

BACHELORARBEIT

VERFOLGUNG VON EINZELZELLEN MIT PET  
UND KUBISCHER SPLINE-INTERPOLATION

Björn Luig  
Matrikelnummer: 448689

Betreuer: Prof. Dr. Benedikt Wirth  
Zweitgutachter: Dr. Frank Wübbeling

30. September 2020

# Inhaltsverzeichnis

<b>1</b>	<b>Kurzfassung</b>	<b>2</b>
<b>2</b>	<b>Einführung</b>	<b>2</b>
2.1	Zielsetzung . . . . .	2
2.2	Positronen-Emissions-Tomografie . . . . .	2
2.3	Bekannte Verfahren . . . . .	3
<b>3</b>	<b>Theorie</b>	<b>4</b>
3.1	Konvexität . . . . .	4
3.2	Splines . . . . .	5
3.2.1	Lineare Splines . . . . .	6
3.2.2	Kubische Splines . . . . .	7
3.2.3	B-Splines . . . . .	8
3.2.4	Dreidimensionale B-Splines . . . . .	12
3.2.5	Approximation mit kubischen B-Splines in einer Dimension . . . . .	12
3.3	PET-Daten und Trajektorie der Zelle . . . . .	14
3.4	Zielfunktionen . . . . .	15
3.4.1	Datenterm . . . . .	15
3.4.2	Regularisierungsterm zur Minimierung der Geschwindigkeit . . . . .	20
3.4.3	Regularisierungsterm zur Minimierung der Beschleunigung . . . . .	22
<b>4</b>	<b>Methoden</b>	<b>22</b>
4.1	Simulieren von PET-Daten . . . . .	23
4.2	Fehlerabschätzung . . . . .	24
4.3	Rekonstruktion der Trajektorie der Zelle aus PET-Daten . . . . .	25
<b>5</b>	<b>Ergebnisse und Diskussion</b>	<b>25</b>
5.1	Datenterm . . . . .	25
5.2	Regularisierung durch Minimierung der Geschwindigkeit . . . . .	26
5.3	Regularisierung durch Minimierung der Beschleunigung . . . . .	26
5.4	Regularisierung durch Minimierung der Beschleunigung bei externen PET-Daten . . . . .	27
<b>6</b>	<b>Schlussfolgerung</b>	<b>29</b>
<b>7</b>	<b>Literaturverzeichnis</b>	<b>30</b>
<b>8</b>	<b>Abbildungsverzeichnis</b>	<b>31</b>
<b>9</b>	<b>Anhang</b>	<b>32</b>
9.1	Matrizen . . . . .	32
9.2	Weitere Abbildungen zur Rekonstruktion der Zellbewegung . . . . .	34
9.3	Plagiatserklärung . . . . .	41

# 1 Kurzfassung

Diese Arbeit beschäftigt sich mit der Verfolgung einzelner Zellen in einem lebenden Organismus. Dabei soll aufbauend auf [5] die zeitabhängige Position  $\vec{r}(t)$  einer solchen Zelle mithilfe von kubischen Splines modelliert und aus Daten der Positronen-Emissions-Tomografie rekonstruiert werden. Es werden Daten- und Regularisierungsterme hergeleitet und auf Existenz und Eindeutigkeit ihrer Minimierer untersucht. Bei den Regularisierungen handelt es sich um eine Minimierung der Geschwindigkeit wie in [5] und eine Minimierung der Beschleunigung. Beides bevorzugt physikalisch wahrscheinlichere Zellbewegungen. Anschließend werden PET-Daten selbst simuliert und extern simulierte PET-Daten aus [8] verwendet, um Datenterm und Regularisierungen zu testen. Wir zeigen, dass sich sowohl eine Minimierung der Geschwindigkeit, als auch eine Minimierung der Beschleunigung eignen, um den Rekonstruktionsfehler bei einer einfachen Trajektorie deutlich zu verkleinern.

## 2 Einführung

### 2.1 Zielsetzung

Es gibt in der Medizin und Forschung viele Szenarien, in denen man daran interessiert ist, die Bewegung einzelner Zellen nachzuverfolgen. Mögliche Anwendungen dafür sind das Aufspüren von Entzündungen und Infektionen durch das Verfolgen von weißen Blutkörperchen, die Untersuchung von biologischen Prozesse wie beispielsweise der Krebsmetastase, das Markieren von transplantierten Zellen und der Untersuchung der Prozesse in der Immuntherapie [5].

### 2.2 Positronen-Emissions-Tomografie

Es gibt mehrere bildgebende Verfahren, mit denen man die Verfolgung einzelner Zellen realisieren kann. Im Folgenden wird aufbauend auf [5] die Positronen-Emissions-Tomografie (PET) behandelt.

Die PET verwendet Radionukleide, welche beim Zerfall Positronen emittieren ( $\beta$ -Zerfall), als Marker im lebenden Körper. Meistens wird hierzu das Isotop  $^{18}\text{F}$  in Form von 2-Fluor-2-desoxy-D-glucose (FDG) verwendet [3]. Zerfällt ein  $^{18}\text{F}$  Atom, so wird ein Positron emittiert. Die Annihilation dieses Positrons mit einem umliegenden Elektron führt zur Emission von zwei Photonen mit einer jeweiligen Energie von 511 keV. Diese Photonen bewegen sich in entgegengesetzter Richtung. Werden beinahe zeitgleich zwei Photonen mit entsprechenden Energien an zwei umliegenden Detektoren gemessen, so fand die Annihilation des Positrons wahrscheinlich auf einer Linie dazwischen statt. Diese Linie wird auch als „line of response“ bezeichnet. In einer kleinen Umgebung davon hat sich dann der Marker während des Zerfalls befunden. Dieser Vorgang ist in Abbildung 1 schematisch dargestellt. Die Liste der gemessenen Ereignisse besteht aus Zeitpunkten und den Positionen der jeweiligen Detektorpaare eines PET-Scans. Diese wird auch mit „list-mode“-Daten bezeichnet.

Normalerweise werden aus diesen Daten ein oder mehrere Bilder rekonstruiert. Diese eignen sich aber schlecht, um einzelne Zellen zu verfolgen, wie in [5] dargestellt. Auch Sequenzen von Bildern sind schlecht um eine kontinuierliche Bewegung darzustellen, da die Anzahl an Messungen pro Zeitabschnitt zu klein ist. Wir sind deshalb nicht an Bildern, sondern an dem zeitabhängigen Pfad  $\vec{r}(t)$  einer einzelnen Zelle interessiert. Dieser muss

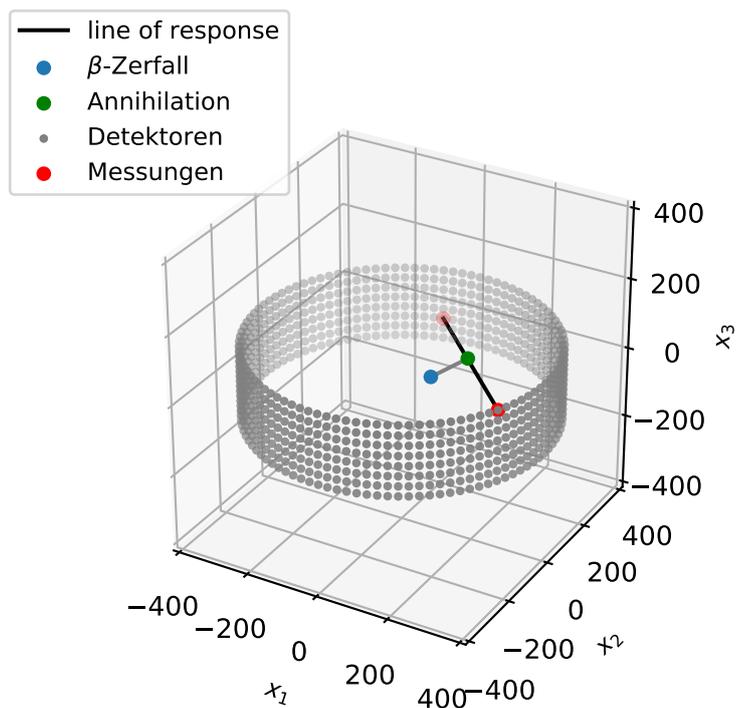


Abbildung 1: Schematische Darstellung der Messung einer Positronenemission (blau). Zwei Detektoren (rot) erfassen zwei Photonen, welche durch die Annihilation (grün) eines Positrons mit einem umliegenden Elektron entstanden sind. Die Annihilation fand auf der „line of response“ (schwarz) zwischen den entsprechenden Detektoren statt. Zur Veranschaulichung wurde eine deutlich größere Weglänge des Positrons gewählt, als später verwendet wird. (Die Standardabweichung je Raumkoordinate wurde hier zur besseren Anschauung auf  $\sigma = 100$  mm gesetzt. In den verwendeten simulierten PET-Daten beträgt sie 10 mm, in realen Szenarien ist sie deutlich kleiner.)

aus den „list-mode“ Daten rekonstruiert werden.

## 2.3 Bekannte Verfahren

Eine schon bekannte Methode, die in der Chemie Anwendung findet, ist das sogenannte „Positron Emission Particle Tracking“ (PEPT) [4]. Ereignisse in einem Zeitintervall werden zusammengefasst und jeweils die Position mit dem kleinsten Abstand zu den Messungen bestimmt (Minimum-Distance Methode). Messungen, die zu weit abweichen, werden schrittweise vernachlässigt und die Rechnung wiederholt, um zufällige Ereignisse herauszufiltern. In den meisten PEPT Studien werden 1–30 MBq  $^{18}\text{F}$  verwendet. Einzelne Zellen können aber nur mit 1–10 Bq FDG markiert werden [5].

Aus diesen Gründen wird in [5] eine neue Methode vorgeschlagen, die mit deutlich weniger Messungen auskommt. Die Trajektorie  $\vec{r}(t)$  einer Zelle wird dabei mit kubischen B-Splines modelliert, sowie eine Regularisierung in Form einer Geschwindigkeitsminimierung vorgenommen.

## 3 Theorie

### 3.1 Konvexität

Zunächst wollen wir angelehnt an [6] den Begriff der Konvexität einer Menge sowie eines Funktionals einführen. Dies hilft uns später die Eindeutigkeit von Minimierern bestimmter Zielfunktionen zu untersuchen.

**Definition 1** (konvexe Menge). Sei  $V$  ein Vektorraum. Die Teilmenge  $X \subseteq V$  ist konvex, wenn

$$\{x_1 + \lambda(x_2 - x_1) | \lambda \in [0, 1]\} \subseteq X$$

für alle  $x_1, x_2 \in X$ . Das heißt, dass die Verbindungsstrecke zweier Elemente aus  $X$  immer in  $X$  liegt.

**Definition 2** (konvexes Funktional). Sei  $F : X \rightarrow \mathbb{R}$  ein Funktional auf einer konvexen Menge  $X$ . Das Funktional  $F$  ist konvex, wenn

$$F(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda F(x_1) + (1 - \lambda)F(x_2) \quad (1)$$

für alle  $x_1, x_2 \in X$  und  $\lambda \in [0, 1]$ . Weiter heißt  $F$  strikt konvex, wenn

$$F(\lambda x_1 + (1 - \lambda)x_2) < \lambda F(x_1) + (1 - \lambda)F(x_2) \quad (2)$$

für alle  $x_1, x_2 \in X$  mit  $x_1 \neq x_2$  und  $\lambda \in (0, 1)$ .

Dabei folgt aus der obigen Definition direkt, dass endliche Summen konvexer Funktionale konvex und endliche Summen konvexer Funktionale, von denen mindestens eins streng konvex ist, streng konvex sind. Zudem ist anzumerken, dass strikte Konvexität Konvexität impliziert. Der nachfolgende Satz gilt somit auch für strikt konvexe Funktionale.

**Satz 3** (Hinreichende Bedingung für globale Minima). Sei  $F : X \rightarrow \mathbb{R}$  konvex und stetig differenzierbar. Sei  $x_0 \in X$  eine Nullstelle der Ableitung  $\partial F(x)/\partial x$ , so ist  $x_0$  ein globaler Minimierer von  $F$ .

**Beweis.** Sei  $x_0 \in X$  eine Nullstelle der Ableitung  $\partial F(x)/\partial x$ . Dann gilt

$$\lim_{\epsilon \rightarrow 0} \frac{F(x_0 + \epsilon v) - F(x_0)}{\epsilon} = 0 \quad (3)$$

für alle  $v \in X - x_0$ . Aus der Konvexität von  $F$  folgt dann für  $x_0 + v \in X$  und  $\lambda \in (0, 1]$

$$\begin{aligned} F(x_0 + \lambda v) &= F(\lambda(x_0 + v) + (1 - \lambda)x_0) \leq \lambda F(x_0 + v) + (1 - \lambda)F(x_0) \\ &\Rightarrow \frac{F(x_0 + \lambda v) - F(x_0)}{\lambda} \leq F(x_0 + v) - F(x_0). \end{aligned}$$

Mit (3) und  $\lambda \rightarrow 0$  folgt dann

$$F(x_0) \leq F(x_0 + v)$$

für alle  $v \in X$ . Dies ist die Behauptung  $\square$ .

**Satz 4** (Eindeutigkeit des Minimums). Ist ein Funktional  $F : X \rightarrow \mathbb{R}$  strikt konvex, so hat  $F$  höchstens ein globales Minimum auf  $X$ .

**Beweis.** Seien  $x_1, x_2 \in X$  mit  $x_1 \neq x_2$  beides globale Minimierer eines strikt konvexen Funktionals  $F : X \rightarrow \mathbb{R}$ . Setze  $F_{\min} = F(x_1) = F(x_2)$ . Aus der strikten Konvexität folgt dann

$$F\left(\frac{x_1 + x_2}{2}\right) < \frac{F(x_1) + F(x_2)}{2} = F_{\min}.$$

Da  $X$  konvex und somit  $\frac{x_1 + x_2}{2}$  in  $X$  liegt, ist dies ein Widerspruch zur Definition von  $F_{\min}$ . Es folgt die Behauptung  $\square$ .

Wegen Satz 3 können wir die globalen Minimierer von den später verwendeten Daten- und Regularisierungstermen durch Ableiten ermitteln, sofern wir Konvexität gezeigt haben. Ist diese strikt, so ist der Minimierer wegen Satz 4 eindeutig. Dies gilt dann auch für die Nullstelle der entsprechenden Ableitung.

