Wir setzen $L_0 := \frac{2a(\xi)}{\delta x}$. Der Fluss f erfülle nun die Lipschitzbedingung für alle $0 \leq k' \leq k$ für ein $k < K$ mit einer Konstante $L_{k'}$. Wir zeigen nun, dass die Lipschitzbedingung dann auch für $k+1$ erfüllt ist. Dazu setzen wir die Gleichung (61) ein:

$$\begin{aligned}\|f(u, t^{k+1}, \xi) - f(v, t^{k+1}, \xi)\|_X &= \|a(\xi) \frac{u_i^{k+1} - v_i^{k+1} - (u_{i-1}^{k+1} - v_{i-1}^{k+1})}{\delta x}\|_X \\ &= \left\| \frac{a(\xi)}{\delta x} \left(u_i^k - v_i^k - (u_{i-1}^k - v_{i-1}^k) \right. \right. \\ &\quad \left. \left. - \frac{\delta t a(\xi)}{\delta x} (u_i^k(\xi) - v_i^k(\xi) - (u_{i-1}^k(\xi) - v_{i-1}^k(\xi))) \right) \right\|_X \\ &\leq \|f(u, t^k, \xi) - f(v, t^k, \xi)\|_X + \frac{\delta t a(\xi)}{\delta x} \|f(u, t^k, \xi) - f(v, t^k, \xi)\|_X \\ &\leq L_k \left(1 + \frac{\delta t a(\xi)}{\delta x} \right) \|u - v\|_X < \infty\end{aligned}$$

□

Die folgende Aussage enthält eine Bedingung für die Stabilität des expliziten Upwind-Schemas (66). Als Spezialfall folgt daraus auch die Stabilität des expliziten Schemas für $d = 2001$.

Korollar 5.1.6 (Erfüllung der CFL-Bedingung). *Die Courant-Friedrichs-Lowy-Bedingung (kurz CFL-Bedingung) ist für das Problem (57) und das explizite Schema (66) durch*

$$\frac{\delta t}{\delta x} a \leq 1 \tag{68}$$

erfüllt. Hierbei ist $a = \max_{\xi \in \Xi} a(\xi)$ und $\delta t = \frac{1}{3} \delta x$ für $\delta x = \frac{1}{d-1}$ für $2 \leq d \in \mathbb{N}$. Insbesondere ist das Verfahren (66) stabil für $d = 2001$.

Beweis. Wir stellen zuerst fest, dass für alle $\xi \in \Xi$

$$\frac{\delta t}{\delta x} a(\xi) \leq \frac{\delta t}{\delta x} a = \leq 1$$

gilt. Falls die Aussage für a gilt, so auch für alle $\xi \in \Xi$. Das Schema (66) ist nach [Krö97, 2.3.16] monoton (Vgl. [Krö97, 2.2.7]), genügt somit den Bedingungen an den Satz [Krö97, 2.3.19]. Die Stabilität folgt dann direkt aus dem in [Krö97, 2.3.22] beschriebenen Spezialfall linearer Probleme und expliziter Schemen des Satzes [Krö97, 2.3.19]. □

	E_2	E_∞
u_{cont}^0	$4.54 \cdot 10^{-1}$	$2.25 \cdot 10^{-2}$
u_{disc}^0	$3.62 \cdot 10^{-1}$	$1.55 \cdot 10^{-2}$

Tabelle 1: Relative Fehler der einmaligen Basiserweiterung anhand zeitabhängiger Basisvektoren für stetige (u_{cont}^0) und unstetige (u_{disc}^0) Anfangsbedingungen. Berechnung der vollen Lösung zu Parameter $\xi = 0.65$ mit explizitem Eulerverfahren.

	E_2	E_∞
u_{cont}^0	$2.29 \cdot 10^{-12}$	$1.04 \cdot 10^{-13}$
u_{disc}^0	$6.16 \cdot 10^{-13}$	$4.29 \cdot 10^{-14}$

Tabelle 2: Relative Fehler der einmaligen einmaligen Basiserweiterung anhand zeitabhängiger Basisvektoren für stetige (u_{cont}^0) und unstetige (u_{disc}^0) Anfangsbedingungen. Berechnung der vollen Lösung zu Parameter $\xi = 0.65$ mit implizitem Eulerverfahren.

Bemerkung 5.1.7. Anders als in der Vorlage [BFN17, 5.1.] betrachten wir im folgenden deterministischen Test zusätzlich das implizite Eulerverfahren wie (67). Wie an den Ergebnissen dieses Versuchs zu erkennen ist, eignet sich das explizite Eulerverfahren nicht für unsere Umsetzung der Algorithmen 1 und 2.

5.2 Deterministischer Fall

Es soll die Genauigkeit der Basisreduktion und des reduzierten Zeitschrittverfahrens aus Definition 4.1.4 für einen festen Parameter überprüft werden. Hierfür wählen wir $\xi^0 = 0.65$, erzeugen die reduzierte Basis wie in Definition (25) und vergleichen den durch POD-Greedy und T-Greedy produzierten relativen Fehler

$$E_q = \frac{\|u_r - u\|_{I,q}}{\|u\|_{I,q}} \text{ für } q \in \{2, \infty\}. \quad (69)$$

Hierbei sind die Normen $\|\cdot\|_{I,q}$ analog zu Definition 3.0.1 für $u \in X$ und $\xi \in \Xi$ durch

$$\|u\|_{I,2} := \left(\int_I \|u(t, \xi)\|_X^2 dt \right)^{\frac{1}{2}} \quad (70)$$

und

$$\|u\|_{I,\infty} := \left(\sup_{t \in I} \|u(t, \xi)\|_X^2 \right)^{\frac{1}{2}} \quad (71)$$

definiert. Da wir eine diskrete Zeitdomäne betrachten, reduzieren sich die L^2 - und L^∞ -Normen zur euklidischen Norm bzw. zur Maximumsnorm.²²

²²Die Berechnungen der Normen wurden konkret anhand der Funktion `relative_error` im Skript `analysis.py` durchgeführt. Für die Norm (71) wurde auf das Quadrieren und Wurzelziehen verzichtet. Dies verspricht eine bessere numerische Stabilität.

Wir erkennen, dass die T-Greedy-Reduktion mit expliziten Eulerverfahren für stetige wie unstetige Anfangsbedingungen zwar keine absurd hohen relativen Fehler produziert, die erzielten Werte jedoch weit von unserer Maschinengenauigkeit $\epsilon \approx 2.22 \cdot 10^{-16}$ entfernt sind (siehe Tabelle 1). In der Entwicklung des exakten Approximationsfehlers (Abbildung 1) zeigt sich für beide Anfangsbedingungen ein monotoner Wachstum mit der Zeit. Für u_{cont}^0 scheint dieses zudem linear zu sein. Den genauen Grund für den hohen Approximationsfehler konnten wir nicht feststellen. Möglich ist, dass das explizite Verfahren (61) für das hoch-dimensionale Problem zum impliziten Verfahren (32) für das niedrig-dimensionale Problem inkonsistent ist. Ein zusätzlicher Faktor könnte die verwendete Maschinengenauigkeit sein.²³

Mit impliziter Berechnung des hoch-dimensionalen Problems erhalten wir hingegen akzeptable relative Fehler näher an der Maschinengenauigkeit (siehe Tabelle 2). Diese Werte sind vergleichbar mit denen aus [BFN17, 5.1.], jedoch sind sie um einige Zehnerpotenzen größer und wir erhalten für E_∞ kleinere Werte als für E_2 . Dieses Größenverhältnis für die relativen Fehler setzt sich in den nachfolgenden Tests auch fort. Wir können zwar nicht die Ergebnisse aus [BFN17] erklären, jedoch sollte sich ganz allgemein die 2-Norm durch die ∞ -Norm von unten abschätzen lassen. Anders als beim expliziten Verfahren wachsen die exakten Fehler beider Anfangsbedingungen nicht gleichmäßig, sondern nehmen sporadisch zu und wieder ab (siehe Abbildung 2).

Die Fehlerabschätzung *fast* (siehe Bemerkung 4.3.5) mittels Offline-/Onlinezerlegung (siehe Definitionen 4.3.1 und 4.3.2) produziert keine verwendbaren Werte. Das Problem liegt im Detail bei der Berechnung der quadrierten Norm des Residuums (siehe (43)). Eigentlich sind dafür ausschließlich nichtnegative Werte zu erwarten. Die konkrete Berechnung liefert jedoch mitunter negative Ergebnisse. Der Grund hierfür konnte nicht zufriedenstellend geklärt werden. Da ungefähr die Hälfte aller berechneten Werte für $\|\tilde{r}^k(\xi)\|_X^2$ negativ sind und die Verteilung dieser Werte zufällig erscheint (siehe Abbildung 4), vermuten wir, dass diese Fehler durch Rundungen der Summanden in (43) zustande kommen. Es sei noch bemerkt, dass im Gegensatz zum impliziten Verfahren diese Fehlerschätzung valide Werte mit explizit berechneten hoch-dimensionalen Lösungen produziert. Wir vermuten, dass die kleinen aber ausschlaggebenden Fehler der schnellen Berechnung für das implizite Verfahren dort zwar auch auftreten, aufgrund des hohen Approximationsfehlers aber nicht ins Gewicht fallen.

Den Ergebnissen zum T-Greedy-Verfahren wollen wir entsprechende Daten zur Konstruktion einer reduzierten Basis gemäß Algorithmus 1 entgegenstellen. Ähnlich wie bei der T-Greedy-Approximation sind die relativen Fehler zur explizit berechneten hoch-dimensionalen Lösung größer als in [BFN17, Table 1] (siehe Tabelle 3). Bereits ab $\dim(X_r) = 20$ scheint die maximale Genauigkeit zu u_{cont}^0 erreicht worden zu sein ($\dim(X_r) = 100$ für u_{disc}^0), da sich die Fehler danach nicht weiter verringern. Hingegen sind die Werte von E_q zur implizit berechneten Anfangsbedingung besser (siehe Tabelle 4). Jedoch erreichen wir keine so hohe Präzision wie in [BFN17, Table 1]. Die Approximation scheint sich für stetige (unstetige) Anfangsbedingung ab einer RB-Dimension von $r = 50$ ($r = 100$) auch nicht weiter zu verbessern. Zur Veranschaulichung der POD-

²³Wir konnten nicht klären mit welcher Präzision die Experimente in [BFN17] durchgeführt wurden. Die von uns verwendeten Löser für lineare Gleichungssysteme arbeiten maximal mit 64-Bit Fließkommazahlen.

u_{cont}^0			u_{disc}^0		
$\dim(X_r)$	E_2	E_∞	$\dim(X_r)$	E_2	E_∞
1	$2.94 \cdot 10^1$	$9.49 \cdot 10^{-1}$	1	$2.17 \cdot 10^1$	$8.09 \cdot 10^{-1}$
2	$2.15 \cdot 10^1$	$7.97 \cdot 10^{-1}$	2	$1.04 \cdot 10^1$	$7.21 \cdot 10^{-1}$
5	$4.51 \cdot 10^0$	$2.39 \cdot 10^{-1}$	5	$3.90 \cdot 10^0$	$5.68 \cdot 10^{-1}$
10	$7.51 \cdot 10^{-1}$	$4.34 \cdot 10^{-2}$	10	$2.11 \cdot 10^0$	$5.99 \cdot 10^{-1}$
20	$7.46 \cdot 10^{-1}$	$4.15 \cdot 10^{-2}$	20	$1.08 \cdot 10^0$	$5.32 \cdot 10^{-1}$
50	$7.46 \cdot 10^{-1}$	$4.15 \cdot 10^{-2}$	50	$9.66 \cdot 10^{-1}$	$1.10 \cdot 10^{-1}$
100	$7.46 \cdot 10^{-1}$	$4.15 \cdot 10^{-2}$	100	$9.67 \cdot 10^{-1}$	$1.09 \cdot 10^{-1}$
200	$7.46 \cdot 10^{-1}$	$4.15 \cdot 10^{-2}$	200	$9.67 \cdot 10^{-1}$	$1.09 \cdot 10^{-1}$

Tabelle 3: Relative Fehler der POD-Greedy Reduktion mit $\dim(X_r)$ POD-Modes für stetige (u_{cont}^0) und unstetige (u_{disc}^0) Anfangsbedingungen. Berechnung der vollen Lösung für den Parameter $\xi = 0.65$ mit explizitem Eulerverfahren.