## 3.2 Splines

Der folgende Abschnitt über Splines ist inhaltlich an [7] und [9] angelehnt. Jedoch wurde die Notation aufgrund der Übersichtlichkeit an den später verwendeten Python-Code und [5] angepasst.

**Definition 5** (Interpolationsfunktion). Man betrachte ein Intervall  $[t_1, t_n] \in \mathbb{R}$ , die Stützstellen  $t_1 < t_2 < \dots < t_n$  und die Stützwerte  $x_1, x_2, \dots, x_n$ .

Die Funktion  $f : [t_1, t_n] \rightarrow \mathbb{R}$  ist eine Interpolationsfunktion zu den Punkten  $(t_i, x_i)$ , wenn  $f(t_i) = x_i$  für alle  $i = 1, \dots, n$ .

Weiter nennen wir eine Interpolationsfunktion  $m$ -mal stückweise stetig differenzierbar, wenn  $f|_{(t_i, t_{i+1})}$  für alle  $i = 1, \dots, n - 1$   $m$ -mal stetig differenzierbar ist.

Eine Möglichkeit der Interpolation von Punkten  $(t_i, x_i)$  ist die Polynominterpolation. Jedoch weisen die entsprechenden Polynome bei hohem Grad unerwünschte Oszillationen auf. Alternativ kann man eine Interpolationsfunktion auch stückweise aus Polynomen mit niedrigerem Grad zusammensetzen. Dies nennt sich dann Spline-Interpolation.

**Definition 6** (Spline). Man betrachte ein Intervall  $[t_1, t_n] \in \mathbb{R}$ , die Stützstellen  $t_1 < t_2 < \dots < t_n$  und die Stützwerte  $x_1, x_2, \dots, x_n$ .

Die Funktion  $s : [t_1, t_n] \rightarrow \mathbb{R}$  ist ein Spline von Grad  $g$  zu den Punkten  $(t_i, x_i)$ , wenn:

- (i)  $s$  eine Interpolationsfunktion zu den Punkten  $(t_i, x_i)$  ist.
- (ii)  $s|_{(t_i, t_{i+1})}$  für alle  $i = 1, \dots, n - 1$  ein Polynom vom Grad  $g$  ist.
- (iii)  $s$   $p$ -mal stetig differenzierbar ist.

**Bemerkung 7** (Splinegrad). Für den Splinegrad gibt es in unterschiedlicher Literatur unterschiedliche Notationen. In [7] entspricht der Splinegrad jeweils dem Polynomgrad. Somit sind lineare Splines vom Grad 1. Dies ist die Notation, die wir hier verwenden. In [9] und Wikipedia ist der Splinegrad jeweils eins höher, als der Polynomgrad. Dort sind

lineare Splines vom Grad 2. Die Bezeichnung als lineare, quadratische oder kubische Splines bleibt konsistent.

**Bemerkung 8** (Freiheitsgrad und Dimension des Splineraums). Typischerweise wird  $p = g - 1$  gewählt. Das heißt, Splines vom Grad  $g$  sind  $g - 1$  mal stetig differenzierbar. Wir betrachten ab hier nur diesen Fall.

$n - 1$  Polynome vom Grad  $g$  haben insgesamt  $(n - 1)(g + 1)$  Freiheitsgrade. Die  $g - 1$ -fache stetige Differenzierbarkeit zwischen den Stützstellen ist schon gegeben, da es sich dort um Polynome vom Grad  $g$  handelt. Lediglich an den Stützstellen  $t_2, \dots, t_{n-1}$  spielt diese Bedingung eine Rolle. Dies sind  $(n - 2)g$  Bedingungen.

Ein Spline vom Grad  $g$  hat dann insgesamt

$$(n - 1)(g + 1) - (n - 2)g = n + g - 1$$

Freiheitsgrade. Dies ist die Dimension des Raumes der Splines von Grad  $g$  zu den entsprechenden Stützstellen  $t_1 < t_2 < \dots < t_n$ , welcher im Folgenden als Splineraum bezeichnet.

### 3.2.1 Lineare Splines

Lineare Splines sind Splines vom Grad 1. Es handelt sich hierbei um einen Polygonzug, der die Punkte  $(t_i, x_i), i = 1, \dots, n$  verbindet. Wir betrachten das Funktional

$$R_1(f) = \int_{t_1}^{t_n} |f'(t)|^2 dt \quad (4)$$

für eine stückweise stetig differenzierbare Interpolationsfunktion  $f$  zu den Punkten  $(t_i, x_i)$ .

**Satz 9** (Minimallängen-Eigenschaft). Ist  $s$  der lineare Spline zu den Punkten  $(x_i, t_i)$ , so minimiert  $s$  das Funktional  $R_1$ .

**Beweis.** Seien  $f_1, f_2$  stückweise stetig differenzierbare Interpolationsfunktionen zu den Punkten  $(t_i, x_i)$  mit  $f_1 \neq f_2$  und  $\lambda \in (0, 1)$ . Dann folgt

$$\begin{aligned} R_1(\lambda f_1 + (1 - \lambda) f_2) &= \int_{t_1}^{t_n} |\lambda f_1'(t) + (1 - \lambda) f_2'(t)|^2 dt \\ &\leq \int_{t_1}^{t_n} |\lambda f_1'(t)|^2 dt + \int_{t_1}^{t_n} |(1 - \lambda) f_2'(t)|^2 dt \\ &< \lambda \int_{t_1}^{t_n} |f_1'(t)|^2 dt + (1 - \lambda) \int_{t_1}^{t_n} |f_2'(t)|^2 dt \\ &= \lambda R_1(f_1) + (1 - \lambda) R_1(f_2). \end{aligned}$$

Dabei wurde im zweiten Schritt verwendet, dass mindestens  $R_1(f_1) > 0$  oder  $R_1(f_2) > 0$ . Dies gilt, da sonst  $f_1$  und  $f_2$  beides konstante Interpolationsfunktionen zu denselben Punkten sind und damit identisch wären.

Somit ist  $R_1$  strikt konvex und der entsprechende Minimierer eindeutig.

Sei  $s$  ein linearer Spline zu den Punkten  $(t_i, x_i)$ . Dieser ist nach Bemerkung 8 eindeutig, da  $n + g - 1 = n$  und hier  $n$  Stützstellen gegeben sind. Sei weiter  $h$  eine beliebige stückweise stetig differenzierbare Interpolationsfunktion zu den Punkten  $(t_i, 0)$ . Dann ist  $s + \epsilon h$  für

$\epsilon \in \mathbb{R}$  eine stückweise stetig differenzierbare Interpolationsfunktion zu den Punkten  $(t_i, x_i)$ .  
Es folgt

$$\begin{aligned}
\left. \frac{d}{d\epsilon} R_1(s + \epsilon h) \right|_{\epsilon=0} &= 2 \int_{t_1}^{t_n} h'(t)(s'(t) + \epsilon h'(t)) dt \Big|_{\epsilon=0} \\
&= 2 \int_{t_1}^{t_n} h'(t)s'(t) dt \\
&= 2 \sum_{i=1}^{n-1} \int_{t_i}^{t_{i+1}} h'(t)s'(t) dt \\
&= 2 \sum_{i=1}^{n-1} \left( h(t)s'(t) \Big|_{t_i}^{t_{i+1}} - \int_{t_i}^{t_{i+1}} h(t)s''(t) dt \right) \\
&= 0.
\end{aligned}$$

Dabei gelten  $s''(t) = 0$ , da  $s$  ein Polynom ersten Grades ist und  $h(t_i) = 0$  für alle  $i = 1, \dots, n$ .

Damit ist  $s$  der eindeutige Minimierer von  $R_1$   $\square$ .

### 3.2.2 Kubische Splines

Kubische Splines sind Splines vom Grad 3. Für die Eigenschaften von diesen Splines gibt es eine physikalische Analogie: Spannt man eine Holzlatte zwischen durch Nägel festgelegte Punkte, so wird diese die Spannenergie natürlich minimieren. Die Holzlatte interpoliert dann diese Punkte mit minimaler Krümmung.

Mathematisch heißt das, dass sich die Holzlatte durch eine zweimal stückweise stetig differenzierbare Interpolationsfunktion  $f$  beschreiben lässt, die das Funktional

$$R_2(f) = \int_{t_1}^{t_n} |f''(t)|^2 dt \quad (5)$$

minimiert.

**Satz 10** (Minimalkrümmung-Eigenschaft). Sei  $s$  der kubische Spline zu den Punkten  $(x_i, t_i)$  und mit natürlichen Randwerten  $s''(t_1) = s''(t_n) = 0$ , so minimiert  $s$  das Funktional  $R_2$ .

**Beweis.** Seien  $f_1, f_2$  zweimal stückweise stetig differenzierbare Interpolationsfunktionen mit  $f_1 \neq f_2$  zu den Punkten  $(t_i, x_i)$  und  $\lambda \in (0, 1)$ . Dann folgt

$$\begin{aligned}
R_2(\lambda f_1 + (1 - \lambda)f_2) &= \int_{t_1}^{t_n} |\lambda f_1''(t) + (1 - \lambda)f_2''(t)|^2 dt \\
&\leq \int_{t_1}^{t_n} |\lambda f_1''(t)|^2 dt + \int_{t_1}^{t_n} |(1 - \lambda)f_2''(t)|^2 dt \\
&< \lambda \int_{t_1}^{t_n} |f_1''(t)|^2 dt + (1 - \lambda) \int_{t_1}^{t_n} |f_2''(t)|^2 dt \\
&= \lambda R_2(f_1) + (1 - \lambda)R_2(f_2).
\end{aligned}$$

Dabei wurde im zweiten Schritt verwendet, dass mindestens  $R_2(f_1) > 0$  oder  $R_2(f_2) > 0$ . Dies gilt, da sonst  $f_1$  und  $f_2$  beides lineare Interpolationsfunktionen zu denselben Punkten sind und damit identisch wären.

Somit ist  $R_2$  streng konvex und der entsprechende Minimierer eindeutig.

Sei  $s$  ein kubischer Spline zu den Punkten  $(t_i, x_i)$  und mit Randwerten  $s''(t_1) = s''(t_n) = 0$ . Dieser ist nach Bemerkung 8 eindeutig, da  $n + g - 1 = n + 2$  und  $n$  Stützstellen sowie zwei Randbedingungen gegeben sind. Sei weiter  $h$  eine zweimal stückweise stetig differenzierbare Interpolationsfunktion zu den Punkten  $(t_i, 0)$ . Dann ist  $s + \epsilon h$  für  $\epsilon \in \mathbb{R}$  eine zweimal stückweise stetig differenzierbare Interpolationsfunktion zu den Punkten  $(t_i, x_i)$ . Es folgt

$$\begin{aligned}
\frac{d}{d\epsilon} R_2(s + \epsilon h) \Big|_{\epsilon=0} &= 2 \int_{t_1}^{t_n} h''(t)(s''(t) + \epsilon h''(t)) dt \Big|_{\epsilon=0} \\
&= 2 \int_{t_1}^{t_n} h''(t)s''(t) dt \\
&= 2 \sum_{i=1}^{n-1} \int_{t_i}^{t_{i+1}} h''(t)s''(t) dt \\
&= 2 \sum_{i=1}^{n-1} \left( h'(t)s''(t) \Big|_{t_i}^{t_{i+1}} - \int_{t_i}^{t_{i+1}} h'(t)s'''(t) dt \right) \\
&= 2 \left( h'(t)s''(t) \Big|_{t_1}^{t_n} + \sum_{i=1}^{n-1} \left( \int_{t_i}^{t_{i+1}} h(t)s'''(t) dt - h(t)s'''(t) \Big|_{t_i}^{t_{i+1}} \right) \right) \\
&= 0.
\end{aligned}$$

Dabei gelten die natürlichen Randbedingungen  $s''(t_0) = s''(t_n) = 0$ . Zudem ist  $s'''(t) = 0$  für alle  $t$ , da  $s$  ein Polynom dritten Grades ist und  $h(t_i) = 0$  für alle  $i = 1, \dots, n$ . Damit ist  $s$  der eindeutige Minimierer von  $R_2$   $\square$ .

### 3.2.3 B-Splines

Splinefunktionen lassen sich als Linearkombination von entsprechenden Basisfunktionen, den sogenannten B-Splines, schreiben. Sie bilden damit eine Basis des entsprechenden Splineriums (Beweis folgt später). Dadurch lassen sich Rechnungen mit Splines vereinfachen.

**Definition 11** (B-Splines vom Grad 0). Für  $N$  B-Splines vom Grad 0 betrachten wir die Knotenpunkte  $k_1 \leq k_2 \leq \dots \leq k_{N+1}$ . Dann ist der  $i$ -te B-Spline stückweise konstant und definiert durch

$$B_{i,0}(t) = \begin{cases} 1, & \text{falls } k_i \leq t < k_{i+1} \\ 0, & \text{sonst} \end{cases} \quad (6)$$

für  $i = 1, \dots, N$ .

Diese Funktionen bilden eine Basis der Splines von Grad 0 zu den Stützstellen beziehungsweise Knotenpunkten  $k_i$ . Höhergradige Splines werden rekursiv definiert. Dabei stimmen die Knotenpunkte jedoch nicht mehr mit den Stützstellen überein.

**Definition 12** (B-Splines). Für  $N$  B-Splines vom Grad  $g$  betrachten wir die Knotenpunkte  $k_1 \leq k_2 \leq \dots \leq k_{N+g+1}$ . Der  $i$ -te B-Spline vom Grad  $g$  ist dann rekursiv definiert durch:

$$B_{i,g}(t) = \frac{t - k_i}{k_{i+g} - k_i} B_{i,g-1}(t) + \frac{k_{i+g+1} - t}{k_{i+g+1} - k_{i+1}} B_{i+1,g-1}(t) \quad (7)$$

für  $i = 1, \dots, N$ .

Dabei werden Brüche mit 0 im Nenner auf Null gesetzt. Dies wird durch das \* angedeutet.

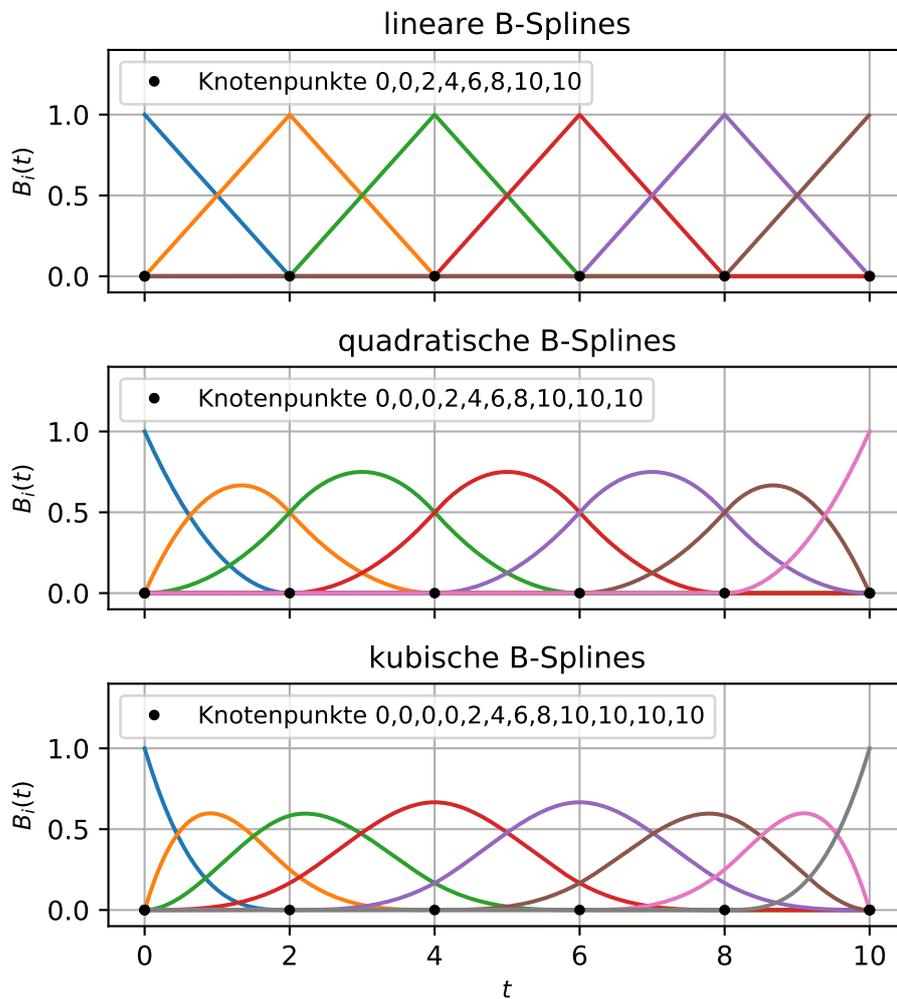


Abbildung 2: B-Splines  $B_i(t)$  für lineare, quadratische und kubische Splines zu den Stützstellen  $0, 2, 4, 6, 8, 10$ . Die jeweiligen Knotenpunkte wurden nach Definition 13 bestimmt.

**Definition 13** (Knotenpunkte). Um die Basis der Splinefunktionen zu den Stützstellen  $t_1 < t_2 < \dots < t_n$  zu erhalten, definiert man die Knotenpunkte  $k_1 \leq k_2 \leq \dots \leq k_{n+2g}$  durch

- (i)  $k_1, \dots, k_g := t_1$
- (ii)  $k_{g+i} := t_i$  für  $i = 1, \dots, n$
- (iii)  $k_{g+n+1}, \dots, k_{n+2g} := t_n$ .

In Abbildung 2 sind lineare, quadratische und kubische B-Spline Basisfunktionen und die entsprechend nach Definition 13 definierten Knotenpunkte zu den Stützstellen  $0, 2, 4, 6, 8, 10$  abgebildet. Man sieht, dass ein B-Spline  $B_{i,g}$  genau innerhalb des Intervalls  $(k_i, k_{i+g+1})$

ungleich Null ist. Dies erleichtert die Auswertung mit dem Computer und führt dazu, dass die später verwendeten Matrizen dünn besetzt sind.

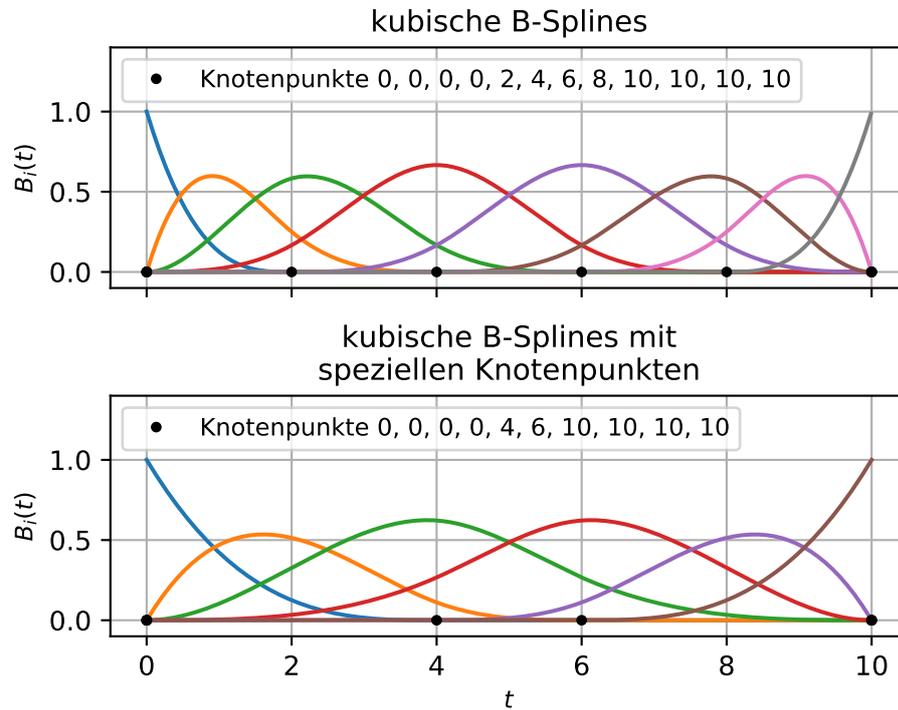


Abbildung 3: Kubische B-Splines  $B_i(t)$ , wobei die Knotenpunkte nach Definition 13 (oben) und nach Definition 14 (unten) gewählt wurden.

**Definition 14** (spezielle Knotenpunkte für kubische Splines). In [7] werden die Knotenpunkte kubischer Splines so definiert, dass die Stützstellen  $t_2$  und  $t_{n-1}$  weggelassen. Zu den Stützstellen  $t_1 < t_2 < \dots < t_n$  definiert man dann die Knotenpunkte  $k_1 \leq k_2 \leq \dots \leq k_{n+4}$  durch

- (i)  $k_1, \dots, k_4 := t_1$
- (ii)  $k_{4+i} := t_{2+i}$  für  $i = 1, \dots, n - 4$
- (iii)  $k_{n+1}, \dots, k_{n+4} := t_n$ .

Dadurch fallen zwei Freiheitsgrade weg und bei der Interpolation zu den Punkten  $(t_i, x_i)$  sind keine Randbedingungen mehr nötig. Wir betrachten nach dem nachfolgenden Satz nur diesen Fall. Es gilt dann  $N = n$ .

Ein Vergleich zwischen den Knotenpunkten nach Definition 13 und Definition 14 ist in Abbildung 3 zu sehen. Auch hier gilt, dass ein B-Spline  $B_{i,g}$  genau innerhalb des Intervalls  $(k_i, k_{i+g+1})$  ungleich Null ist.

**Satz 15** (B-Splines bilden eine Basis des entsprechenden Splineriums). Seien  $k_1 \leq \dots \leq k_{n+2g}$  die nach Definition 13 definierten Knotenpunkte zu den Stützstellen  $t_1 < \dots < t_n$ . Die B-Splines  $B_{i,g}$  zu diesen Knotenpunkten bilden dann eine Basis des Splineriums von Grad  $g$  zu den Stützstellen  $t_1 < \dots < t_n$ .

**Beweis.** Damit die B-Splines  $B_{i,g}$  eine Basis des entsprechenden Splineraums bilden, müssen folgende Aussagen gelten:

- (i) Alle  $B_{i,g}$  sind Splines vom Grad  $g$  zu den Stützstellen  $t_1 < \dots < t_n$ .
- (ii) Alle  $B_{i,g}$  sind linear unabhängig.
- (iii) Die Anzahl der  $B_{i,g}$  entspricht der Dimension des entsprechenden Splineraums.

Wir zeigen nun alle drei Aussagen.

- (i) Die B-Splines vom Grad 0 sind auf den entsprechenden Intervallen stückweise konstant und somit Splines vom Grad 0.  
B-Splines vom Grad  $g$  werden rekursiv durch B-Splines vom Grad  $g - 1$  definiert. Sind die  $B_{i,g-1}$  stückweise Polynome von Grad  $g - 1$  und  $g - 2$ -mal stetig differenzierbar, so sind die  $B_{i,g}$  aus (7) stückweise Polynome von Grad  $g$  und  $g - 1$ -mal stetig differenzierbar. Das  $*$  spielt dabei nur für die äußersten B-Splines eine Rolle, weshalb die  $g - 1$ -fache stetige Differenzierbarkeit auch dort gilt. Induktiv folgt die Behauptung für alle B-Splines.
- (ii) B-Splines  $B_{i,g}$  vom Grad  $g = 0$  sind genau innerhalb der Intervalle  $[k_i, k_{i+1})$  ungleich Null. Diese gehen nicht durch Schnitte oder Vereinigungen auseinander hervor. Somit sind die  $B_{i,0}$  linear unabhängig.  
B-Splines  $B_{i,g}$  vom Grad  $g > 0$  sind genau innerhalb der Intervalle  $(k_i, k_{i+g+1})$  ungleich Null. Diese gehen ebenfalls nicht durch Schnitte oder Vereinigungen auseinander hervor. Somit sind auch die  $B_{i,g}$  linear unabhängig.
- (iii) Die Dimension eines Splineraums mit  $n$  Stützstellen ist nach Bemerkung 8 gegeben durch  $n + g - 1$ . Die Anzahl an Knotenpunkten zu  $n$  Stützstellen ist nach Definition 14 gegeben durch  $n + 2g$  und nach der Definition der B-Splines  $N + g + 1$ . Es folgt  $N = n + 2g - g - 1 = n + g - 1$ . Die Anzahl  $N$  der B-Splines entspricht somit der Dimension des Splineraums.

Es folgt die Behauptung  $\square$ .

Satz 15 gilt ebenfalls für Knotenpunkte nach Definition 14. Es handelt sich dabei lediglich um B-Splines beziehungsweise einen Splineraum, bei denen die zwei gleichen ausgewählten Stützstellen fehlen. Ab hier verwenden wir nur noch diese Definition für die Knotenpunkte von kubischen B-Splines.

Wegen Satz 15 kann ein Spline  $s$  vom Grad  $g$  durch einen Koeffizientenvektor

$\vec{a} = (a_1, \dots, a_N)^T \in \mathbb{R}^N$  als

$$s(t) = s_{\vec{a}}(t) = \sum_{i=1}^N a_i B_i(t) \quad (8)$$

beschrieben werden.

Für  $t_j, j = 1, \dots, K$ , welche unabhängig von den Knotenpunkten sein können, und  $s_j = s(t_j)$  lässt sich dies als Matrix-Vektor-Produkt  $\vec{s} = \tilde{M} \vec{a}$  aufschreiben, wobei

$$\tilde{M} = (\tilde{M}_{j,i}) = (B_i(t_j)) \in \mathbb{R}^{K \times N} \quad (9)$$

und  $\vec{s} = (s_1, \dots, s_K)^T \in \mathbb{R}^K$ .

**Bemerkung 16** (Polynome). Ein Polynom vom Grad  $p$  auf dem Intervall  $[t_1, t_n]$  ist glatt und somit auch ein Spline vom Grad  $g \geq p$ . Dementsprechend lässt sich dieses nach Satz 15 durch einen Koeffizientenvektor  $\vec{a}$  als Linearkombination von B-Splines  $B_{i,g}$  schreiben. Ein Polynom vom Grad  $p$  hat  $p+1$  Freiheitsgrade. Damit hat der entsprechenden Koeffizientenvektor auch  $p+1$  Freiheitsgrade.

### 3.2.4 Dreidimensionale B-Splines

Im Fall eines dreidimensionalen Splines  $\vec{s}: [t_1, t_n] \rightarrow \mathbb{R}^3$  schreiben wir für den Koeffizientenvektor  $\vec{a} = (a_1^1, \dots, a_N^1, a_1^2, \dots, a_N^2, a_1^3, \dots, a_N^3)^T \in \mathbb{R}^{3N}$ , wobei

$$\vec{a}_i = \begin{pmatrix} a_i^1 \\ a_i^2 \\ a_i^3 \end{pmatrix}. \quad (10)$$

Dann ist

$$\vec{s}(t) = \vec{s}_{\vec{a}}(t) = \sum_{i=1}^N \vec{a}_i B_i(t). \quad (11)$$

Für  $t_j, j = 1, \dots, K$ , welche auch hier unabhängig von den Knotenpunkten sein können, lässt sich dies als Matrix-Vektor-Produkt  $\vec{s} = M\vec{a}$  aufschreiben, wobei

$$M = \begin{pmatrix} \tilde{M} & 0 & 0 \\ 0 & \tilde{M} & 0 \\ 0 & 0 & \tilde{M} \end{pmatrix} \in \mathbb{R}^{3K \times 3N} \quad (12)$$

und  $\vec{s} = (s_1^1, \dots, s_K^1, s_1^2, \dots, s_K^2, s_1^3, \dots, s_K^3)^T \in \mathbb{R}^{3K}$ . In den nachfolgenden Rechnungen werden wir zudem für  $j = 1, \dots, K$  die Schreibweise

$$\vec{s}_j = \begin{pmatrix} s_j^1 \\ s_j^2 \\ s_j^3 \end{pmatrix} = \begin{pmatrix} \tilde{M}_j & 0 & 0 \\ 0 & \tilde{M}_j & 0 \\ 0 & 0 & \tilde{M}_j \end{pmatrix} \vec{a} = M_j \vec{a} \quad (13)$$

verwenden, wobei  $\tilde{M}_j$  die  $j$ -te Zeile von  $\tilde{M}$  bezeichnet.  $M_j \in \mathbb{R}^{3 \times 3N}$  bezeichnet die entsprechende Blockmatrix aus den Zeilen  $\tilde{M}_j$ .