Greedy-Approximation zeigt Abbildung 3 die Lösung des reduzierten Systems u_r zur finalen Zeit $T = 0.2$ für ausgewählte Dimensionen r . Im stetigen Fall (Abbildung 3 oben) erkennen wir starke Oszillationen der reduzierten Lösung bis einschließlich $r = 10$. Außerdem ist die globale Extremstelle der Lösungen dieser Dimension entlang der x-Achse verschoben und nähert sich mit zunehmender Dimension der Position der exakten Lösung an. Eine der vollen Lösung getreue Approximation erhalten wir erst ab $r = 50$. Für die unstetige Anfangsbedingung stimmt selbst die reduzierte Lösung zur Dimension $r = 100$ nicht genau mit der exakten Lösung überein (siehe Abbildung 3 unten). Vor allem irregulär wirkende Abschnitte der Kurve werden schlecht approximiert.

Weiter fällt auf, dass sämtliche Lösungen nicht vollständig mit denen aus [BFN17, 5.1.] übereinstimmen. Für die Lösung zu u_{cont}^0 in Abbildung 3 (oben) liegt der erreichte maximale Wert leicht unter dem aus [BFN17, Fig. 1. (links)]. Die Grafik zur Lösung zu u_{disc}^0 (Abbildung 3 (unten)) weist größere Unterschiede zu [BFN17, Fig. 1 (rechts)] auf. Die Lösung erreicht einen maximalen Wert von über 8 (Ungefähr 3 in [BFN17]). Diese Beobachtungen setzen sich in den folgenden Tests fort. Wir berücksichtigen dies bei unseren Vergleichen mit den Ergebnissen aus [BFN17].²⁴

5.3 Allgemeiner Fall

Wir möchten nun einen reduzierten Zustandsraum der Dimension $r = 20$ mit beiden Greedy-Verfahren aus Abschnitt 3 konstruieren. Da wir im deterministischen Experiment die explizit berechnete Lösung zum Parameter der Basiserweiterung nicht mit dem resultierenden reduzierten System rekonstruieren konnten, nutzen wir hier das implizite Eulerverfahren. Wir betrachten wieder das Problem (57), diesmal für Parameter aus einer gleichmäßig verteilten Menge $\Xi_{train} \subset [-1, 1]$ mit 30 Elementen und beginnen die Konstruktion einer re-

²⁴Zur Überprüfung der verwendeten Anfangsbedingungen siehe den im Anhang enthaltenen Quelltext: `test_1_deterministic.py` (Z. 129 ff.) für den deterministischen Fall und `test_1_greedy.py` (Z. 161 ff.) für den allgemeinen Fall.

dim(X_r)	u_{cont}^0		dim(X_r)	u_{disc}^0	
	E_2	E_∞		E_2	E_∞
1	$2.92 \cdot 10^1$	$9.46 \cdot 10^{-1}$	1	$2.15 \cdot 10^1$	$8.08 \cdot 10^{-1}$
2	$2.10 \cdot 10^1$	$7.88 \cdot 10^{-1}$	2	$1.01 \cdot 10^1$	$7.19 \cdot 10^{-1}$
5	$4.08 \cdot 10^0$	$2.20 \cdot 10^{-1}$	5	$3.26 \cdot 10^0$	$5.69 \cdot 10^{-1}$
10	$6.74 \cdot 10^{-2}$	$5.22 \cdot 10^{-3}$	10	$1.30 \cdot 10^0$	$5.77 \cdot 10^{-1}$
20	$8.44 \cdot 10^{-7}$	$8.73 \cdot 10^{-8}$	20	$2.23 \cdot 10^{-1}$	$3.78 \cdot 10^{-1}$
50	$3.51 \cdot 10^{-9}$	$1.32 \cdot 10^{-10}$	50	$5.72 \cdot 10^{-4}$	$2.67 \cdot 10^{-4}$
100	$3.51 \cdot 10^{-9}$	$1.32 \cdot 10^{-10}$	100	$3.04 \cdot 10^{-4}$	$1.32 \cdot 10^{-4}$
200	$3.51 \cdot 10^{-9}$	$1.32 \cdot 10^{-10}$	200	$3.04 \cdot 10^{-4}$	$1.32 \cdot 10^{-4}$

Tabelle 4: Relative Fehler der POD-Greedy Reduktion mit $\dim(X_r)$ POD-Modes für stetige (u_{cont}^0) und unstetige (u_{disc}^0) Anfangsbedingungen. Berechnung der vollen Lösung für den Parameter $\xi = 0.65$ mit implizitem Eulerverfahren.

duzierten Basis mit Snapshots für den Parameter $\xi^0 = -1$.²⁵ Wir werten die erhaltenen Lösungen für $\xi = 0.65$ aus. Dieser Parameter ist nicht in Ξ_{train} enthalten, es sind also größere Fehler als im deterministischen Fall zu erwarten.

Die T-Greedy-Approximation in Abbildung 5 ist bis auf die schon im deterministischen Fall angemerkt Unterschiede der Kurven vergleichbar mit [BFN17, Fig. 2].

Genau wie in [BFN17, 5.1.] scheinen beide Reduktionsmethoden die hochdimensionale Lösung zur Anfangsbedingung u_{cont}^0 originalgetreu zu approximieren (siehe Abbildung 6 oben.). Auch für u_{disc}^0 wirken die Lösungen nahezu identisch (siehe Abbildung 6 unten). Jedoch sind um $x = 0.2$ und im Intervall $[0.65, 0.8]$ leichte Abweichungen der POD-Greedy-reduzierten Lösung zu erkennen. Vergleichbare Beobachtungen sind auch in [BFN17, Fig. 3.] zu machen. In Abbildung 7 vergleichen wir die Entwicklung der relativen Fehler E_q (siehe (69)) für $q \in \{2, \infty\}$ und die stetige Anfangsbedingung, wobei $\Xi_{50} \subset [0, 1]$ 50 zufällig gewählte Parameter enthält.²⁶ Für das in [BFN17, 5.1.] so genannte *expectation* $\mathbb{E}(E_q)$ haben wir den Durchschnitt der relativen Fehler über alle Ξ_{50} angenommen:

$$\mathbb{E}(E_q) = \frac{1}{50} \left(\sum_{\xi \in \Xi_{50}} E_q(\xi) \right) \quad (72)$$

Für das Maximum $\max_{\xi \in \Xi_{50}} (E_q)$ schreiben wir kurz $\max(E_q)$.

Die Fehler der Lösungen zu u_{cont}^0 fallen mit zunehmender Basisgröße monoton ab. Die Abbildungen sind beinahe identisch zu denen in [BFN17, Fig. 4. oben]. Anders als dort sind bei uns die relativen Fehler E_2 für den T-Greedy-Algorithmus im Durchschnitt größer als der maximale relative Fehler E_∞ . Wieder beobachten wir, dass im Mittel wie auch im Maximum die Werte für $q = 2$ stets größer sind als für $q = \infty$. Tatsächlich zeigt die Abbildung eine bessere Approximation für $r = 20$ in allen vier Messungen. Dies erklärt sich

²⁵In der Versuchsbeschreibung in [BFN17, 5.1] wurde kein Parameter für die erste reduzierte Basis angegeben. Wir wählen $\xi^0 = -1$, da in dem - von unserem Programm jedoch nicht abgedeckten - Fall $X_0 = \{0\}$ die Fehlerschätzung aus Definition 3.0.1 für alle $\xi \in \Xi_{train}$ den gleichen Betrag hätte und somit nach Bemerkung 3.0.10 der erste Parameter gewählt würde.

²⁶Es wurde stets dieselbe Seed für die Zufallsauswahl der Parameter verwendet. Siehe Quelltext `analysis.py`, Zeile 12.

wohl an der Verwendung unterschiedlicher Parametermengen Ξ_{50} . Ungefähr ab $\dim(X_r) = 14$ erhalten wir durch die T-Greedy-Approximation mit allen vier Größen eine Genauigkeit, die durch die POD-Greedy-Reduktion im Mittel erst zu $r = 20$ erreicht wird. Bemerkenswert ist weiterhin, dass die relativen Fehler für $q = 2$ und $q = \infty$ jeweils gleich zu sein scheinen. Offenbar sind alle Approximationen über Ξ_{50} ähnlich gut.

Für die relativen Fehler der Lösungen zur unstetigen Anfangsbedingung u_{disc}^0 ergibt sich für das T-Greedy-Verfahren ein ähnliches Bild (siehe Abbildung 8 oben; Vgl. [BFN17, Fig. 4.] unten links). Von den vertauschten Werten für $q = 2$ und $q = \infty$ abgesehen, erhalten wir mit ansteigender Dimension der reduzierten Basis sehr ähnliche Ergebnisse. Auffällig ist der im Vergleich zum Problem mit stetiger Anfangsbedingung hohe Fehler für die finale Dimension $r = 20$. Um eine ähnlich gute Approximation für alle $\xi \in \Xi_{50}$ zu erhalten, müssten wir eine höher-dimensionale reduzierte Basis konstruieren. Für die POD-Greedy-Lösungen zeigt sich eine sehr ungenaue Approximation (siehe Abbildung 8 unten). Im Gegensatz zu [BFN17, Fig. 4. unten rechts] erreichen unsere Berechnungen keine Annäherung bis auf die erste Nachkommastelle.

In Abbildung 9 vergleichen nun noch die verschiedenen in Bemerkung 4.3.5 vorgeschlagenen Modifikationen der a-posteriori Fehlerschätzung $\tilde{\Delta}_r$. Da das POD-Greedy-Schema im Gegensatz zum T-Greedy-Schema für das explizite Zeitschrittverfahren (61) bis zur Dimension $r = 20$ orthonormalisierbare Snapshots produziert, wurden die Ergebnisse dieses Testlaufs in die Abbildung mit einbezogen. Insgesamt liefert das T-Greedy-Verfahren für alle Fehlerschätzungen und Dimensionen der reduzierten Basis eine präzisere Approximation als das POD-Greedy-Verfahren.