### 3.2.5 Approximation mit kubischen B-Splines in einer Dimension

Im Folgenden werden wir zwischen den Daten-Stützstellen  $t_j$  der Daten für  $j = 1, \dots, K$  und den  $N$  Spline-Stützstellen, auf denen die Knotenpunkte  $(k_1, k_{N+4})$  der  $N$  kubischen B-Splines beruhen, unterscheiden. Wählt man weniger Spline-Stützstellen als Daten-Stützstellen, so eignen sich B-Splines auch um gegebene Datenpunkte zu approximieren.

**Bemerkung 17** (Besetztheitsstruktur vom  $\tilde{M}$  und  $M$ ). Die Matrix  $\tilde{M}$  ist dünn besetzt und hat nur nahe einer „schiefen Diagonalen“ Einträge ungleich Null, da die B-Splines  $B_{i,g}$  außerhalb der Intervalle  $(k_i, k_{i+g+1})$  Null sind. Jede Spalte stellt dabei diskrete Funktionsauswertungen eines B-Splines dar. Aufgrund der Wahl der Knotenpunkte nach Definition 14 haben die vierte Spalte von links und von rechts bei äquidistanten Spline- und Daten-Stützstellen jeweils etwas mehr Einträge, als die anderen Spalten. Ein Beispiel für die Matrix  $\tilde{M}$  ist im Anhang in Abbildung 9 zu sehen.

Die Matrix  $M$  ist dementsprechend ebenfalls dünn besetzt.

In Abbildung 4 wurde eine Beispielapproximation durch kubischen B-Splines mit den äquidistanten Funktionsauswertungen  $f(t_j)$  für  $j_1, \dots, j_K$  mit  $K = 21$  der Funktion  $f(t) = \sin(t) + \frac{1}{2}t$  und  $N = 4, 6, 8$  äquidistanten Spline-Stützstellen durchgeführt. Dazu wurde die Zielfunktion

$$F(\vec{a}) = \frac{1}{K} \sum_{j=1}^K |s_{\vec{a}}(t_j) - f(t_j)|^2 = \frac{1}{K} \|\tilde{M}\vec{a} - \vec{f}\|^2 \quad (14)$$

minimiert, wobei  $\vec{f} = (f(t_1), \dots, f(t_K))^T \in \mathbb{R}^K$ . Die verwendete Norm  $\|\cdot\|$  ist hierbei die Frobeniusnorm.

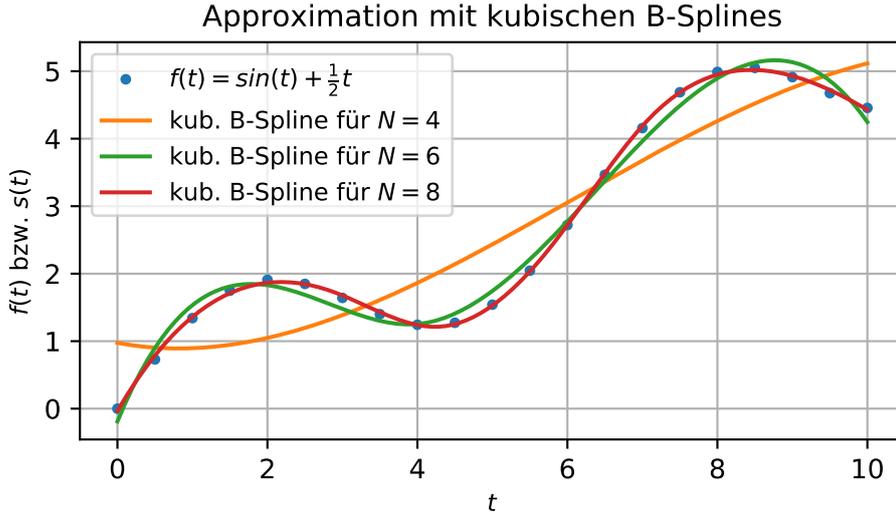


Abbildung 4: Approximation der Funktion  $f(t) = \sin(t) + \frac{1}{2}t$  mit 21 äquidistanten Daten-Stützstellen (blau) durch kubische B-Splines für  $N = 4$  (orange), 6 (grün) und 8 (rot) äquidistanten Spline-Stützstellen.

**Satz 18** (Hinreichende Bedingung für Injektivität von  $\tilde{M}$  und  $M$ ). Die Matrizen  $\tilde{M} \in \mathbb{R}^{K \times N}$  aus (9) und  $M \in \mathbb{R}^{3K \times 3N}$  aus (12) sind bei der Wahl der Knotenpunkte nach Definition 14 injektiv unter folgender Voraussetzung:

In jedem der Intervalle  $(k_{4+i}, k_{5+i})$ ,  $i = 1, \dots, N - 5$  liegen mindestens ein und in den Intervallen  $(k_4, k_5)$  und  $(k_N, k_{N+1})$  mindestens drei unterschiedliche Daten-Stützstellen  $t_j$  mit  $j \in \{1, \dots, K\}$ .

**Beweis.** Da wir nicht mehr Spline-Stützstellen als Daten-Stützstellen wählen, ist  $K \geq N$  und  $\tilde{M} \in \mathbb{R}^{K \times N}$  genau dann injektiv, wenn alle Spalten linear unabhängig sind.

Jede  $i$ -te Spalte für  $i = 1, \dots, N$  stellt dabei diskrete Auswertungen des B-Splines  $B_{i,3}$  zu den Zeiten  $t_j$  dar. Diese sind genau auf den Intervallen  $(k_i, k_{i+4})$  ungleich Null. Die  $i$ -te Spalte kann somit nur von den Spalten 3 weiter rechts oder 3 weiter links linear abhängig sein. Wird jeder B-Spline  $B_{i,3}$  in den Intervallen  $(k_i, k_{i+1})$  und  $(k_{i+3}, k_{i+4})$  ausgewertet, so ist die lineare Unabhängigkeit zu den anliegenden Spalten gegeben, da diese dort Null sind.

Für die jeweils 3 äußersten Spalten links und rechts gilt dies nicht, da diese keine 3 Spalten rechts beziehungsweise links von sich haben. Die lineare Unabhängigkeit voneinander, folgt dann aus der linearen Unabhängigkeit der zugehörigen B-Splines (siehe Satz 15) und drei unterschiedlichen Auswertungen in den Intervallen  $(k_4, k_5)$  beziehungsweise  $(k_N, k_{N+1})$ .

Es folgt die Injektivität von  $\tilde{M}$  und aus der Darstellung durch Blockmatrizen die auch Injektivität von  $M$   $\square$ .

Die Voraussetzungen von Satz 18 gelten für  $N < K$  und äquidistante Spline- und Daten-Stützstellen. Dies gilt für die hier verwendeten Stützstellen. Damit ist  $\tilde{M}$  injektiv. Es folgt für zwei Koeffizientenvektoren  $\vec{a}_1, \vec{a}_2 \in \mathbb{R}^N$  mit  $\vec{a}_1 \neq \vec{a}_2$  und  $\lambda \in (0, 1)$

$$\begin{aligned} F(\lambda\vec{a}_1 + (1-\lambda)\vec{a}_2) &= \frac{1}{K} \|\lambda(\tilde{M}\vec{a}_1 - \vec{f}) + (1-\lambda)(\tilde{M}\vec{a}_2 - \vec{f})\|^2 \\ &\leq \frac{1}{K} \|\lambda(\tilde{M}\vec{a}_1 - \vec{f})\|^2 + \frac{1}{K} \|(1-\lambda)(\tilde{M}\vec{a}_2 - \vec{f})\|^2 \\ &< \frac{\lambda}{K} \|\tilde{M}\vec{a}_1 - \vec{f}\|^2 + \frac{1-\lambda}{K} \|\tilde{M}\vec{a}_2 - \vec{f}\|^2 \\ &= \lambda F(\vec{a}_1) + (1-\lambda)F(\vec{a}_2). \end{aligned}$$

Dabei wurde im zweiten Schritt verwendet, dass  $\tilde{M}$  injektiv und mindestens  $F(\vec{a}_1)$  oder  $F(\vec{a}_2)$  größer Null sind, da  $\vec{a}_1 \neq \vec{a}_2$ .

Somit ist  $F$  streng konvex und der entsprechende Minimierer eindeutig. Ableiten und Nullsetzen liefert das lineare Gleichungssystem  $\tilde{M}^T \tilde{M} \vec{a} = \tilde{M}^T \vec{f}$ , welches sich mit dem Computer lösen lässt. Die Invertierbarkeit von  $\tilde{M}^T \tilde{M}$  folgt dabei aus der Injektivität von  $\tilde{M}$ . Die entsprechenden Ergebnisse für  $N = 4, 6, 8$  äquidistante Spline-Stützstellen sind in Abbildung 4 zu sehen. Es ist zu erkennen, dass auch für wenige Spline-Stützstellen im Verhältnis zu den Daten-Stützstellen gute Approximationen möglich sind. Dies hängt jedoch von den Eigenschaften der zu approximierenden Daten ab.

### 3.3 PET-Daten und Trajektorie der Zelle

PET-Daten liefern zu  $K$  Zeitpunkten  $t_j, j = 1, \dots, K$  die Positionen zweier Detektoren  $D_{1,j}, D_{2,j} \in \mathbb{R}^3$ , zwischen denen die „line of response“ verläuft. Die reale Trajektorie, die die Bewegung einer Zelle beschreibt, bezeichnen wir mit  $\vec{r}(t)$  und setzen  $\vec{x}_j = \vec{r}(t_j)$ .

Lineare und kubische Splines eignen sich aufgrund ihrer Eigenschaften gut, um die physikalischen Bewegungen einer Zelle in einem Organismus zu beschreiben.

Die Minimallängen-Eigenschaft einer zeitabhängigen Position  $x(t)$  entspricht dabei einer minimalen Geschwindigkeit und damit einer minimalen kinetischen Energie. Die Minimalkrümmungs-Eigenschaft entspricht einer minimalen Beschleunigung und damit einer minimalen Krafteinwirkung auf die Bewegung der Zelle. Solche Bewegungen sind im physikalischen Kontext wahrscheinlicher als Bewegungen mit hohen Geschwindigkeiten und Beschleunigungen.

Ersteres wird für die Regularisierungen in [5] und [8] verwendet. Zudem wurde laut [5] die Trajektorie  $\vec{r}(t)$  der Zelle mit kubischen Splines modelliert. Jedoch ist nicht ganz klar, ob es sich dabei auch wirklich um kubische Splines handelt, denn die entsprechende Abbildung 2 in diesem Artikel zeigt quadratische B-Splines (vgl. Abbildung 2). Dies entspricht nach der Dokumentation von B-Splines in Matlab [2] einem B-Spline vom Grad 3, da dort eine andere Notation verwendet wird. Siehe dazu Bemerkung 7. Nach dieser Notation hätten die gewünschten kubischen B-Splines den Grad 4. Es ist also möglich, dass es sich in dem gesamten Artikel [5] in Wahrheit nur um quadratische Splines handelt. Um dies zu überprüfen, müsste man den verwendeten Quellcode einsehen.

Wir werden im Folgenden die Rekonstruktion der Trajektorie  $\vec{r}(t)$  der Zelle mit einem dreidimensionalem kubischen Spline  $\vec{s}(t)$  der Form (11) modellieren. Dadurch verkleinern wir

bei  $N \ll K$  die Krümmung der Trajektorie. Zudem werden wir den Effekt Minimierung der Geschwindigkeit, sowie der Beschleunigung als Regularisierung untersuchen. Dies ist nur sinnvoll, da wir mindestens kubische Splines verwenden.

### 3.4 Zielfunktionen

Die Rekonstruktion der realen Trajektorie  $\vec{r}(t)$  werden wir durch das Minimieren von Zielfunktionen vornehmen, die angeben, wie gut die Rekonstruktion  $\vec{s}(t)$  zu den Daten passt, und wie physikalisch wahrscheinlich sie ist.

#### 3.4.1 Datenterm

Zunächst müssen wir bestimmen können, wie gut eine rekonstruierte Trajektorie  $\vec{s}(t)$  zu den Daten eines PET-Scans passt. Dazu wird in [5] der Datenterm  $I(\vec{a})$  folgendermaßen definiert.

Sei  $\vec{s}_{\vec{a}}(t)$  ein Spline der Form (11) zum Koeffizientenvektor  $\vec{a}$ .

Der Abstand von  $\vec{s}_j = M_j \vec{a} = \vec{s}_{\vec{a}}(t_j) \in \mathbb{R}^3$  zu der jeweilige „line of response“ zum Zeitpunkt  $t_j$  wird mit  $d_j(\vec{a})$  bezeichnet. Die „line of response“ verläuft parallel zu

$$\vec{u}_j = \frac{D_{2,j} - D_{1,j}}{\|D_{2,j} - D_{1,j}\|}. \quad (15)$$

Für den Abstand  $d_j(\vec{a})$  gilt dann

$$d_j^2(\vec{a}) = \|\vec{s}_j - D_{1,j}\|^2 - |\vec{u}_j \cdot (\vec{s}_j - D_{1,j})|^2. \quad (16)$$

Seien  $\vec{a}_1$  und  $\vec{a}_2$  Koeffizientenvektoren und  $\lambda \in [0, 1]$ . Sei weiter  $\vec{x}_1$  der nächste Punkt zu  $M_j \vec{a}_1$ , welcher auf der „line of response“ liegt und  $\vec{x}_2$  der nächste Punkt zu  $M_j \vec{a}_2$ , welcher auf der „line of response“ liegt. Der nächste Punkt zu  $M_j(\lambda \vec{a}_1 + (1 - \lambda) \vec{a}_2)$ , welcher auf der „line of response“ liegt ist dann  $\lambda \vec{x}_1 + (1 - \lambda) \vec{x}_2$ . Es folgt

$$\begin{aligned} d_j^2(\lambda \vec{a}_1 + (1 - \lambda) \vec{a}_2) &= \|\lambda M_j \vec{a}_1 + (1 - \lambda) M_j \vec{a}_2 - \lambda \vec{x}_1 - (1 - \lambda) \vec{x}_2\|^2 \\ &\leq \|\lambda (M_j \vec{a}_1 - \vec{x}_1)\|^2 + \|(1 - \lambda) (M_j \vec{a}_2 - \vec{x}_2)\|^2 \\ &\leq \lambda \|M_j \vec{a}_1 - \vec{x}_1\|^2 + (1 - \lambda) \|M_j \vec{a}_2 - \vec{x}_2\|^2 \\ &= \lambda d_j^2(\vec{a}_1) + (1 - \lambda) d_j^2(\vec{a}_2). \end{aligned}$$

Damit ist  $d_j^2(\vec{a})$  konvex.