Im unstetigen Fall (Abbildung 9 unten) sind alle modifizierten Fehlerschätzungen gleichwertig zu $\tilde{\Delta}_r$. Es ist absehbar, dass sich der maximale Fehler über die Parameter in Ξ_{train} noch weiter verringern würde, sollte die Basisreduktion fortgesetzt werden. Zwar stagniert der maximale Fehler der POD-Greedy-Reduktion bis $r = 20$ auch nicht, jedoch ist die Verbesserung der Genauigkeit der Approximation über alle Dimensionen der reduzierten Basis hinweg sehr gering. Die Abnahme der maximalen Fehler ist im T-Greedy-Verfahren tendenziell monoton. Allein der Zustandsraum zur Dimension $r = 4$ produziert einen höheren maximalen Fehler als der vorige Zustandsraum zu $r = 3$. Die Entwicklung ist für das POD-Greedy-Schema auch nicht exakt monoton. Bemerkenswert ist noch, dass die POD-Greedy-Reduktion mit den explizit berechneten Snapshots hier leicht bessere Resultate liefert als die POD-Greedy-Reduktion mit implizit berechneten Snapshots. Da das Verhalten des expliziten Verfahrens im deterministischen Experiment (siehe Abschnitt 5.2) nicht geklärt ist, fällt es schwer, Rückschlüsse auf das Verhalten im Greedy-Verfahren zu ziehen. Wir belassen es bei der Tatsache, dass in diesem Fall - wie auch allgemein für die POD-Greedy-Approximationen zur unstetigen Anfangsbedingung - keine so gute Präzision wie mit dem T-Greedy-Algorithmus erreicht wird.

Der stetige Fall (Abbildung 9 oben) lässt mehr Folgerungen zur Qualität der modifizierten Fehlerschätzungen zu. Diese decken sich für die T-Greedy-Reduktion nur bis zu einer RB-Dimension von $r = 15$ mit der theoretisch exakten Variante *slow*. Für $r = 16$ weichen *vol* und *abs* von dieser leicht nach oben ab und für $r \geq 17$ bleiben diese maximalen Fehler dann konstant. Offenbar ist mit ungefähr 10^{-5} die maximale Genauigkeit dieser Fehlerschätzungen erreicht. Die Tatsache, dass *abs* eine schlechtere a-posteriori Fehlerschätzung als *slow*

(und zudem auch *vol*) liefert, schließt somit einen einfachen Vorzeichenfehler als Grund für das Scheitern der Berechnung des Residuums nach (46) aus (siehe Abschnitt 5.2). Der maximale Fehler der Abschätzung *slow* nimmt dagegen weiter ab und liefert bei der endgültigen RB-Dimension von $r = 20$ den mit Abstand besten Wert. Sollte die T-Greedy-Basiskonstruktion fortgesetzt werden, so ist zu erwarten, dass der durch *slow* geschätzte maximale Fehler bis auf ein den Ergebnissen aus Tabelle 2 entsprechendes Niveau sinkt.

5.4 Untersuchung der Fehlerabschätzung

Wir wollen die Richtigkeit der Fehlerabschätzung $\tilde{\Delta}_r$ überprüfen. Hierzu vergleichen wir die Zeitentwicklung von $\tilde{\Delta}_r$ mit dem exakten Fehler $\|e_r\|_X = \|u - u_r\|_X$ für $\xi = 0.65$. Für das POD-Greedy-Verfahren (Abbildung 10 unten) ergibt sich ein zu [BFN17, Fig. 5. oben rechts] vergleichbares Bild. Die Fehlerschätzung erfüllt überall die Ungleichung 16. Für die Fehler der T-Greedy-Approximation (Abbildung 10 oben) sieht das hingegen anders aus. Der exakte Fehler $\|e_r\|_X$ ist ab den ersten Zeitschritten stets größer als der geschätzte Fehler $\tilde{\Delta}_r$. Diese Beobachtung widerspricht den theoretischen Aussagen über die Fehlerschätzung in Proposition 2.3.5. Die Abbildung [BFN17, Fig. 5. oben links] weist für mehr als die Hälfte der Zeitschritte eine ähnliche Eigenschaft auf. Dort lässt sich diese Beobachtung wohl mit der Stabilität des dort verwendeten expliziten Eulerverfahrens erklären. Zwar stellt die dort erfüllte *CFL-Bedingung*²⁷ die Stabilität sicher, diese garantiert jedoch entsprechend gute Werte für die finale Zeit, nicht für Zwischenschritte. Wir benutzen zwar das implizite Verfahren zur Bestimmung der vollen Lösungen, trotzdem nehmen wir wieder Stabilitätsprobleme als Grund unserer Beobachtung an. Die Verwendung verschiedener Lösungsverfahren kann die zu [BFN17] unterschiedlichen Formen des Graphen zum exakten Fehler erklären. Trotz der Diskrepanzen in den Beobachtungen liegen die Fehler ungefähr im selben Bereich wie in [BFN17, Fig. 5. oben].

Für den Vergleich der Fehler der Lösungen zur unstetigen Anfangsbedingung in Abbildung 11 ergibt sich für das POD-Greedy-Verfahren wieder ein vergleichbares Bild zu [BFN17, Fig. 5. unten rechts]. Wie dort dazu ausgeführt, ist $\tilde{\Delta}_r$ eine zwar korrekte, jedoch sehr grobe Abschätzung für die Lösungen des POD-Greedy im stetigen wie auch unstetigen Fall. Weiterhin fällt auf, dass die angezeigten Fehler sehr hoch sind. Zur finalen Zeit ist der exakte Fehler größer als 1. Dies mag an den allgemein höheren Werten für die Lösung zur unstetigen Anfangsbedingung liegen (vgl. Abbildung 5 unten).

Der Vergleich der Fehler der durch den T-Greedy-Algorithmus approximierten Lösungen (siehe Abbildung 11 oben) weist wiederum Unterschiede zur entsprechenden Abbildung [BFN17, Fig. 5 unten links] auf. Bereits für die ersten Zeitschritte steigen exakter und geschätzter Fehler abrupt an. In [BFN17] hingegen scheinen beide Fehler eher gleichmäßig anzusteigen.²⁸ Wie schon für u_{cont}^0 ist die Nichterfüllung von Proposition 2.3.5 zu beobachten. Jedoch gilt dies nur für die erste Hälfte der Zeitschritte, danach erfüllen die Fehler die Ungleichung (16). Der finale Fehler von knapp unter 10^{-1} hat ungefähr die gleiche Größenordnung wie in [BFN17, Fig. 5 unten links] und ist ebenso kleiner als der der POD-Greedy-Approximation.

²⁷Vgl. (5.1.6)

²⁸Beachte, dass die y-Achse der Grafik eine logarithmische Skala besitzt.

Die in [BFN17, Fig. 6.] gezeigten Ergebnisse zum *effectivity index* $\kappa(t, \xi) = \frac{\tilde{\Delta}_r(t, \xi)}{\|e_r(t, \xi)\|_x}$ haben wir nicht nachgestellt. Dieser Quotient zeigt die Qualität der Fehlerabschätzung $\tilde{\Delta}_r$ an. Statt wie in [BFN17] den Mittelwert dieser Größe über die Parametermenge Ξ_{50} in seiner Entwicklung über alle Zeitschritte darzustellen, haben wir eine Abbildung dieser Entwicklung für nur einen Parameter $\xi = 0.65$ zur reduzierten Basis $r = 20$ produziert (siehe Abbildung 12). Wir beobachten, dass der Wert von κ mit zunehmendem Zeitschritt für stetige und auch unstetige Anfangsbedingungen mit der POD-Greedy-Approximation Werte nahe der 1 annimmt (siehe Abbildung 12 oben). Für den hier verwendeten Parameter $\xi = 0.65$ scheint die Fehlerabschätzung nahe dem exakten Fehler zu liegen. Hingegen nimmt κ für die T-Greedy-Approximation ausschließlich Werte unter 1 an (siehe Abbildung 12). Dies deckt sich gut mit dem Vergleich des a-posteriori Fehlers mit dem exakten Fehler in Abbildungen 10 und 11. Dort ist der exakte Fehler für sehr viele Zeitschritte größer als die Fehlerschätzung.

5.5 CPU-Zeiten

Wir möchten die Berechnungszeiten beider Verfahren für die Tests im allgemeinen Fall vergleichen. Hierzu sei angemerkt, dass die Experimente auf unterschiedlichen Maschinen durchgeführt wurden. Auf welchem Gerät die folgenden Zeiten gemessen wurden, ist mit einem entsprechenden Buchstaben gekennzeichnet.²⁹

Zuerst betrachten wir die Dauer des gesamten T-Greedy-Verfahrens bis zu einer Basiserweiterung der Dimension $r = 20$ mit den entsprechenden Zeiten für das POD-Greedy-Verfahren (siehe Tabelle 5; Spalten Offline). Im Vergleich zum T-Greedy geschieht die Basiskonstruktion mit dem POD-Greedy-Algorithmus etwas schneller. Wie auch beim entsprechenden Vergleich in [BFN17, 5.1.] ist der POD-Greedy Algorithmus etwas schneller als der T-Greedy-Algorithmus. Dort wurde diese Zeitdifferenz durch die verschieden aufwändigen Offline-Schritte erklärt (vgl. Bemerkung 4.3.4(ii)). Die aufgeführten Zeiten sind nicht proportional zu den Ergebnissen aus [BFN17]. Wir erklären das durch unterschiedliche Hardware. Insbesondere wurden die Berechnungen in unserem Falle ohne GPU durchgeführt. Weiterhin fällt auf, dass recht große Unterschiede zwischen den Zeiten für stetige und unstetige Anfangsbedingungen bestehen. Die Systeme zu u_{cont}^0 werden schneller berechnet. Die Zeiten für die POD-Greedy-Basiskonstruktion mit Offline-/Online-Zerlegung (siehe Tabelle 6) zeigen ein genau gegenteiliges Bild. Es ist nicht auszuschließen, dass die Computer während der langwierigen Tests zusätzlich beansprucht wurden. Es ist unwahrscheinlich, dass verschiedene Anfangsbedingungen zu solch unterschiedlichen Zeiten führen.

Als nächstes wollen wir die Dauer für die Lösung der reduzierten Systeme vergleichen (siehe die Spalten *Online* in Tabelle 5). Es fällt auf, dass diese ungefähr im selben Bereich liegen. Jedoch scheinen die T-Greedy-reduzierten Systeme etwas mehr Zeit zum Lösen der Anfangswertprobleme zu benötigen. Wir wollen aus diesen unterschiedlichen Zeiten jedoch keine Rückschlüsse auf die allgemeine Performance ziehen. Die Experimente, aus denen die Zeiten stammen, wurden parallel auf dem selben Rechner durchgeführt. Es ist gut möglich, dass eine ungleichmäßige Verteilung der Systemressourcen die hier aufgelisteten

²⁹Zur besseren Einordnung befinden sich im Anhang unter `system/lshw.X.txt` Auflistungen zur Hardware der jeweiligen Geräte.

Zeiten beeinflusst hat.

Zuletzt wollen wir noch Daten zur Dauer der Greedy-Reduktion mit Fehlerschätzer (*vol*) nach Bemerkung 4.3.5 betrachten (siehe Tabelle 6). Für den stetigen Fall ist wieder ein leicht höhere Berechnungszeit für die T-Greedy-Reduktion zu erkennen. Auffällig ist, dass das Greedy-Verfahren für beide Reduktionsmethoden mehr Zeit beansprucht als die theoretische langsamere Variante in Tabelle 5 ohne Offline-/Online-Zerlegung. Unsere Erklärung dafür ist, dass die Online-Berechnungen auf einer großen Anzahl an Matrixmultiplikationen beruhen. Da wir die Berechnungen ohne GPU durchgeführt haben, mussten diese ineffizient über die CPU durchgeführt werden.

	T-Greedy		POD-Greedy	
	Offline	Online	Offline	Online
u_{cont}^0	32711 (A)	42.36 (A)	27838 (A)	41.90 (A)
u_{disc}^0	64936 (B)	47.09 (A)	39809 (A)	39.96 (A)

Tabelle 5: Dauer der Greedy-Reduktion mit Fehlerabschätzung (*slow*) bis zu einer reduzierten Basis der Größe $r = 20$ (Offline) verglichen mit Dauer der Lösung des entsprechenden reduzierten Systems (Online). Angabe in Sekunden.

	T-Greedy	POD-Greedy
	Offline	Offline
u_{cont}^0	38108 (A)	36628 (A)
u_{disc}^0	76819 (B)	28083 (A)

Tabelle 6: Dauer der Greedy-Reduktion mit Fehlerabschätzung (*vol*) bis zu einer reduzierten Basis der Größe $r = 20$ für die Anfangsbedingung u_{cont}^0 (links) und u_{disc}^0 (rechts). Angabe in Sekunden.

5.6 Speicherbedarf

Zuletzt wollen wir noch kurz auf den benötigten Speicherplatz für die erzeugten Reduktoren eingehen. Wie an der Abbildung 13 zu erkennen ist, steigt die Größe der T-Greedy-Reduktor-Objekte mit zunehmender Dimension der reduzierten Basis linear an, während die entsprechenden Objekte aus dem POD-Greedy-Verfahren eine nahezu konstante Größe behalten.³⁰ Dies deckt sich mit unseren Vorüberlegungen in Abschnitt 3.0.13. Hierbei ist zu bedenken, dass die reduzierten Basen wie auch die Offline-Matrizen (siehe Definition 4.3.1) in diesen Objekten gespeichert sind. Erhöhen wir die Dimension von $d = 501$ auf $d = 2001$, so wächst der benötigte Speicherplatz für beide POD-Greedy und T-Greedy-Reduktoren um etwa den Faktor 15. Dies erklärt sich damit, dass zusätzlich zu einer Erhöhung der Stützstellen auf Ω auch die Anzahl Zeitschritte nach Bemerkung 5.1.3 um den Faktor 4 wächst. Zwar lässt das ein Anwachsen des benötigten Speicherplatzes um das 16-fache vermuten, jedoch speichern die Objekte auch Daten, die sich mit steigender Dimension der Diskretisierung nicht oder nur geringfügig ändern.

³⁰Den Rohdaten ist zu entnehmen, dass auch die Größe der POD-Reduktoren linear ansteigt.

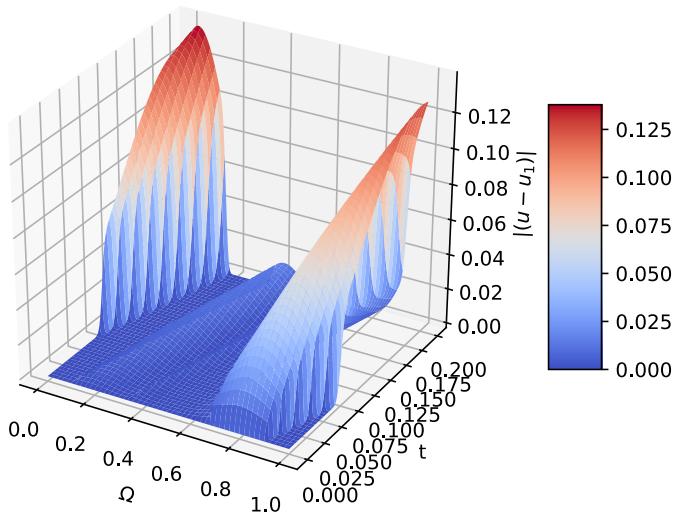
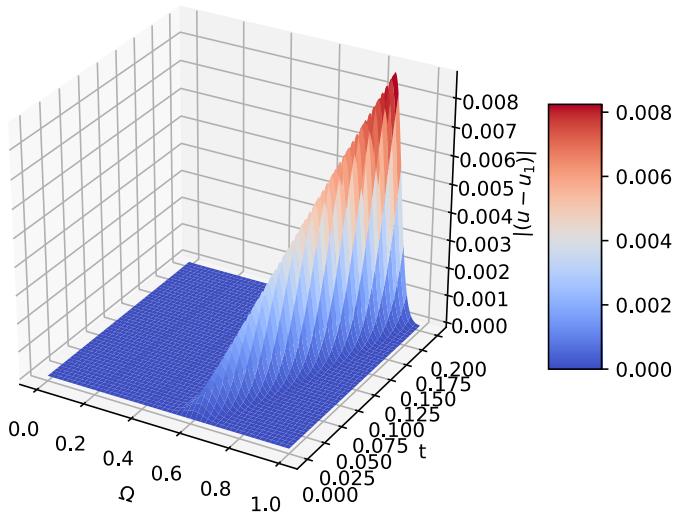


Abbildung 1: Betragsmäßiger exakter Fehler $|e_1| = |u - u_1|$ zwischen hochdimensionaler Lösung u aus explizitem Eulerverfahren und reduzierter Lösung mit T-Greedy für Parameter $\xi = 0.65$ für Anfangsbedingungen u_{cont}^0 (oben) und u_{disc}^0 (unten).

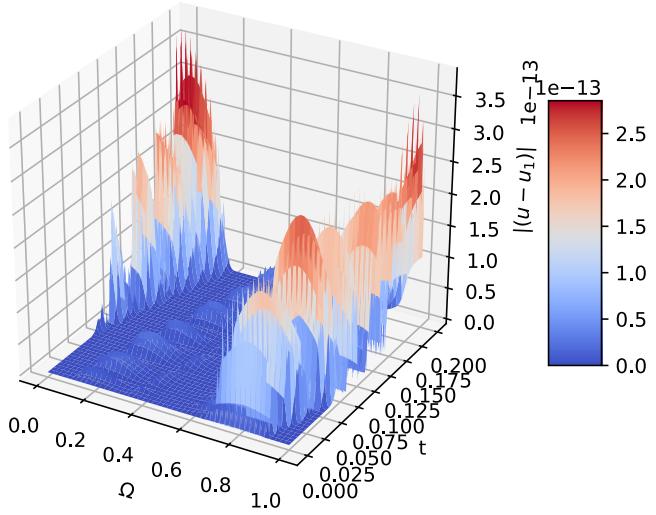
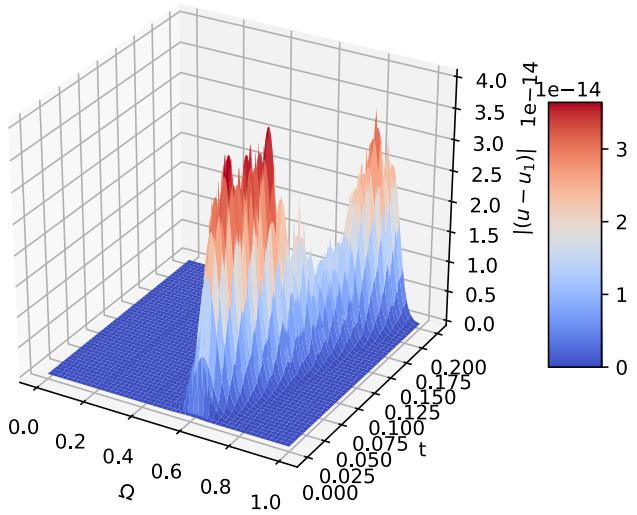


Abbildung 2: Betragsmäßiger exakter Fehler $|e_1| = |u - u_1|$ zwischen hochdimensionaler Lösung u aus implizitem Eulerverfahren und reduzierter Lösung mit T-Greedy für Parameter $\xi = 0.65$ für Anfangsbedingungen u_{cont}^0 (oben) und u_{disc}^0 (unten).

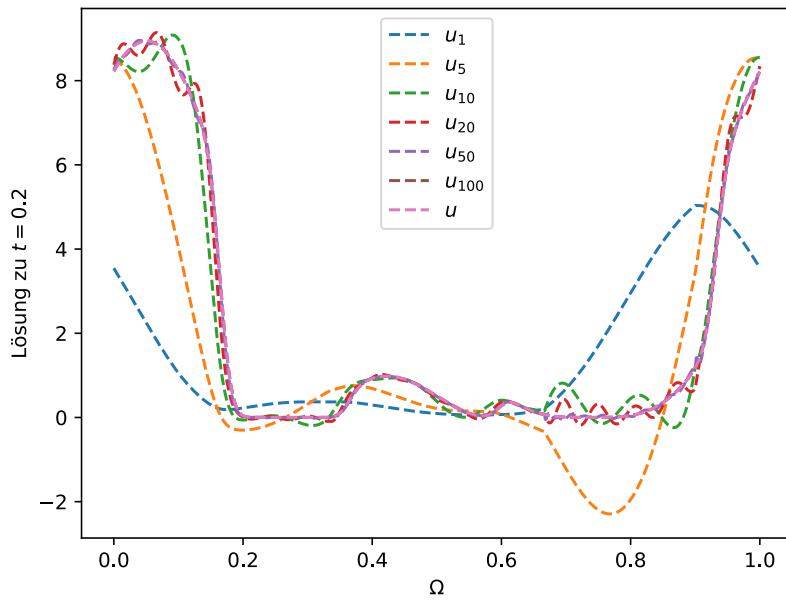
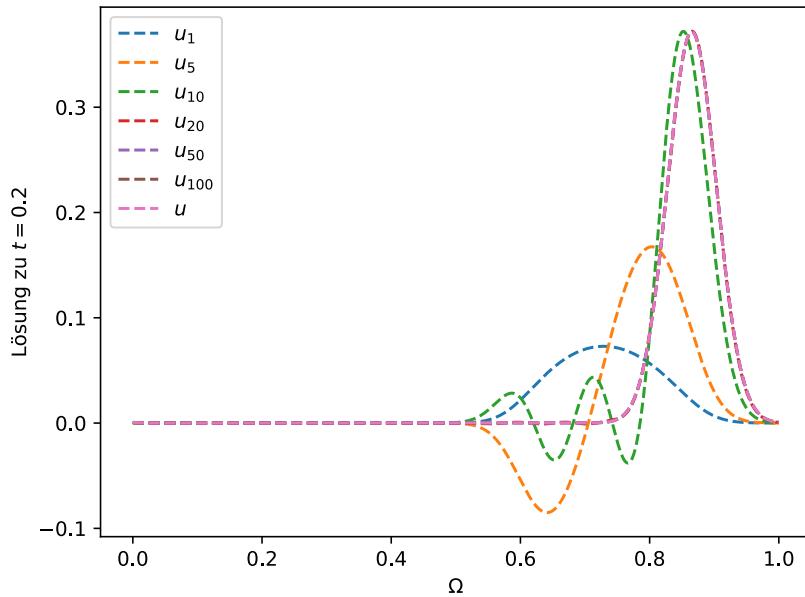


Abbildung 3: Reduzierte Lösung u_r für $r \in \{1, 2, 5, 10, 20, 50, 100\}$ und volle Lösung u im Vergleich. Die Basis wurde anhand entsprechend vieler POD-Modes der vollen Lösung zum Parameter $\xi = 0.65$ konstruiert. Anfangsbedingungen u_{cont}^0 oben, u_{disc}^0 unten. (Die Grafiken zum expliziten Eulerverfahren sind vergleichbar und wurden daher an dieser Stelle weggelassen, sind jedoch im Anhang enthalten.)