Weiter ist der Datenterm definiert durch

$$I(\vec{a}) = \frac{1}{K} \sum_{j=1}^K d_j^2(\vec{a}). \quad (17)$$

als Summe von konvexen Funktionen ebenfalls konvex.

Minimiert man  $I(\vec{a})$ , so erhält man den B-Spline, der am besten zu den Daten passt. In [5] wird dieser Ausdruck so weiterverwendet und ein maximaler Abstand  $d_{\max}$  eingeführt, um Fehlmessungen zu ignorieren, die beispielsweise auftreten können, wenn zwei Photonen gemessen werden, welche nicht zum gleichen Ereignis gehören, oder auch zu stark abgelenkt wurden.

Diese Fälle werden wir in dieser Arbeit nicht behandeln. Stattdessen wollen wir  $I(\vec{a})$  in



**Beweis.** Ausgeschrieben ist

$$\begin{aligned}
& E - U^T U \\
& = \begin{pmatrix} 1 - (u_1^1)^2 & & -u_1^2 u_1^1 & & -u_1^3 u_1^1 & & \\ & \ddots & & \ddots & & \ddots & \\ -u_1^1 u_1^2 & & 1 - (u_K^1)^2 & & -u_K^2 u_K^1 & & -u_K^3 u_K^1 \\ & \ddots & & \ddots & & \ddots & \\ -u_1^1 u_1^3 & & -u_K^1 u_K^2 & & 1 - (u_K^2)^2 & & -u_K^3 u_K^2 \\ & \ddots & & \ddots & & \ddots & \\ & & -u_K^1 u_K^3 & & -u_K^2 u_K^3 & & 1 - (u_K^3)^2 \end{pmatrix} \\
& \in \mathbb{R}^{3K \times 3K}. \tag{24}
\end{aligned}$$

Die Invertierbarkeit wäre gegeben, wenn alle Spalten linear unabhängig sind. Aufgrund der Form der Matrix müsste dies nur für jeweils drei Spalten gezeigt werden. Es gilt jedoch

$$\begin{aligned}
& \det \begin{pmatrix} 1 - (u_j^1)^2 & -u_j^2 u_j^1 & -u_j^3 u_j^1 \\ -u_j^1 u_j^2 & 1 - (u_j^2)^2 & -u_j^3 u_j^2 \\ -u_j^1 u_j^3 & -u_j^2 u_j^3 & 1 - (u_j^3)^2 \end{pmatrix} \\
& = (1 - (u_j^1)^2)(1 - (u_j^2)^2)(1 - (u_j^3)^2) - 2(u_j^1)^2(u_j^2)^2(u_j^3)^2 \\
& \quad - (1 - (u_j^1)^2)(u_j^2)^2(u_j^3)^2 - (u_j^1)^2(1 - (u_j^2)^2)(u_j^3)^2 - (u_j^1)^2(u_j^2)^2(1 - (u_j^3)^2) \\
& = 1 - \left( (u_j^1)^2 + (u_j^2)^2 + (u_j^3)^2 \right) = 0
\end{aligned}$$

für alle  $j = 1, \dots, K$ . Dabei wurde verwendet, dass  $(u_j^1)^2 + (u_j^2)^2 + (u_j^3)^2 = 1$  aufgrund der Normierung in (15). Somit ist  $E - U^T U$  nicht invertierbar  $\square$ .

Wir brauchen also neben der notwendigen Bedingung, dass  $M$  injektiv ist, weitere Voraussetzungen, um Invertierbarkeit von  $A$  zu erhalten.

**Bemerkung 20** (Rang von  $E - U^T U$ ). Die Matrix  $E - U^T U$  hat Rang  $2K$ .

**Beweis.** Wir betrachten wieder die Darstellung (24). Aufgrund der Form der Matrix können je  $j = 1, \dots, K$  nur drei Spalten linear abhängig sein. Wie oben schon gezeigt ist dies für drei Spalten der Fall. Wir betrachten also nur je zwei Spalten und berechnen die nachfolgenden Determinanten.

Fall 1:  $u_j^1 \neq 0$ :

Es folgt wegen der Normierung aus (15), dass  $(u_j^2)^2 + (u_j^3)^2 \neq 1$  und weiter

$$\begin{aligned}
& \det \begin{pmatrix} 1 - (u_j^2)^2 & -u_j^3 u_j^2 \\ -u_j^2 u_j^3 & 1 - (u_j^3)^2 \end{pmatrix} \\
& = (1 - (u_j^2)^2)(1 - (u_j^3)^2) - (u_j^2)^2(u_j^3)^2 \\
& = 1 - (u_j^2)^2 - (u_j^3)^2 \neq 0.
\end{aligned}$$

Fall 2:  $u_j^3 \neq 0$ : Es folgt wegen der Normierung aus (15), dass  $(u_j^1)^2 + (u_j^2)^2 \neq 1$  und weiter

$$\begin{aligned} & \det \begin{pmatrix} 1 - (u_j^1)^2 & -u_j^2 u_j^1 \\ -u_j^1 u_j^2 & 1 - (u_j^2)^2 \end{pmatrix} \\ &= (1 - (u_j^1)^2)(1 - (u_j^2)^2) - (u_j^1)^2 (u_j^2)^2 \\ &= 1 - (u_j^1)^2 - (u_j^2)^2 \neq 0. \end{aligned}$$

Somit sind für alle  $j = 1, \dots, K$  jeweils genau zwei Spalten linear unabhängig und insgesamt genau  $2K$  Spalten. Es folgt die Behauptung  $\square$ .

Damit wissen wir, dass  $M$  nicht nur injektiv, sondern auch  $\text{Rang } M = 3N \leq 2K$  gelten muss, damit die Matrix  $A = M^T(E - U^T U)M$  invertierbar sein kann. Eine hinreichende Bedingung ist dies jedoch nicht.

**Satz 21** (Hinreichende Bedingung für Invertierbarkeit von  $M^T(E - U^T U)M$ ). Die Matrix  $M^T(E - U^T U)M$  ist invertierbar unter folgenden beiden Voraussetzungen:

- (i) In jedem der Intervalle  $(k_{4+i}, k_{5+i})$ ,  $i = 1, \dots, N - 5$  liegen mindestens ein und in den Intervallen  $(k_4, k_5)$  und  $(k_N, k_{N+1})$  mindestens drei unterschiedliche Paare an Daten-Stützstellen  $\{t_{j'}, t_{j''}\}$ ,  $j', j'' \in \{1, \dots, K\}$  mit  $j' \neq j''$ .
- (ii) Die zugehörigen „lines of response“ beziehungsweise Richtungsvektoren  $\vec{u}_{j'}$  und  $\vec{u}_{j''}$  aus (19) sind nicht parallel.

**Beweis.** Wir indizieren die oben genannten Paare mit  $\beta = 1, \dots, K'$  und nennen sie  $\{t_{j'_\beta}, t_{j''_\beta}\}$ . Dabei ist aufgrund der oben geforderten Anzahl an Paaren  $K' \geq N + 2$ .

Wir definieren die Summe der Quadratischen Abstände eines Paares durch

$$d_\beta^2(\vec{a}) = d_{j'_\beta}^2(\vec{a}) + d_{j''_\beta}^2(\vec{a}). \quad (25)$$

Im Allgemeinen ist dieser Term als Summe zweier konvexer Terme lediglich konvex, aber nicht streng konvex.

Wir definieren nun den Untervektorraum  $V_\beta = (\text{Kern}(M_{j'_\beta} - M_{j''_\beta}))$ . Dann gilt für alle  $\vec{a} \in V_\beta$

$$M_{j'_\beta} \vec{a} = M_{j''_\beta} \vec{a} \quad (26)$$

Seien nun  $\vec{a}_1, \vec{a}_2 \in V_\beta$  mit  $\vec{a}_1 \neq \vec{a}_2$  und  $\lambda \in (0, 1)$ . Die jeweils nächsten Punkte zu  $M_{j'_\beta} \vec{a}_1$  und  $M_{j'_\beta} \vec{a}_2$  auf der „line of response“ parallel zu  $\vec{u}_{j'_\beta}$  nennen wir  $\vec{x}'_1$  und  $\vec{x}'_2$ . Die jeweils nächsten Punkte zu  $M_{j''_\beta} \vec{a}_1$  und  $M_{j''_\beta} \vec{a}_2$  auf der „line of response“ parallel zu  $\vec{u}_{j''_\beta}$  nennen

wir  $\vec{x}'_1$  und  $\vec{x}''_2$ . Es folgt analog zum Beweis der Konvexität von  $d_j(\vec{a})$

$$\begin{aligned}
d_\beta^2(\lambda\vec{a}_1 + (1-\lambda)\vec{a}_2) &= d_{j'_\beta}^2(\lambda\vec{a}_1 + (1-\lambda)\vec{a}_2) + d_{j''_\beta}^2(\lambda\vec{a}_1 + (1-\lambda)\vec{a}_2) \\
&= \|\lambda M_{j'_\beta} \vec{a}_1 + (1-\lambda) M_{j'_\beta} \vec{a}_2 - \lambda \vec{x}'_1 - (1-\lambda) \vec{x}'_2\|^2 \\
&\quad + \|\lambda M_{j''_\beta} \vec{a}_1 + (1-\lambda) M_{j''_\beta} \vec{a}_2 - \lambda \vec{x}''_1 - (1-\lambda) \vec{x}''_2\|^2 \\
&\leq \|\lambda(M_{j'_\beta} \vec{a}_1 - \vec{x}'_1)\|^2 + \|(1-\lambda)(M_{j'_\beta} \vec{a}_2 - \vec{x}'_2)\|^2 \\
&\quad + \|\lambda(M_{j''_\beta} \vec{a}_1 - \vec{x}''_1)\|^2 + \|(1-\lambda)(M_{j''_\beta} \vec{a}_2 - \vec{x}''_2)\|^2 \\
&< \lambda \|M_{j'_\beta} \vec{a}_1 - \vec{x}'_1\|^2 + (1-\lambda) \|M_{j'_\beta} \vec{a}_2 - \vec{x}'_2\|^2 \\
&\quad + \lambda \|M_{j''_\beta} \vec{a}_1 - \vec{x}''_1\|^2 + (1-\lambda) \|M_{j''_\beta} \vec{a}_2 - \vec{x}''_2\|^2 \\
&= \lambda d_{j'_\beta}^2(\vec{a}_1) + (1-\lambda) d_{j''_\beta}^2(\vec{a}_2)
\end{aligned}$$

Dabei wurde im vierten Schritt verwendet, dass mindestens einer der vier Summanden größer als Null sein muss. Ansonsten, lägen sowohl  $M_{j'_\beta} \vec{a}_1$ , als auch  $M_{j''_\beta} \vec{a}_2$  auf beiden nicht parallelen „lines of response“.

Da die Richtungsvektoren  $\vec{u}_{j'_\beta}$  und  $\vec{u}_{j''_\beta}$  nicht parallel sind, gibt es keine zwei verschiedenen Punkte, die jeweils auf beiden zugehörigen „lines of response“ gleichzeitig liegen können. Maximal liegt einer von beiden im Schnittpunkt.

Damit wären  $M_{j'_\beta} \vec{a}_1 = M_{j''_\beta} \vec{a}_2$  und wegen  $\vec{a}_1, \vec{a}_2 \in V_\beta$  würde  $\vec{a}_1 = \vec{a}_2$  folgen. Letzteres wäre ein Widerspruch.

Somit ist  $d_\beta^2(\vec{a})$  beschränkt auf  $\vec{a} \in V_\beta$  streng konvex und

$$I'(\vec{a}) = \sum_{\beta=1}^{K'} d_\beta^2(\vec{a}) \quad (27)$$

beschränkt auf  $\text{Span}_{\beta=1}^{K'} V_\beta$  streng konvex.

Wir betrachten nun die Matrizen

$$M' = \begin{pmatrix} M_{j'_1} \\ \vdots \\ M_{j'_{K'}} \end{pmatrix} \in \mathbb{R}^{3K' \times 3N} \quad \text{und} \quad M'' = \begin{pmatrix} M_{j''_1} \\ \vdots \\ M_{j''_{K'}} \end{pmatrix} \in \mathbb{R}^{3K' \times 3N}. \quad (28)$$

Durch umsordieren der Zeilen erhalten wir ähnliche Matrizen

$$M' \sim \begin{pmatrix} \tilde{M}' & 0 & 0 \\ 0 & \tilde{M}' & 0 \\ 0 & 0 & \tilde{M}' \end{pmatrix} \in \mathbb{R}^{3K' \times 3N} \quad \text{und} \quad M'' \sim \begin{pmatrix} \tilde{M}'' & 0 & 0 \\ 0 & \tilde{M}'' & 0 \\ 0 & 0 & \tilde{M}'' \end{pmatrix} \in \mathbb{R}^{3K' \times 3N}. \quad (29)$$

wobei

$$\tilde{M}' = \begin{pmatrix} \tilde{M}_{j'_1} \\ \vdots \\ \tilde{M}_{j'_{K'}} \end{pmatrix} \in \mathbb{R}^{3K' \times 3N} \quad \text{und} \quad \tilde{M}'' = \begin{pmatrix} \tilde{M}_{j''_1} \\ \vdots \\ \tilde{M}_{j''_{K'}} \end{pmatrix} \in \mathbb{R}^{3K' \times 3N} \quad (30)$$

Dabei bezeichnet  $\sim$  die Ähnlichkeit zweier Matrizen.

Die letzten Beiden Matrizen sind nach Satz 18 und den oben genannten Voraussetzungen über die Existenz von Daten-Stützstellen in bestimmten Intervallen beide injektiv. Ihre jeweiligen Zeilen sind zudem paarweise linear Unabhängig, da sie Auswertungen der B-Splines zu den Zeiten  $t_{j'_\beta}$  und  $t_{j''_\beta}$  darstellen und diese nicht identisch sein können.