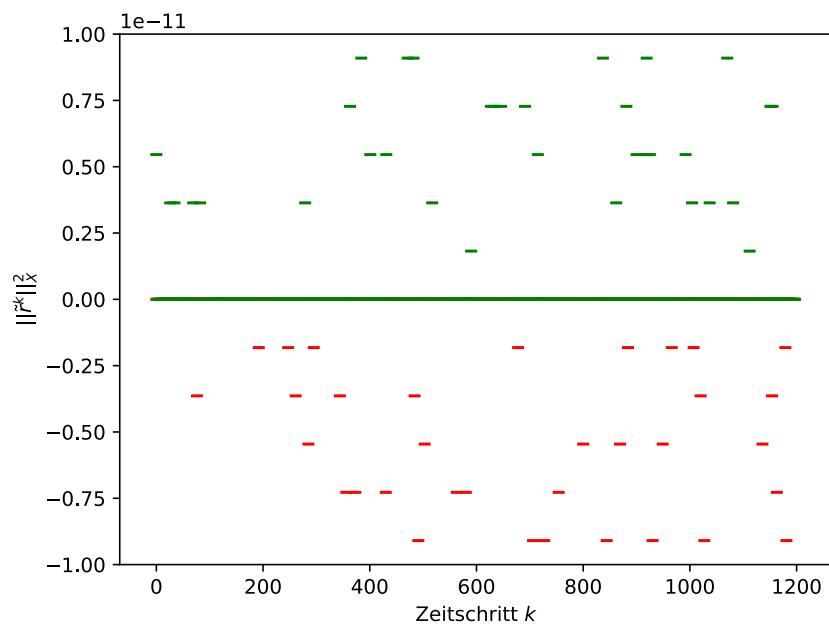


Abbildung 4: Verteilung der durch die Offline/Online-Zerlegung berechneten Werte zu $\|\tilde{r}^k(\xi)\|_X^2$ für die Approximation der implizit berechneten hochdimensionalen Lösung zu je $\xi = 0.65$ und Anfangsbedingung u_{cont}^0 . Negative Werte sind rot eingezeichnet, nichtnegative Werte in grün. Die Anhäufung der Werte um 0 entlang der x-Achse ist der logarithmischen Skala geschuldet. Für betragsmäßig kleinere Exponenten sind die berechneten Normen ähnlich verteilt (siehe auch die zu jedem deterministischen Test ausgegebenen Dateien `data.json`).

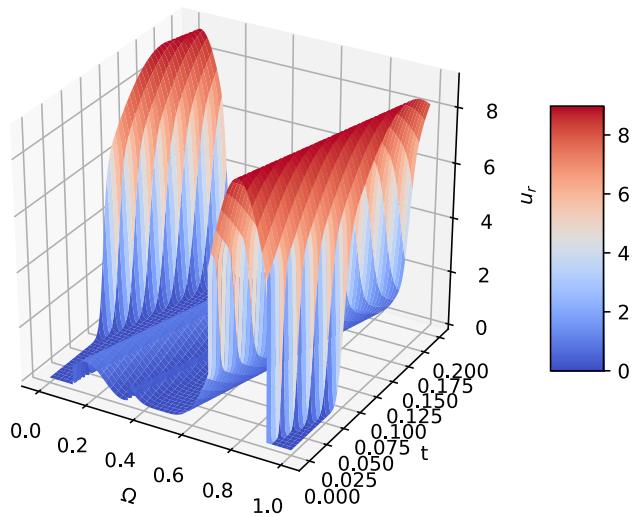
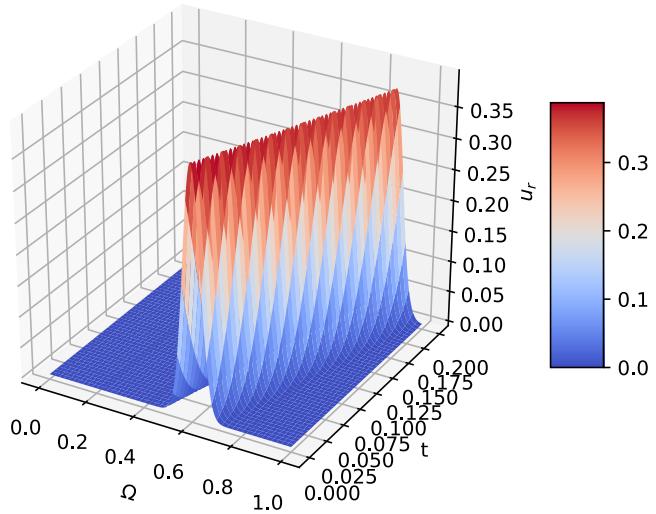


Abbildung 5: T-Greedy-reduzierte Lösung $u_r(t, \xi)$ für $r = 20$, $t = 0.2$ und $\xi = 0.65$ für Anfangsbedingungen u_{cont}^0 (oben) und u_{disc}^0 (unten). Vgl. [BFN17, Fig. 2].

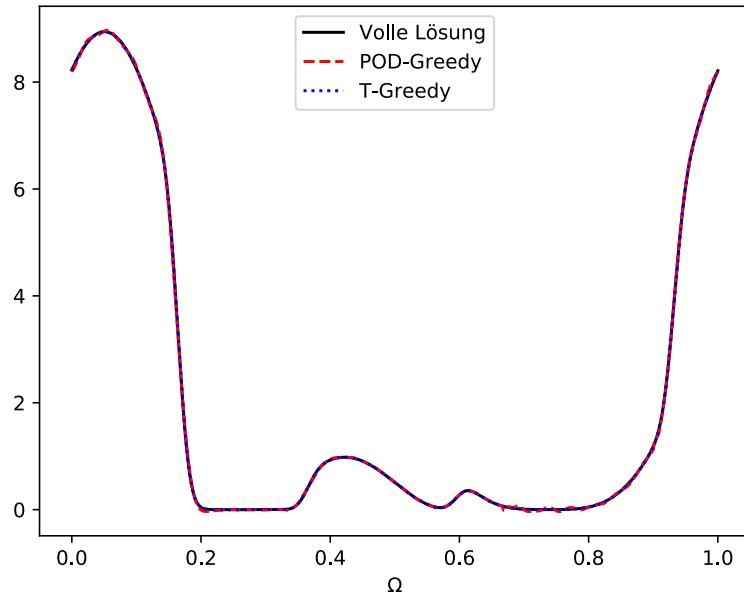
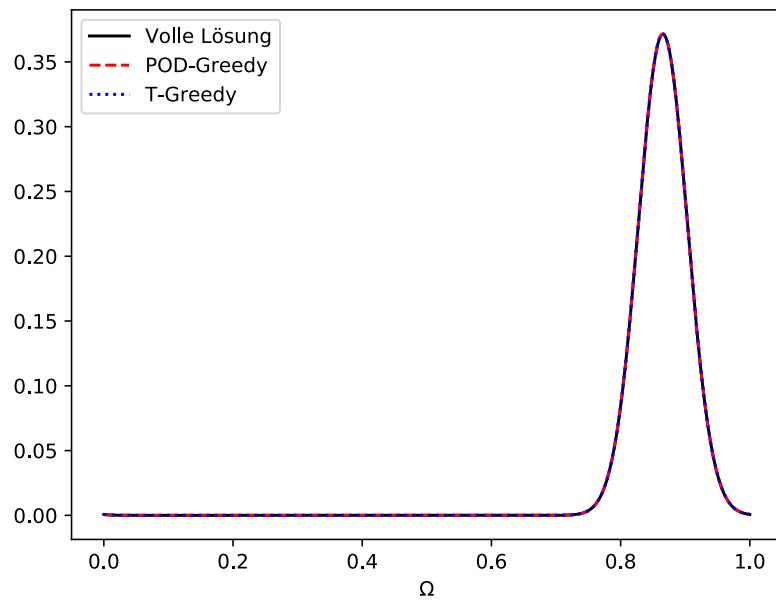


Abbildung 6: Vergleich der T-Greedy- und POD-Greedy-Approximationen mit hoch-dimensionalen Lösung jeweils zum Parameter $\xi = 0.65$ für $t = 0.2$, $r = 20$ und Anfangsbedingungen u_{cont}^0 (oben), u_{disc}^0 (unten). Vgl. [BFN17, Fig. 3].

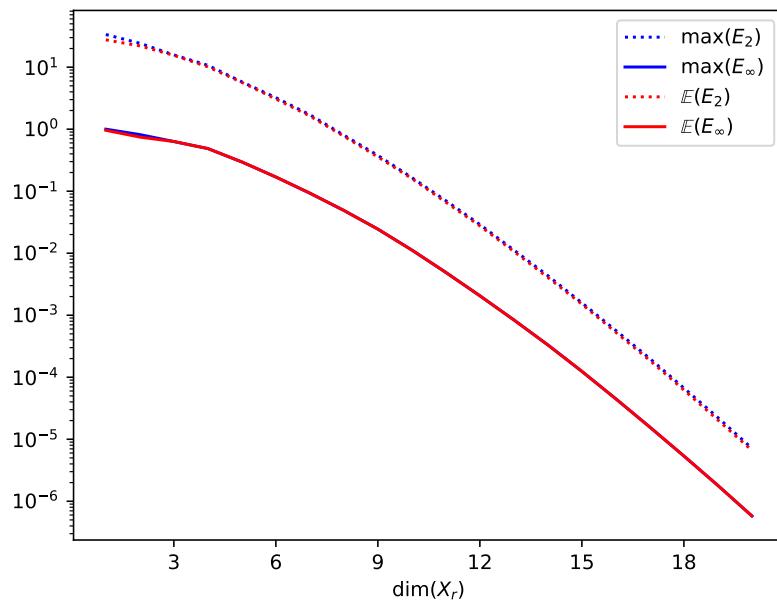
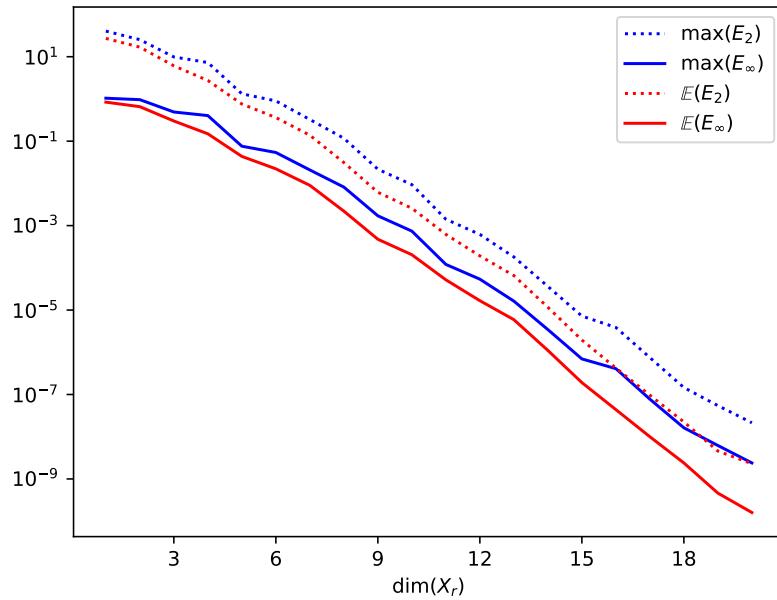


Abbildung 7: Vergleich der Entwicklung der maximalen und der nach (72) normierten relativen Fehler der Lösungen zu u_{cont}^0 und den Parametern in Ξ_{50} . T-Greedy: oben, POD-Greedy: unten.

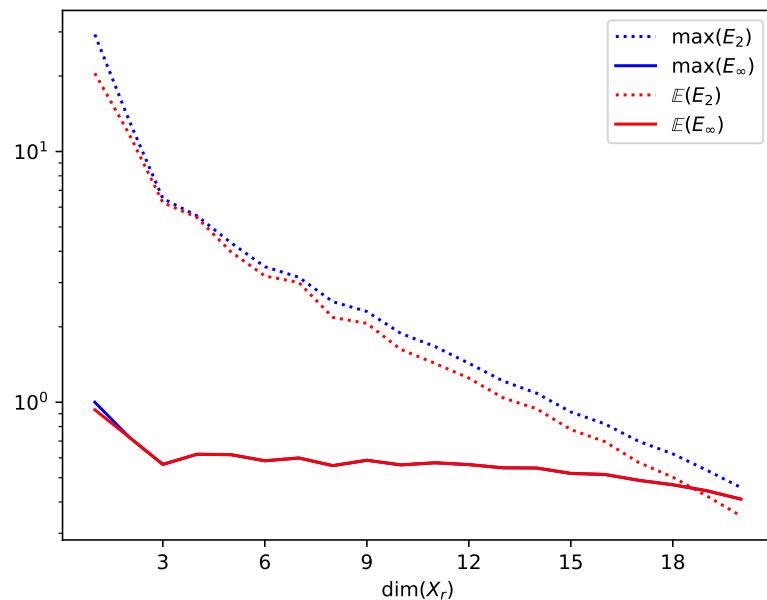
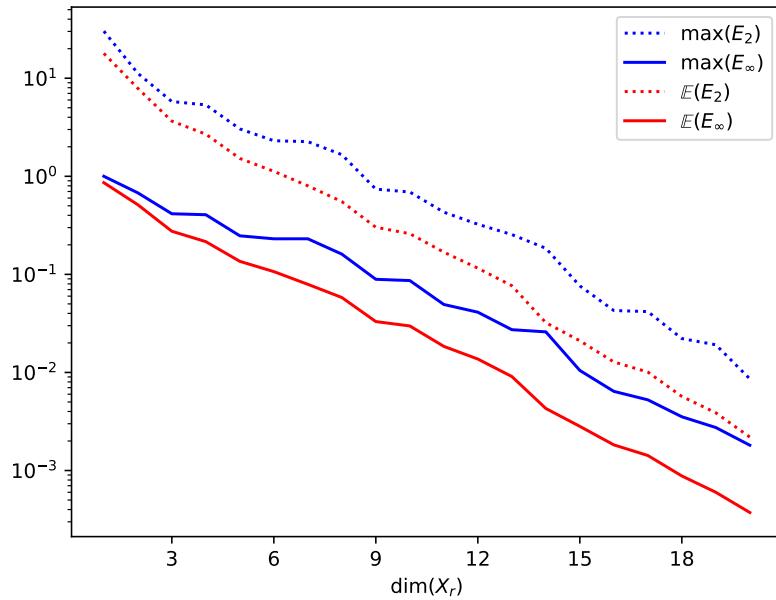


Abbildung 8: Vergleich der Entwicklung der maximalen und der nach (72) normierten relativen Fehler der Lösungen zu u_{disc}^0 und den Parametern in Ξ_{50} . T-Greedy: oben, POD-Greedy: unten.

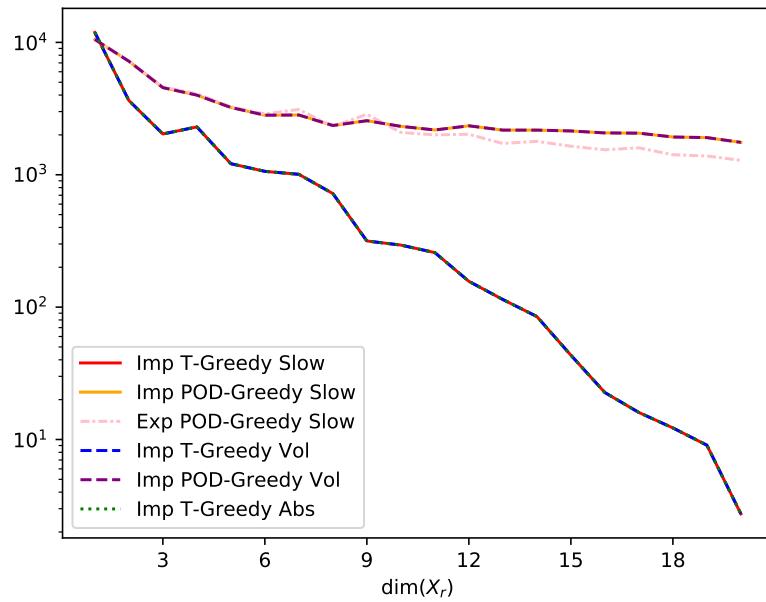
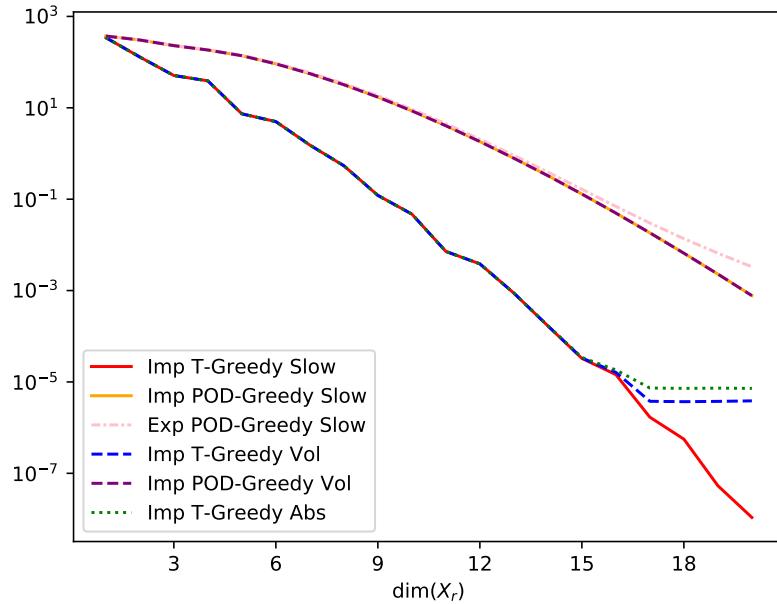


Abbildung 9: Vergleich der Entwicklung der maximalen Fehler während des Greedy-Verfahrens für die verschiedenen Fehlerabschätzungen aus Bemerkung 4.3.5. *Imp* steht für mit implizitem Eulerverfahren berechnete hochdimensionale Lösungen, *Exp* für das explizite Eulerverfahren. Anfangsbedingung u_{cont}^0 oben, u_{disc}^0 unten.

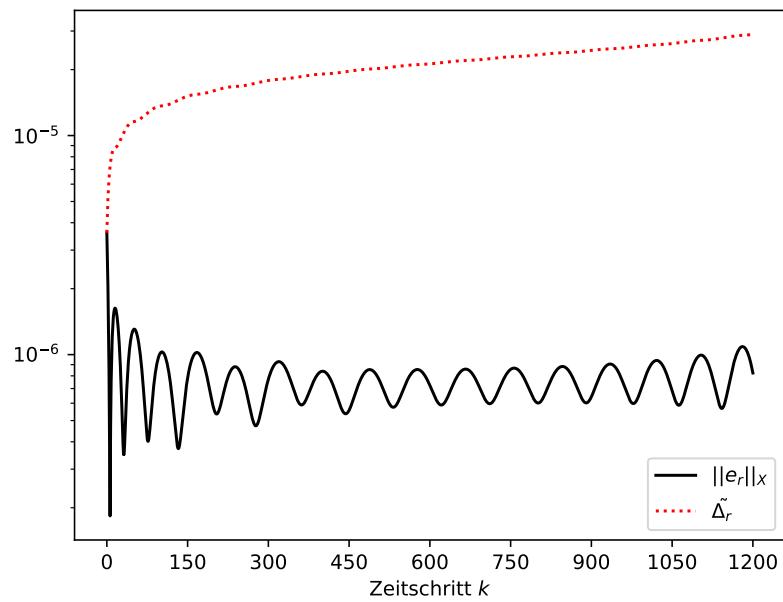
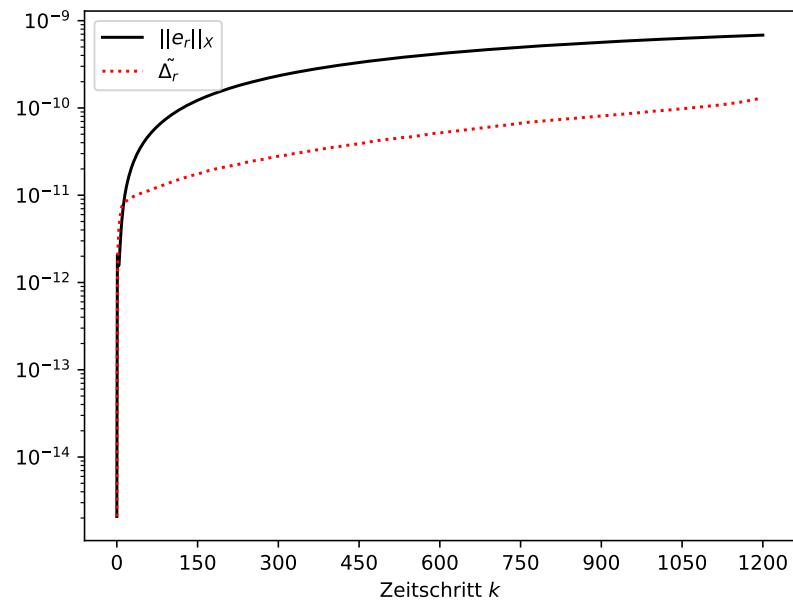


Abbildung 10: Vergleich des exakten Fehlers $\|e_r\|_X$ mit der Fehlerabschätzung $\tilde{\Delta}_r$ (slow) für $r = 20$ und Anfangsbedingung u_{cont}^0 zum Parameter $\xi = 0.65$. T-Greedy: oben, POD-Greedy: unten.