Daraus folgt, dass  $M' - M''$  injektiv ist und genau  $3N$  Zeilen von  $M' - M''$  linear unabhängig sind. Die lineare Hülle  $\text{Span}_{\beta=1}^{K'} V_\beta = \cup_{\beta=1}^{K'} (\text{Kern}(M_{j'_\beta} - M_{j''_\beta}))$  spannt dann den Vektorraum  $\mathbb{R}^{3N}$  auf.

Es folgt, dass  $I'(\vec{a})$  auf ganz  $\mathbb{R}^{3N}$  streng konvex ist. Da  $I(\vec{a})$  als Summe aus konvexen Funktionalen geschrieben werden kann, von denen mindestens  $I'(\vec{a})$  streng konvex ist, ist  $I(\vec{a})$  ebenfalls streng konvex. Es folgt die Existenz eines eindeutigen Minimierers und somit die Invertierbarkeit von  $M^T(E - U^T U)M$   $\square$ .

Sind die Daten, wie in Satz 21 beschrieben, dann ist der Minimierer von  $I(\vec{a})$  eindeutig. Es sind  $K \geq 2(N + 2)$  Daten-Stützstellen nötig, um diese Hinreichende Bedingung zu erfüllen. Es gilt dann auch die Notwendige Bedingung  $\text{Rang } M = 3N \leq 3K/2 \leq 2K$ , wie weiter oben gefordert. Theoretisch wären auch Werte von  $3N/2 \leq K \leq (N + 2)$  denkbar, jedoch haben wir dafür keine Hinreichenden Kriterien. In der späteren Anwendung wählen wir  $N \ll K$ , damit die Invertierbarkeit bei zufällig verteilten Daten-Stützstellen wahrscheinlich gegeben ist. Garantieren können wir sie damit nicht.

### 3.4.2 Regularisierungsterm zur Minimierung der Geschwindigkeit

Wenn die Voraussetzungen für die Eindeutigkeit des Minimums von  $I(\vec{a})$  nicht gegeben sind, kann man eine Regularisierung einführen. Dies tritt beispielsweise ein, wenn es Intervalle  $(k_i, k_{i+g+1})$  gibt, in denen keine Messungen liegen. Die Koeffizienten zu dem entsprechenden B-Spline sind dann entkoppelt [5]. Ein weiterer Grund für eine Regularisierung ist, dass die gemessenen „lines of response“ aufgrund des Positronenweges und Streuung der Photonen nicht auf der realen Trajektorie liegen. Eine Regularisierung kann helfen, Trajektorien zu bestimmen, die nicht nur nah an den Daten, sondern auch physikalisch wahrscheinlich sind.

Wählt man  $N \ll K$  und äquidistante Spline-Stützstellen zu zufälligen Daten-Stützstellen, minimiert man die Wahrscheinlichkeit, dass dieser Fall eintritt und erzwingt zudem eine indirekte Regularisierung, welche bei der Wahl von kubischen Splines die Krümmung minimiert. Alternativ kann man einen Regularisierungsterm  $R(\vec{a})$  einführen, der konvex ist und versuchen so eine Eindeutigkeit des Minimierers zu erhalten. In [5] wurde dazu der Term

$$R(\vec{a}) = \frac{1}{N} \sum_{i=1}^{N-1} \|\vec{a}_{i+1} - \vec{a}_i\|^2 \quad (31)$$

eingeführt und mit einem Relaxationsparameter  $\lambda > 0$  zum Datenterm addiert. Die neue Zielfunktion lautet dann

$$J(\vec{a}) = I(\vec{a}) + \lambda R(\vec{a}). \quad (32)$$

(Strenggenommen wurde in [5] eine abgeänderte Version von  $I(\vec{a})$  verwendet.)

Die Regularisierung im neuen Term führt dazu, dass kürzere Trajektorien bevorzugt werden, wenn in einem Intervall  $(k_i, k_{i+g+1})$  keine Messungen liegen. Diese Eigenschaft wollen wir analog zu (4) durch minimieren der Länge erreichen. Wir definieren den Regularisierungsterm

$$R_1(\vec{a}) = \int_{t_1}^{t_K} \|\vec{s}'_{\vec{a}}(t)\|^2 dt. \quad (33)$$

Dieser minimiert die Geschwindigkeit und bevorzugt damit physikalisch wahrscheinlichere Zellbewegungen. Ohne weitere Bedingungen an  $\vec{a}$  ist  $R_1(\vec{a})$  jedoch nicht streng konvex, da alle konstanten dreidimensionalen Splines eine Nullstelle und damit globale Minimierer

von  $R_1 \vec{a}$  sind. Nach Bemerkung 16 haben die entsprechenden Koeffizientenvektoren 3 Freiheitsgrade.  $R_1$  hat damit einen dreidimensionalen Raum an globalen Minimierern. Es gilt lediglich Konvexität, da

$$\begin{aligned} R_1(\lambda \vec{a}_1 + (1 - \lambda) \vec{a}_2) &= \int_{t_1}^{t_K} \|\lambda \vec{s}'_{\vec{a}_1}(t) + (1 - \lambda) \vec{s}'_{\vec{a}_2}(t)\|^2 dt \\ &\leq \int_{t_1}^{t_K} \|\lambda \vec{s}'_{\vec{a}_1}(t)\|^2 dt + \int_{t_1}^{t_K} \|(1 - \lambda) \vec{s}'_{\vec{a}_2}(t)\|^2 dt \\ &\leq \lambda \int_{t_1}^{t_K} \|\vec{s}'_{\vec{a}_1}(t)\|^2 dt + (1 - \lambda) \int_{t_1}^{t_K} \|\vec{s}'_{\vec{a}_2}(t)\|^2 dt \\ &= \lambda R_1(\vec{a}_1) + (1 - \lambda) R_1(\vec{a}_2). \end{aligned}$$

für Koeffizientenvektoren  $\vec{a}_1, \vec{a}_2$  und  $\lambda \in [0, 1]$ .

Um  $R_1$  mit dem Computer berechnen zu können, bringen wir  $R_1(\vec{a})$  in Matrixform. Es folgt

$$\begin{aligned} R_1(\vec{a}) &= \int_{t_1}^{t_K} \|\vec{s}'_{\vec{a}}(t)\|^2 dt = \int_{t_1}^{t_K} \left\| \sum_{i=1}^N \vec{a}_i B_i(t) \right\|^2 dt \\ &= \sum_{i=1}^N \sum_{l=1}^N \vec{a}_i \int_{t_1}^{t_K} B_i'(t) B_l'(t) dt \vec{a}_l = \vec{a} L_1 \vec{a}. \end{aligned} \quad (34)$$

wobei

$$L_1 = \begin{pmatrix} \tilde{L}_1 & 0 & 0 \\ 0 & \tilde{L}_1 & 0 \\ 0 & 0 & \tilde{L}_1 \end{pmatrix} \text{ mit } \tilde{L}_1 = (\tilde{L}_{1,i,l}) = \left( \int_{t_1}^{t_K} B_i'(t) B_l'(t) dt \right) \in \mathbb{R}^{N \times N}. \quad (35)$$

Da ein kubischer B-Spline  $B_{i,3}(t)$  nur auf dem Intervall  $(k_i, k_{i+4})$  ungleich Null ist, gilt dies auch für die Ableitungen. Die Matrizen  $\tilde{L}_1$  und  $L_1$  haben somit Siebenbandgestalt und sind dünn besetzt. Ein Beispiel für eine Matrix  $\tilde{L}_1$  findet sich in Abbildung 35 im Anhang.

Ist der Minimierer von  $I(\vec{a})$  aus den weiter oben genannten Gründen nicht eindeutig, so eignet sich die Addition von  $\lambda R_1(\vec{a})$  mit einem Relaxationsparameter  $\lambda > 0$ , um den Raum der Minimierer zu verkleinern. Wir erhalten eine neue Zielfunktion

$$J_1(\vec{a}) = I(\vec{a}) + \lambda R_1(\vec{a}). \quad (36)$$

Diese ist mindestens konvex, da  $I(\vec{a})$  und  $R_1(\vec{a})$  konvex sind. Die Eindeutigkeit eines Minimierers ist auch hier nicht garantiert. In der Anwendung ist jedoch davon auszugehen, dass  $I(\vec{a})$  den dreidimensionalen Raum der Minimierer von  $R_1(\vec{a})$  durch Randbedingungen verkleinert, wodurch der Minimierer von  $J_1(\vec{a})$  eindeutig wird.

Ableiten und Nullsetzen liefert

$$\frac{\partial}{\partial \vec{a}} J_1(\vec{a}) = \frac{\partial}{\partial \vec{a}} I(\vec{a}) + \lambda \frac{\partial}{\partial \vec{a}} R_1(\vec{a}) = 0 \quad (37)$$

und damit das Gleichungssystem

$$\left( \frac{2}{K} M^T (E - U^T U) M + \lambda 2 L_1 \right) \vec{a} = \left( \frac{2}{K} M^T (E - U^T U) \right) \vec{d}. \quad (38)$$

Dieses lässt sich unter der Annahme, dass der Minimierer eindeutig und somit die entsprechende Matrix invertierbar ist, mit dem Computer lösen.

### 3.4.3 Regularisierungsterm zur Minimierung der Beschleunigung

Weiter definieren wir analog zu (5) den Regularisierungsterm:

$$R_2(\vec{a}) = \int_{t_1}^{t_K} \|\vec{s}''(t)\|^2 dt = \vec{a} L_2 \vec{a}, \quad (39)$$

wobei

$$L_2 = \begin{pmatrix} \tilde{L}_2 & 0 & 0 \\ 0 & \tilde{L}_2 & 0 \\ 0 & 0 & \tilde{L}_2 \end{pmatrix} \text{ mit } \tilde{L}_2 = (\tilde{L}_{2,i,l}) = \left( \int_{t_1}^{t_K} B_i''(t) B_l''(t) dt \right) \in \mathbb{R}^{N \times N}. \quad (40)$$

Dieser minimiert die Beschleunigung und bevorzugt analog zu (34) physikalisch wahrscheinlichere Zellbewegungen. Die Matrix  $L_2$  hat dabei wie  $L_1$  Siebenbandgestalt und ist dementsprechend dünn besetzt. Ein Beispiel ist im Anhang in Abbildung 40 zu sehen. So wie  $R_1(\vec{a})$  lediglich konvex, aber nicht streng konvex ist, gilt dies auch für  $R_2(\vec{a})$ . Die Rechnung verläuft analog.

Der Raum der Minimierer ist hier sechsdimensional. Dies folgt daraus, dass alle dreidimensionalen linearen Splines eine Nullstelle und somit ein globaler Minimierer von  $R_2(\vec{a})$  sind. Die entsprechenden Koeffizientenvektoren haben dann nach Bemerkung 16 sechs Freiheitsgrade.

Addieren wir ebenfalls diese Regularisierung mit einem Relaxationsparameter  $\lambda > 0$  zu dem Datenterm  $I(\vec{a})$ , so erhalten wir eine weitere neue Zielfunktion

$$J_2(\vec{a}) = I(\vec{a}) + \lambda R_2(\vec{a}). \quad (41)$$

Diese ist ebenfalls konvex aber nicht zwingend streng konvex. Die Eindeutigkeit eines Minimierers ist auch hier nicht garantiert. In der Anwendung ist jedoch davon auszugehen, dass  $I(\vec{a})$  den sechsdimensionalen Raum der Minimierer von  $R_2(\vec{a})$  durch Randbedingungen verkleinert, wodurch der Minimierer von  $J_2(\vec{a})$  eindeutig wird.

Ableiten und Nullsetzen liefert

$$\frac{\partial}{\partial \vec{a}} J_2(\vec{a}) = \frac{\partial}{\partial \vec{a}} I(\vec{a}) + \lambda \frac{\partial}{\partial \vec{a}} R_2(\vec{a}) = 0 \quad (42)$$

und damit das Gleichungssystem

$$\left( \frac{2}{K} M^T (E - U^T U) M + \lambda 2L_2 \right) \vec{a} = \left( \frac{2}{K} M^T (E - U^T U) \right) \vec{d}. \quad (43)$$

Auch dieses lässt sich unter der Annahme, dass der Minimierer eindeutig und somit die entsprechende Matrix invertierbar ist, mit dem Computer lösen.

## 4 Methoden

Die Umsetzung der Theorie und die Simulation der PET-Daten findet in Python in Form eines beigefügten Jupyter-Notebooks statt.

## 4.1 Simulieren von PET-Daten

In [5] und [8] wird das GATE Monte Carlo Paket [1] verwendet, um die PET-Daten eines Siemens Inveon microPET Scanners zu simulieren. Im Folgenden werden wir uns einfache PET-Daten selbst konstruieren und anschließend externe simulierte PET-Daten aus [8] verwenden.

Zunächst überlegen wir uns eine angemessene Trajektorie für eine Zelle. In [4],[5] und [8] werden neben komplizierteren Zellbewegungen einfache Kreisbewegungen in einer Ebene betrachtet. In den Daten von [8], die wir im Anschluss verwenden, ist die Trajektorie gegeben durch

$$\vec{r}(t) = r \cdot \begin{pmatrix} \cos\left(\frac{2\pi}{t_{\max}} t\right) \\ \sin\left(\frac{2\pi}{t_{\max}} t\right) \\ 0 \end{pmatrix}. \quad (44)$$

Dabei wird eine Messung im Zeitintervall  $[0, t_{\max}]$  mit  $t_{\max} = 120$  s betrachtet. Der Radius der Bewegung ist  $r = 60$  mm. Diese Trajektorie eignet sich gut dazu eine physikalisch wahrscheinliche Bewegung zu simulieren und die Regularisierungsterme  $R_1$  und  $R_2$  zu testen, da Geschwindigkeit und Beschleunigung beschränkt sind.

Für den Aufbau des PET-Scanners betrachten wir zylindrisch um den Ursprung angeordnete Detektoren. Der Zylinder hat hier eine Höhe von  $h = 200$  mm und einen Radius von  $R = 400$  mm. Dies entspricht den Maßen des simulierten PET-Scanners aus den später verwendeten PET-Daten von [8]. In diesem Zylinder positionieren wir  $8 \times 100$  Detektoren (vgl. Abbildung 1). Dies ist eine relativ schlechte Auflösung, macht aber die Effekte der Regularisierung deutlicher.

Der PET-Scanner, der in [5] mit dem GATE Monte Carlo Paket simuliert wird hat eine Sensitivität von 8,4%. Betrachten wir die Länge des hier verwendeten Zeitintervalls von  $t_{\max} = 120$  s und eine nach [5] realistische Aktivität von bis zu 10 Bq, so sollten wir im Durchschnitt  $0,084 \cdot 10 \text{ Bq} \cdot 120 \text{ s} \approx 100 = K$  Zerfälle messen. Wir wählen also mit Python 100 gleichverteilte Zeitpunkte  $t_1, \dots, t_K$  aus dem Intervall  $[0, t_{\max}]$  aus.

Zu jedem Zeitpunkt  $t_j$  erhalten wir die Position  $\vec{x} = \vec{r}(t_j)$  der Positronenemission. Dazu addieren wir je Koordinate einen normalverteilten Wert zur Standardabweichung  $\sigma$ , um den zurückgelegten Weg des Positrons bis zur Annihilation mit einem Elektron zu simulieren. Die Weglänge eines realen emittierten Positrons hängt vom Material ab, durch das es sich bewegt. In Wasser ist diese beim Zerfall von  $^{18}\text{F}$  durchschnittlich 0,6 mm und maximal 2,4 mm [3]. Um auch hier den Effekt der Regularisierung deutlicher zu machen, wählen wir eine deutlich größeren Standardabweichung der Weglänge von  $\sqrt{3}\sigma$  und setzen  $\sigma = 10$  mm. Den Ort der Annihilation nennen wir  $\vec{y} = (y_1, y_2, y_3)^T$ .

Anschließend müssen wir einen zufälligen normierten Vektor  $\vec{v} = (v_1, v_2, v_3)^T$  wählen, der beschreibt, in welche Richtung sich die bei der Annihilation entstandenen Photonen bewegen. Dazu reicht es nicht, drei gleichverteilte Zufallswerte  $v'_1, v'_2, v'_3 \in [-1, 1]$  zu wählen und anschließend  $\vec{v}'$  zu normieren, da so Richtungen nahe der Diagonalen wahrscheinlicher sind als Richtungen nahe der Achsen. Um dies zu verhindern, wählen wir solange gleichverteilte  $v'_1, v'_2, v'_3 \in [-1, 1]$ , bis  $\|\vec{v}'\| < 1$ . Danach wird  $\vec{v}'$  zu  $\vec{v} = \vec{v}'/\|\vec{v}'\|$  normiert. Mit dieser Methode erhält man Radialsymmetrie und gleichverteilte Punkte auf der Einheitskugel (siehe Abbildung 5).

Anschließend wird der Schnittpunkt, der durch die Position  $\vec{y}$  und die Richtung  $\vec{v}$  beschriebenen „line of response“, mit dem Zylinder bestimmt, auf dem die Detektoren liegen. Dazu betrachten wir zunächst die  $x_1$ - $x_2$ -Ebene und nehmen an, dass sich  $(y_1, y_2)^T$  innerhalb ei-

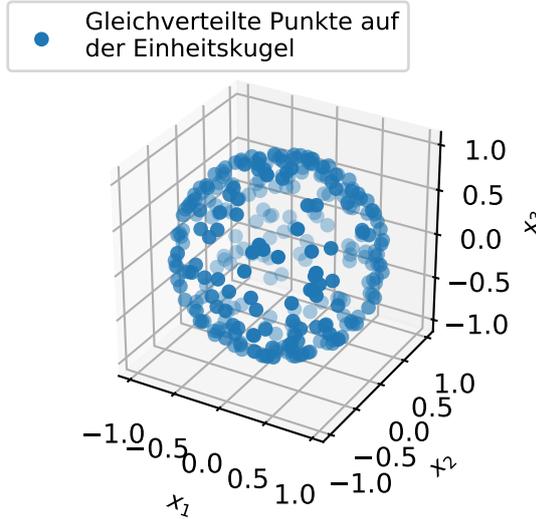


Abbildung 5: Gleichverteilte normierte Richtungsvektoren als Punkte auf der Einheitskugel.

nes Kreises um den Ursprung mit dem Zylinderradius  $R$  befindet. Wir lösen die Gleichung

$$(y_1 + \lambda v_1)^2 + (y_2 + \lambda v_2)^2 = R^2 \quad (45)$$

nach  $\lambda$  auf und erhalten

$$\lambda_{1,2} = -\frac{v_1 y_1 + v_2 y_2}{v_1^2 + v_2^2} \pm \sqrt{\left(\frac{v_1 y_1 + v_2 y_2}{v_1^2 + v_2^2}\right)^2 - \frac{y_1^2 + y_2^2 - R^2}{v_1^2 + v_2^2}}. \quad (46)$$

Die Punkte, an denen die „line of response“ den Zylinder schneidet, liegen somit bei  $\vec{y} + \lambda_1 \vec{v}$  und  $\vec{y} + \lambda_2 \vec{v}$ . Liegen diese außerhalb des definierten Zylinders, so wird der gesamte Prozess wiederholt. Durch passendes Runden der entsprechenden Zylinderkoordinaten erhält man dann die Positionen der nächstliegenden Detektoren. Das Ergebnis dieses Vorgangs ist für  $\sigma = 100$  in Abbildung 1 zu sehen. Das Ergebnis für einen vollständigen PET-Scan mit 100 Messungen ist in Abbildung 6 zu sehen.

Unter realen Bedingungen werden die Photonen auf dem Weg zum Detektor durch die umgebende Materie abgelenkt. Dies kann dazu führen, dass die Annihilation von Positron und Elektron nicht mehr auf der „line of response“ liegt. Bei schwacher Streuung ist die Abweichung minimal und kann in unserem Fall durch das relativ große  $\sigma$  simuliert werden. Starke Streuung kann hingegen als zufällige Messung von Photonen an zwei zufälligen Detektoren angesehen werden (vgl. [8]). Diesen Fall simulieren wir hier nicht, er kann jedoch in den später verwendeten externen PET-Daten auftreten.

## 4.2 Fehlerabschätzung

Den Rekonstruktionsfehler error einer Rekonstruktion  $\vec{s}(t)$  definieren wir wie in [5] durch

$$\text{error}^2 = \frac{1}{Q} \sum_{k=1}^Q \|\vec{r}(\theta_k) - \vec{s}(\theta_k)\|^2, \quad (47)$$

wobei  $\theta_k, k = 1, \dots, Q$  äquidistante Zeitpunkte auf dem Intervall  $[k_0, k_N + 4]$  sind. Es reicht hier nicht, das Intervall  $[0, t_{\max}]$  zu wählen, da bei der Wahl der Daten-Stützstellen als

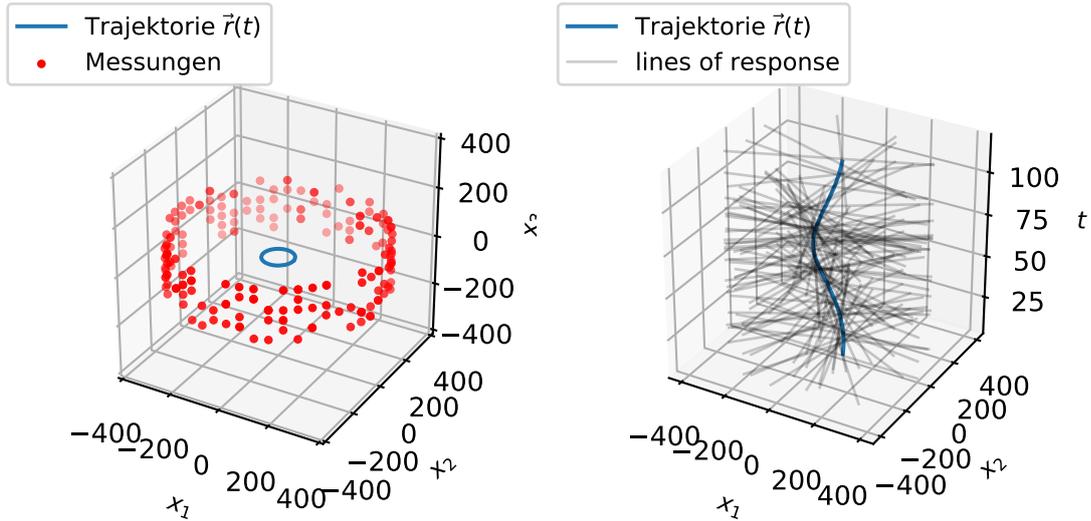


Abbildung 6: Selbstgenerierte PET-Daten. Links: Die Positionen der Detektoren, welche die  $\beta$ -Zerfälle messen (rot) und die Trajektorie  $\vec{r}(t)$  der Zelle (blau) im  $x_1$ - $x_2$ - $x_3$ -Raum. Rechts: Die „lines of response“ (schwarz) und die Trajektorie  $\vec{r}(t)$  der Zelle (blau) im  $x_1$ - $x_2$ - $t$ -Raum.

Spline-Stützstellen die Splines außerhalb dieser nicht definiert sind. Da wir nur PET-Daten mit 100 Daten-Stützstellen betrachten, reicht es in unserem Fall aus wie in [5]  $Q = 5000$  zu setzen.

### 4.3 Rekonstruktion der Trajektorie der Zelle aus PET-Daten

Um die Trajektorie  $\vec{r}(t)$  durch einen dreidimensionalen Spline  $\vec{s}(t)$  zu rekonstruieren, verwenden wir die Zielfunktionen  $I(\vec{a})$ ,  $J_1(\vec{a})$  und  $J_2(\vec{a})$  aus der Theorie.

Für die B-Splines verwenden wir die Methode `BSpline` aus dem Python Paket `scipy.interpolate`. Weiter sind die Matrizen  $M$ ,  $U$ ,  $L_1$  und  $L_2$  dünn besetzt, weshalb diese mit dem Paket `scipy.sparse` erstellt werden, um den Rechenaufwand zu minimieren. Entsprechend werden die Gleichungssysteme (23), (38) und (43) mit der Methode `scipy.sparse.linalg.spsolve` gelöst.

Eine Besonderheit stellen die Matrizen  $L_1$  und  $L_2$  dar. Die Ausdrücke (35) und (40) können theoretisch analytisch gelöst werden, da es sich bei den B-Splines um stückweise Polynome handelt. Bei der ersten und zweiten Ableitung ist dies jeweils mit der Methode `.derivative()` möglich. Das anschließende Integral berechnen wir numerisch mit `scipy.integrate.quad`. Dabei nutzen wir aus, dass die Matrizen  $\tilde{L}_1$  und  $\tilde{L}_2$  Fünfbandgestalt haben und die B-Splines  $B_{i,3}(t)$  und ihre Ableitungen außerhalb des Intervalls  $[k_i, k_{i+4}]$  Null sind.

## 5 Ergebnisse und Diskussion

### 5.1 Datenterm

Um den Datenterm  $I(\vec{a})$  zu testen, wählen wir deutlich weniger Spline-Stützstellen als Daten-Stützstellen, damit zwischen den Spline-Stützstellen möglichst viele Messungen liegen und der Minimierer wahrscheinlich eindeutig und die Matrix  $M^T(E - U^T U)M$  invertierbar ist.

Wir testen jeweils 10 verschiedene PET-Datensätze für jeweils  $N = 4, 8, 12, 16, 20$  äquidistante Spline-Stützstellen und berechnen den durchschnittlichen Rekonstruktionsfehler (siehe Abbildung 11 oben). Dieser ist bei  $N = 8$  minimal und liegt bei 13,375 mm. Unterhalb davon scheint die Auflösung des Splineraums nicht gut genug zu sein, um die Trajektorie  $\vec{r}(t)$  abzubilden. Oberhalb davon wird der Effekt der indirekten Regularisierung durch die Darstellung als kubische Splines kleiner. Es stehen immer mehr Freiheitsgrade zur Verfügung, um die Messfehler abzubilden, die durch die Wege der Positronen entstehen. Dies ist unten in Abbildung 11 im Anhang sichtbar. Die Rekonstruierten Trajektorien haben bei höherem  $N$  mehr Ausreißer.

Eine Wahl von wenigen Spline-Stützstellen hat den Effekt, über mehrere Datenpunkte zu mitteln, und somit Messfehler zu reduzieren. Dies ist ähnlich zur PEPT aus [4], nur dass hier mit kubischen Splines gearbeitet wird. Wir nennen dies indirekte Regularisierung. Oberhalb von  $N = 30$  wird der Rekonstruktionsfehler vierstellig und höher. In diesem Fall scheinen zu wenig Messungen zwischen den Spline-Stützstellen zu liegen. Dementsprechend werden dort die Regularisierungsterme  $R_1(\vec{a})$  und  $R_2(\vec{a})$  notwendig.

## 5.2 Regularisierung durch Minimierung der Geschwindigkeit

Zunächst wird die Zielfunktion  $J_1(\vec{a})$  für jeweils 10 verschiedenen PET-Datensätze für jeweils  $\lambda = 10^{-3}, \dots, 10^1$  getestet. Wir kombinieren den Regularisierungseffekt einer kleinen Zahl von  $N = 20$  Spline-Stützstellen und einer Minimierung der Geschwindigkeit. Dies entspricht dem vorgehen aus [5], da dort im Schnitt 5 Daten-Stützstellen zwischen die Knotenpunkte der B-Splines gelegt wurden. Die Ergebnisse sind in Abbildung 12 im Anhang zu sehen. Die optimale Wahl von  $\lambda$  liegt hier in der Größenordnung von  $10^1$ . Der durchschnittliche Rekonstruktionsfehler kann so auf 13,288 mm reduziert werden, was eine Verbesserung zum entsprechenden Versuch ohne Regularisierung darstellt. Ist  $\lambda$  zu klein, so hat die Rekonstruktion immer mehr Ausreißer, ist  $\lambda$  zu groß, so kürzt der Algorithmus Wege zu stark ab. Dies ist bei der roten Kurve aus Abbildung 12 unten zu sehen. Die rekonstruierte Spiralbewegung hat einen kleineren Radius, als die reale Trajektorie. Dieses Verhalten ergibt Sinn, da  $R_1(\vec{a})$  die Länge minimiert.