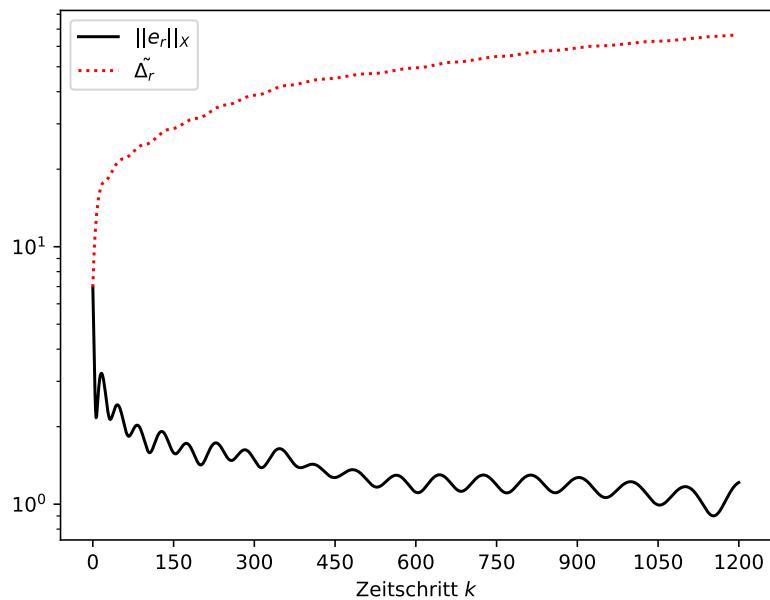
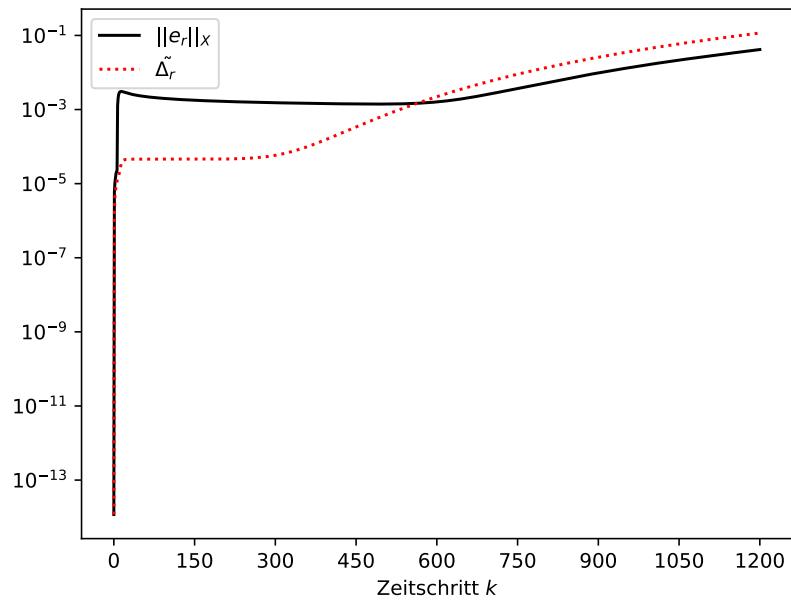


Abbildung 11: Vergleich des exakten Fehlers $\|e_r\|_X$ mit der Fehlerabschätzung $\tilde{\Delta}_r$ (slow) für $r = 20$ und Anfangsbedingung u_{disc}^0 zum Parameter $\xi = 0.65$. T-Greedy: oben, POD-Greedy: unten.

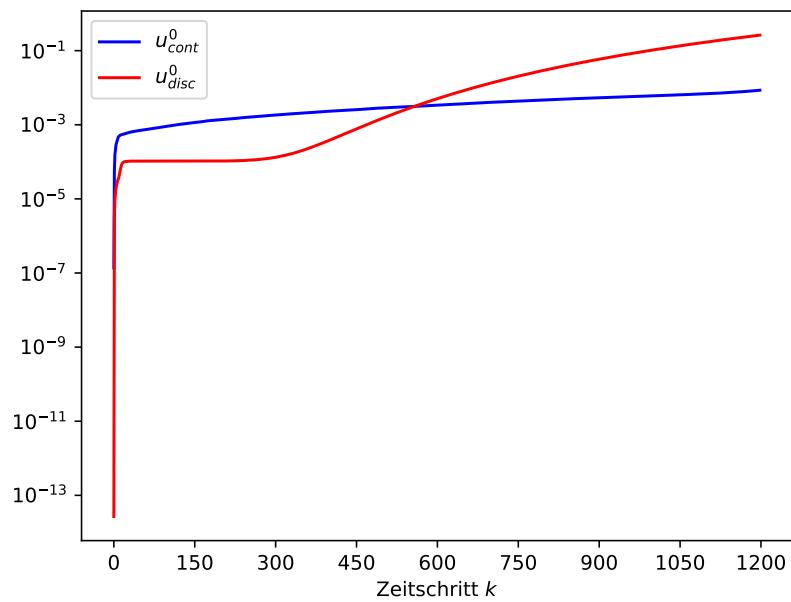


Abbildung 12: Vergleich des *effectivity index* $\kappa(t, \xi)$ für stetige und unstetige Anfangsbedingung und $r = 20$ und den Parameter $\xi = 0.65$. T-Greedy oben, POD-Greedy unten.

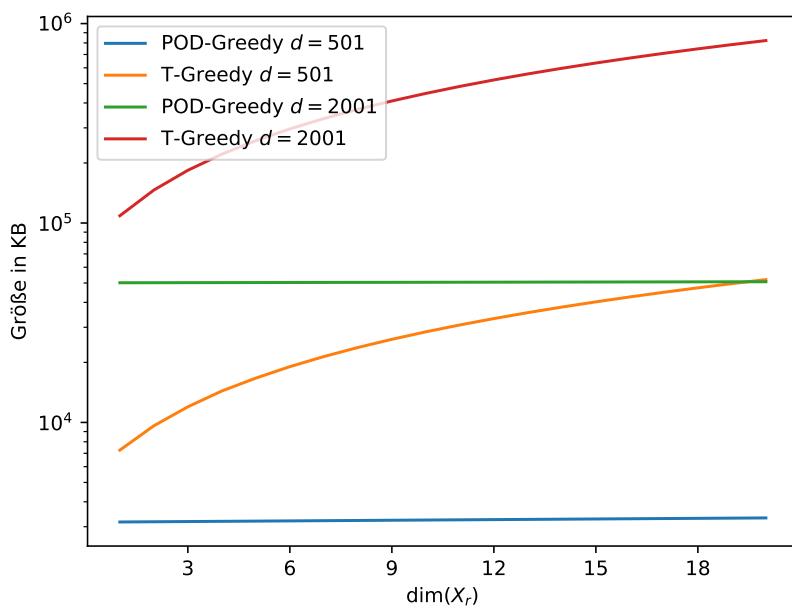


Abbildung 13: Vergleich des Speicheraufwands für die serialisierten Reduktoren zu u_{cont}^0 , Fehlerabschätzung *slow* und angegebener Dimension d .

6 Fazit

Nun möchten wir noch die wichtigsten Ergebnisse dieser Arbeit zusammengefasst darstellen und den Schlussfolgerungen aus [BFN17, 6.] gegenüberstellen. Wir haben in Abschnitt 2 formale Definitionen für das in [BFN17, 1.] eingeführte parametrisierte dynamische System sowie dessen Lösungen gegeben und Bedingungen für die Wohldefiniertheit dieser Lösungen angegeben. Die dem originären Basisreduktionsverfahren aus [BFN17] zugrunde liegende zeitabhängige Projektion des dynamischen Systems konnten wir mittels einfacher linearer Algebra in Abschnitt 2.2 erklären und Existenz- und Eindeutigkeitsaussagen der Lösungen auch für diese Systeme angeben. Ebenso wurden solche Aussagen in Abschnitt 2.2 auch für das reduzierte System niedriger Dimension getroffen.

Die theoretischen Grundlagen für die a-posteriori-Fehlerschätzung aus [BFN17, 2.2.] wurden in Abschnitt 2.3 abgebildet und erklärt. Anschließend wurde diese Fehlerschätzung in Proposition 2.3.5 vorgestellt.

Die Algorithmen zur Konstruktion reduzierter Basen, POD-Greedy und T-Greedy, wurden im Abschnitt 3 genau beschrieben und zusätzlich Erwägungen zu Speicher- und Rechenaufwand gezogen.

Die in [BFN17, 4.1.] angegebenen Zeitschrittverfahren zur Lösung des projizierten Systems und der Fehlerabschätzungen wurden in Abschnitt 4 anhand der abstrakten Konstruktionen aus Abschnitt 2 erklärt und daraus das niedrig-dimensionale Zeitschrittverfahren zur Lösung des reduzierten Systems (32) aus der Projektion des hoch-dimensionalen Verfahrens erklärt.

In Abschnitt 4 haben wir die Offline-/Online-Zerlegung aus [BFN17, 4.2.2.] zur effizienten Berechnung des Residuums detailliert nachvollzogen. Bis auf eine kleine Einschränkung für (53) haben wir die in [BFN17] vorgeschlagene Zerlegung formal beweisen können. Wir können keine Aussage über die Richtigkeit der entsprechenden Formel in [BFN17] treffen, jedoch garantiert unsere Herleitung der Online-Matrizen in Satz 4.3.3 eine korrekte Zerlegung des Residuums.

Die Reproduktion der Testergebnisse aus [BFN17, 5.1.] durch das eigens geschriebene Programm ist teilweise gelungen. Die für die stetige Anfangsbedingung erzielten relativen Fehler der reduzierten Lösung konnten in unseren Versuchen für eine mit explizitem Eulerverfahren errechnete hoch-dimensionale Lösung³¹ nicht bestätigt werden (siehe Tabelle 1). Die in unseren Tests errechneten Werte liegen im Gegensatz zu denen aus [BFN17] weit von der verwendeten Maschinenpräzision entfernt. Wie im Vergleich der Abbildungen 1 und 2 zu erkennen ist, diffundiert die mit expliziten Verfahren berechnete hoch-dimensionale Lösung stärker als die implizit berechnete, was wir als Grund für den abweichenden relativen Fehler ansehen. Wir haben mit Korollar 5.1.6 zwar die Erfüllung einer notwendigen Bedingung für die Stabilität des expliziten Verfahrens gezeigt³², konnten die entsprechenden Resultate der Experimente in Abschnitt 5.2 jedoch nicht zufriedenstellend erklären.

Hingegen liefert das T-Greedy-Verfahren für implizit berechnete hoch-dimensionale Lösungen zufriedenstellendere Ergebnisse. Wie Tabelle 2 zu entnehmen ist, sind die relativen Fehler dieser Basisreduktion deutlicher näher an der Maschinengenauigkeit. Inwiefern diese Resultate sich mit denen aus [BFN17, 5.1.] vergleichen lassen, hängt von der dort genutzten Rechengenauigkeit ab.

³¹Vgl. Schema [BFN17, (28)]

³²Zur CFL-Condition als notwendige Bedingung für Stabilität siehe [LeV02, 4.4, S.69].

Sollte in [BFN17] eine ähnliche Präzision wie für unsere Tests, verwendet worden sein, so überschreiten die durch unsere Implementation errechneten Approximationsfehler die Resultate aus [BFN17] um ungefähr den Faktor 1000. Im Falle einer in [BFN17] verwendeten Maschinengenauigkeit von etwa 10^{-19} sind unsere Ergebnisse - vom verwendeten hoch-dimensionalen Lösungsverfahren abgesehen - mit denen aus [BFN17, 5.1.] vergleichbar. Eine weiterer Grund für die abweichenden Ergebnisse zum relativen Fehler könnte die Toleranz verwendet numerischer Verfahren, etwa zur Orthonormalisierung, sein.

Nicht auszuschließen ist darüber hinaus ein Fehler in der eigenen Implementation des Verfahrens. Trotz umfassender Unterstützung für RB-Verfahren des Frameworks pyMOR, nutzt das für die Experimente erstellte Programm dessen Funktionen nicht optimal. Anstatt eigene Reduktor-Klassen von Grund auf zu implementieren, hätten wir die bereits existierenden Klassen aus dem Package **reductors** verwenden können. Die iterativen Lösungsverfahren der hoch-dimensionalen dynamischen Systeme wurden selbst geschrieben, um möglichst nah an der Vorgabe in [BFN17, (28)] arbeiten zu können. Im Nachhinein wäre eine Nutzung der entsprechenden pyMOR Implementationen von Vorteil gewesen, um die Komplexität des eigenen Codes überschaubar zu halten. Der eigene Quellcode befindet sich im Anhang, sodass ihn jeder auf etwaige Fehler untersuchen kann.

Unwahrscheinlich - jedoch nicht ganz auszuschließen - ist, dass die ursprüngliche Versuchsbeschreibung in [BFN17, 5.1.] nicht exakt den präsentierten Ergebnissen entspricht. Dies steht natürlich nur dann im Raum, wenn kein Fehler in unserer Implementation der Greedy-Verfahren mehr in Frage kommt. Die Benutzung eines impliziten Eulerverfahrens in [BFN17, 5.1.] würde unsere Ergebnisse des deterministischen Versuchs aus Abschnitt 5.2 gut erklären.

Der wohl einfachste Weg, die Unterschiede zu den Resultaten aus [BFN17, 5.1.] zu analysieren und zu erklären, wäre eine Einsicht in den Matlab-Code, mit dem die dort gezeigten Ergebnisse erzeugt wurden.

Die Durchführung der Greedy-Verfahren, basierend auf den implizit berechneten hoch-dimensionalen Lösungen, ist gelungen. Wir haben gute Approximationen mit einer durch das T-Greedy-Verfahren reduzierten Basis der Dimension 20 erhalten (siehe Abbildung 5). Die in [BFN17, 5.1.] gezeigten Ergebnisse zum Vergleich von T-Greedy und POD-Greedy konnten in diesem Rahmen nachgebildet werden. Trotz bereits angesprochener Unterschiede zu [BFN17] in der konkreten Präzision der durch das reduzierte System approximierten Lösungen, sind die durch uns berechneten Fehler der T-Greedy-Reduktion ebenso um einige Größenordnungen geringer als die der POD-Greedy-Reduktion. Diese Beobachtung wurde für eine stetige und eine unstetige Anfangsbedingung gemacht, wobei das T-Greedy-Verfahren deutlich bessere Approximationen für den unstetigen Fall liefert.

Wie aufgrund der Ergebnisse des deterministischen Tests zu erwarten war, konnte das T-Greedy-Verfahren nicht für eine explizit berechnete volle Lösung durchgeführt werden. Dies ist bedauerlich, da das explizite Zeitschrittverfahren generell weniger Rechenarbeit erfordert.

Die effiziente Offline-/Online-Zerlegung für die a-posteriori Fehlerabschätzung konnte experimentell nicht bestätigt werden. Aufgrund unseres Beweises zu Satz 4.3.3 ist zwar die formale Korrektheit dieser Zerlegung für den in Abschnitt 5 betrachteten Fall belegt, jedoch scheitert dieses Verfah-

ren in der Praxis. Wir vermuten, dass Abschneidefehler bei der Berechnung der Offline-Matrizen dafür verantwortlich sind. Aufgrund dessen haben wir mit Bemerkung 4.3.5 einige von der Offline-/Online-Zerlegung abgeleitete Verfahren zur Fehlerschätzung präsentiert, welche die beobachteten Fehlerquellen des ursprünglichen Verfahrens vermeiden und trotzdem die vorab berechneten Matrizen nutzen. Diese alternativen Methoden wurden experimentell untersucht. Zwar konnten wir die abgeänderten Fehlerschätzungen für die Greedy-Auswahl für einige Basiserweiterungsschritte gut nutzen, es wurde aber nicht dieselbe Präzision wie mit dem langsamen, nicht Offline-/Online-zerlegten Verfahren, erreicht.

Für die ansonsten sehr zuverlässige Fehlerabschätzung *slow* (siehe Bemerkung 4.3.5) haben wir in den Abbildungen 10 und 11 eine Verletzung der eigentlich durch Proposition 2.3.5 garantierten Beschränkung der a-posteriori Abschätzung durch die Norm des exakten Fehlers beobachtet. Dies tritt nur für T-Greedy-reduzierte Lösungen auf und liegt nicht in den vergleichbaren Grafiken in [BFN17] vor. Der Grund hierfür konnte nicht geklärt werden, die Vermutung liegt wieder bei Problemen mit der Stabilität.

Der Vergleich der benötigten Rechenzeit mit den Daten aus [BFN17, Table 2] erweist sich als schwierig. Ohne Spezifikation der dort verwendeten Hardware ist eine genaue Beurteilung der vorgestellten Ergebnisse kaum möglich. Trotzdem können wir wie in [BFN17] eine vergleichbare Rechendauer für beide Greedy-Verfahren beobachten.

Zusätzlich haben wir durch die Reduktionsverfahren angefallene Datenmengen untersucht. Im Gegensatz zum POD-Greedy-Verfahren wächst der für die zeitabhängigen reduzierten Basen benötigte Speicherplatz sehr schnell an. Je nach konkreter Anwendung mag der T-Greedy-Algorithmus deshalb trotz besserer Approximation keine gute Alternative zu anderen RB-Verfahren wie dem POD-Greedy darstellen.

Insgesamt ergibt über die durchgeföhrten Experimente ein eher gemischtes Bild. In der Praxis scheint der T-Greedy-Algorithmus zulasten des Speicherplatzes nach gleich vielen Basiserweiterungen deutlich präzisere Approximationen als das POD-Greedy-Verfahren zu produzieren. Dies gilt vor allem für unstetige Anfangswerte. Jedoch scheint der Algorithmus auch anfälliger für leicht gestörte Zwischenergebnisse zu sein. Dies äußert sich vor allem in der Berechnung der Fehlerabschätzung. Die hier getroffenen Aussagen zum POD-Greedy-Algorithmus gelten zudem nur für den Spezialfall der Basiserweiterung um je einen POD-Mode zu jeder Greedy-Auswahl. Es bleibt zu überprüfen, ob andere Varianten des POD-Greedy im Vergleich mit dem T-Greedy-Schema vergleichbare Resultate zeigen. In dieser Arbeit wurde nur einer von drei in [BFN17, 5.] durchgeföhrten Tests realisiert. Reproduktionen der anderen Experimente, vor allem des inhomogenen Problems mit nichtlinearem Fluss in [BFN17, 5.3.], wären zusätzlich von Interesse. Die Herleitung und Implementation einer zeitabhängigen Variante der in Bemerkung 4.3.6 angesprochenen stabilen Zerlegung des Residuums wäre aufgrund der hier vielfach vermuteten Stabilitätsprobleme auch von großem Interesse bei einer erneuten Untersuchung der Fehlerabschätzung mit Offline-/Online-Zerlegung.

Literatur

- [ABCM02] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.
- [BCD⁺11] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM Journal on Mathematical Analysis*, 43(3):1457–1472, 2011.
- [BEOR14] A. Buhr, C. Engwer, M. Ohlberger, and S. Rave. A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations. *arXiv e-prints*, July 2014.
- [BFN17] M. Billaud-Friess and A. Nouy. Dynamical model reduction method for solving parameter-dependent dynamical systems. *SIAM Journal on Scientific Computing*, 39(4):A1766–A1792, 2017.
- [BMS13] Mario Bebendorf, Yvon Maday, and Benjamin Stamm. Comparison of some Reduced Representation Approximations. *arXiv e-prints*, page arXiv:1305.5066, May 2013.
- [Bos10] Siegfried Bosch. *Lineare Algebra*. Springer-Verlag Berlin Heidelberg, 2010.
- [Chi99] C. Chicone. *Ordinary Differential Equations with Applications*. Texts in applied mathematics. Springer, 1999.
- [Haa13] Haasdonk, Bernard. Convergence rates of the pod-greedy method. *ESAIM: M2AN*, 47(3):859–873, 2013.
- [HO08] Bernard Haasdonk and Mario Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 42(2):277–302, 2008.
- [HP57] E. Hille and R.S. Phillips. *Functional Analysis and Semi Groups*. Colloquium Publications - American Mathematical Society. American Mathematical Soc., 1957.
- [Krö97] Dietmar Kröner. *Numerical schemes for conservation laws*. Chichester ; New York : Wiley ; Stuttgart : Teubner, 1997.
- [LeV02] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.
- [MRF16] R. Milk, S. Rave, and F. Schindler. pymor - generic algorithms and interfaces for model order reduction. *SIAM J. Sci. Comput.*, 38(5), pp S194–S216, 2016.
- [Qua15] Alfio Quarteroni. *Reduced Basis Methods for Partial Differential Equations: An Introduction (UNITEXT)*. Springer, 2015.

- [Tem11] V. Temlyakov. *Greedy Approximation*. Cambridge University Press, 2011.
- [Tho06] Vidar Thomée. *Galerkin Finite Element Methods for Parabolic Problems (Springer Series in Computational Mathematics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [Wal00] W. Walter. *Gewöhnliche Differentialgleichungen*. Springer, 2000.
- [WSH14] D. Wirtz, D. Sorensen, and B. Haasdonk. A posteriori error estimation for deim reduced nonlinear dynamical systems. *SIAM Journal on Scientific Computing*, 36(2):A311–A338, 2014.