Neben  $N = 20$  äquidistanten Stützstellen testen wir die Zielfunktion  $J_1(\vec{a})$  auch bei identischen Spline- und Daten-Stützstellen. Die Ergebnisse dazu sind in Abbildung 13 im Anhang zu sehen. Bei zu kleinem  $\lambda$  hat die Rekonstruktion zu viele Ausreißer, bei zu großem  $\lambda$  kürzt das Verfahren wieder ab. Die optimale Größenordnung liegt auch hier bei  $\lambda = 10^{-1}$ . Der durchschnittliche Rekonstruktionsfehler ist dann mit 14,613 mm etwas schlechter, als der bei den  $N = 20$  äquidistanten Stützstellen. Wahrscheinlich liegt dies daran, dass keine indirekte Regularisierung stattfindet.

## 5.3 Regularisierung durch Minimierung der Beschleunigung

Wir betrachten nun die Rekonstruktion der Trajektorie durch die Minimierung von  $J_2(\vec{a})$ . Wieder wählen wir erst  $N = 20$  äquidistante Spline-Stützstellen. Die Ergebnisse sind in Abbildung 14 im Anhang dargestellt. Der optimale Wert von  $\lambda$  liegt wieder in der Größenordnung von  $10^{-1}$ . Der Rekonstruktionsfehler beträgt dann im Durchschnitt 10,78 mm, was eine deutliche Verbesserung zum selben Szenario mit  $J_1(\vec{a})$  darstellt.

Im Weiteren betrachten wir wieder identische Spline- und Daten-Stützstellen. Die Ergebnisse dazu sind in Abbildung 15 im Anhang dargestellt. Die Rekonstruktion ist sichtbar schlecht und der durchschnittliche Rekonstruktionsfehler nie kleiner, als 59,82 mm. Ist  $\lambda$  zu groß, so wird die Rekonstruktion aufgrund der starken Regularisierung beinahe konstant. Ist  $\lambda$  zu klein, so hat ein Teil der Rekonstruktion Ausreißer, die nicht mehr gut zur realen Trajektorie passen, während ein Teil der Rekonstruktion immer noch beinahe konstant ist. Es scheint so, als würde die Regularisierung in unterschiedlichen Zeitabschnitten unterschiedlich stark wirken. Eine Erklärung dafür ist, dass durch die Wahl der Spline-Stützstellen als Daten-Stützstellen, die Intervalle  $(k_i, k_i + 1)$  unterschiedlich groß sind. Die entsprechenden Ableitungen und vor allem zweiten Ableitungen der B-Splines sind damit zu unterschiedlichen Zeiten unterschiedlich groß und die Regularisierung unterschiedlich stark.

Um das oben beschriebene Verhalten zu umgehen testen wir das Verfahren zusätzlich noch für  $N = 100$  äquidistante Spline-Stützstellen. Damit sollte kaum eine indirekte Regularisierung stattfinden, die Regularisierung durch den Term  $R_2(\vec{a})$  aber zu allen Zeiten gleichstark sein. Das Ergebnis dazu ist in Abbildung 16 im Anhang zu sehen. Der gewünschte Effekt tritt ein und die Regularisierung ist gleichmäßig und für  $\lambda$  in der Größenordnung von  $10^1$  optimal. Der Rekonstruktionsfehler ist dann durchschnittlich 10,795 mm und genauso gut, wie im Fall von  $N = 20$  Spline-Stützstellen. Hier findet zwar keine indirekte Regularisierung mehr statt, der Effekt ist jedoch in beiden Fällen eine Minimierung der Krümmung, da wir die  $\vec{s}(t)$  durch kubische Splines darstellen.

#### 5.4 Regularisierung durch Minimierung der Beschleunigung bei externen PET-Daten

Abschließend testen wir den Term  $J_2(\vec{a})$  noch für  $N = 20$  äquidistante Spline-Stützstellen an jeweils einem PET-Datensatz aus [8] mit schwacher und starker Streuung. Da die gegebenen Datensätze in 120s mehr als 100 Messungen liefern, wählen wir jeweils zufällig hundert davon aus, um die Ergebnisse besser mit unseren selbst simulierten Daten zu vergleichen.

Die Ergebnisse zu der schwachen Streuung sind in Abbildung 17 zu sehen. Die optimale Größenordnung von  $\lambda$  liegt hier bei  $10^0$ . Der entsprechende Rekonstruktionsfehler ist 2,22 mm. Dieser ist deutlich kleiner, als in unseren selbst generierten Daten, da wir dort eine unrealistisch hohe Positronenweglänge simuliert haben.

Die Ergebnisse zu starker Streuung sind in Abbildung 7 im Anhang zu sehen. Die optimale Größenordnung von  $\lambda$  ist hier höher, als bei schwacher Streuung und liegt bei  $10^1$ . Dies liegt daran, dass bei schlechteren Daten mehr Regularisierung notwendig ist. Der entsprechende Rekonstruktionsfehler liegt bei 18,52 mm und ist wie zu erwarten deutlich höher, als bei schwacher Streuung.

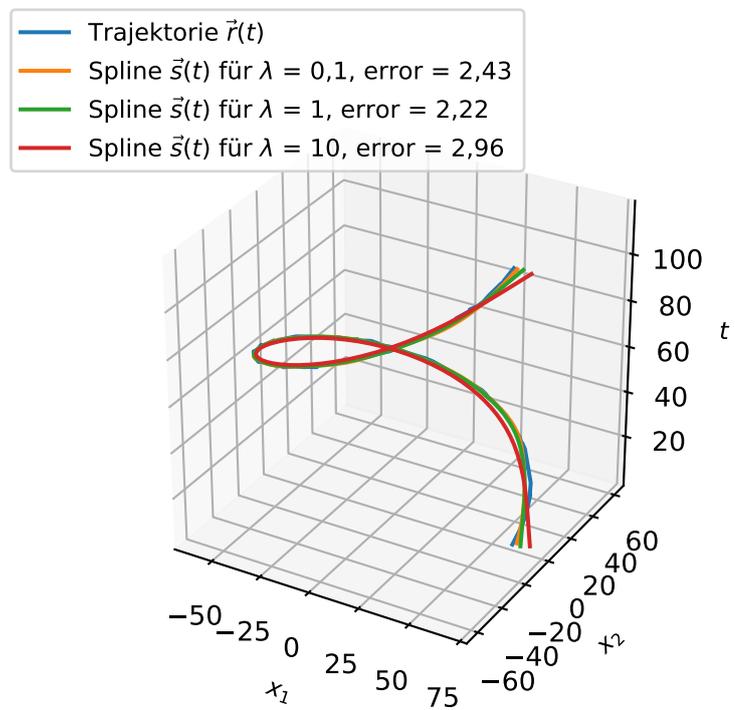
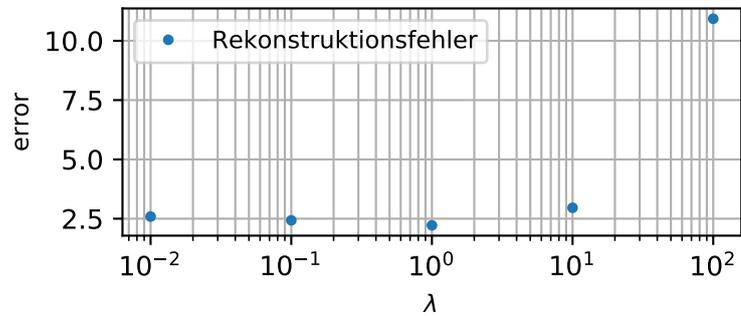


Abbildung 7: Rekonstruktion der Trajektorie aus externen PET-Daten  $\vec{r}(t)$  mit Minimierung der Beschleunigung bei  $N = 20$  äquidistanten Spline-Stützstellen. Oben: Rekonstruktionsfehler in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^0, \dots, 10^4$ . Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 10$  (orange), 100 (grün) und 1000 (rot) im  $x_1$ - $x_2$ - $t$ -Raum.

## 6 Schlussfolgerung

In dieser Arbeit haben wir uns mit den Grundlagen der Positronen-Emissions-Tomographie auseinandergesetzt und bekannte Verfahren zur Verfolgung einzelner Zellen im lebendem Organismus erläutert. Die hauptsächliche Schwierigkeit solcher Verfahren liegt in der geringen Dosis eines radioaktiven Markers, mit dem eine einzelne Zelle markiert werden kann. Aus diesem Grund werden in [5] anstelle von Bildern Trajektorien  $\vec{r}(t)$  rekonstruiert.

Aufbauend auf [5] haben wir die Trajektorie einer Zelle mit B-Splines modelliert. Diese erfüllen beim Grad von 1 die Minimallängen-Eigenschaft (Definition 9) und beim Grad 3 die Minimalkrümmungs-Eigenschaft (Definition 10), welche im physikalischen Kontext wahrscheinliche Zellbewegungen beschreiben.

Anschließend wurde der Datenterm  $I(\vec{a})$  und Regularisierungsterme  $R_1(\vec{a})$  und  $R_2(\vec{a})$  zur Minimierung von Länge und Krümmung hergeleitet und auf Existenz und Eindeutigkeit globaler Minimierer untersucht.

Um die hergeleiteten Zielfunktionen zu testen wurden einfache PET-Daten selbst generiert und externe simulierte Daten aus [8] verwendet.

Den geringsten Rekonstruktionsfehler bei den selbst generierten Daten wies dabei die Regularisierung durch Minimierung der Beschleunigung auf. Dies war nur dann der Fall, wenn die Spline-Stützstellen äquidistant gewählt wurden. Damit sollte die Minimierung der Beschleunigung und Modellierung der Trajektorie durch kubische Splines etwas besser sein, als das in [5] vorgestellte Verfahren, bei dem die Splines wahrscheinlich nur quadratisch waren und die Geschwindigkeit minimiert wurde. Ob dies auch für kompliziertere Trajektorien, als der hier verwendeten gilt, müsste in weiteren Experimenten untersucht werden.

Zu der Rekonstruktion der externen Daten, kann festgehalten werden, dass der optimale Relaxationsparameter abhängig von der Stärke der Streuung gewählt werden muss. Je stärker die Streuung, desto stärker muss die Regularisierung sein.

Die hier verwendeten Verfahren haben gegenüber Bildrekonstruktionsverfahren den Vorteil, dass sie auch mit geringer Radioaktivität umgehen können. Jedoch kann immer nur eine Zelle verfolgt werden, da das Verfahren nicht differenzieren kann, welche Zelle zu welcher Messung gehört. Möchte Man mehrere Zellen verfolgen, so kann man diese anstatt durch eine Trajektorie  $\vec{r}(t)$  durch eine Radionukleiddichte  $\rho = (\rho_t(x))$  modellieren und ebenfalls eine Regularisierung einführen, wie es in [8] getan wird.

## 7 Literaturverzeichnis

- [1] GATE a Monte-Carlo simulation toolkit for medical physics applications, 9 2020. <https://opengate.readthedocs.io/en/latest/introduction.html>.
- [2] Matlab Dokumentation zu bspline, 9 2020. <https://www.mathworks.com/help/curvefit/bspline.html#d122e24637>.
- [3] Dale L Bailey, Michael N Maisey, David W Townsend, and Peter E Valk. *Positron emission tomography*, volume 2. Springer, 2005.
- [4] C.R. Bemrose, P. Fowles, M.R. Hawkesworth, and M.A. O'Dwyer. Application of positron emission tomography to particulate flow measurement in chemical engineering processes. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 273(2):874 – 880, 1988.
- [5] K. S. Lee, T. J. Kim, and G. Pratz. Single-cell tracking with pet using a novel trajectory reconstruction algorithm. *IEEE Transactions on Medical Imaging*, 34(4):994–1003, 2015.
- [6] H.J. Oberle. Skript zur Vorlesung Variationsrechnung u. Optimale Steuerung, 9 2020. <https://www.math.uni-hamburg.de/home/oberle/skripte/varopt/varopt.pdf>.
- [7] Hans Rudolf, Schwarz, and Norbert Köckler. *Numerische Mathematik*. Teubner, 2006.
- [8] B. Schmitzer, K. P. Schäfers, and B. Wirth. Dynamic cell imaging in pet with optimal transport regularization. *IEEE Transactions on Medical Imaging*, 39(5):1626–1635, 2020.
- [9] Benedikt Wirth. Skript zur Vorlesung Numerische Analysis, 9 2020. <https://www.uni-muenster.de/AMM/Veranstaltungen/SS20/NumerischeAnalysis/notes.pdf>.

## 8 Abbildungsverzeichnis

1	Messung einer Positronenemission . . . . .	3
2	Lineare, quadratische und kubische B-Splines . . . . .	9
3	Kubische B-Splines mit unterschiedlich gewählten Knotenpunkten . . . . .	10
4	Approximation mit kubischen Splines . . . . .	13
5	Gleichverteilte normierte Richtungsvektoren als Punkte auf der Einheitskugel	24
6	Selbstgenerierte PET-Daten . . . . .	25
7	Rekonstruktion der Trajektorie $\vec{r}(t)$ aus externen PET-Daten mit schwacher Streuung durch Minimierung der Beschleunigung bei $N = 20$ äquidistanten Spline-Stützstellen . . . . .	28
8	Matrix $\tilde{M}$ . . . . .	32
9	Matrix $\tilde{L}_1$ . . . . .	32
10	Matrix $\tilde{L}_2$ . . . . .	33
11	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung des Datenterms $I(\vec{a})$ bei äquidistanten Spline-Stützstellen . . . . .	34
12	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung der Geschwindigkeit bei $N = 20$ äquidistanten Spline-Stützstellen . . . . .	35
13	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung der Geschwindigkeit bei identischen Daten- und Spline-Stützstellen . . . . .	36
14	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung der Beschleunigung bei $N = 20$ äquidistanten Spline-Stützstellen . . . . .	37
15	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung der Beschleunigung bei identischen Daten- und Spline-Stützstellen . . . . .	38
16	Rekonstruktion der Trajektorie $\vec{r}(t)$ mit Minimierung der Beschleunigung bei $N = 100$ äquidistanten Spline-Stützstellen . . . . .	39
17	Rekonstruktion der Trajektorie $\vec{r}(t)$ aus externen PET-Daten mit starker Streuung durch Minimierung der Beschleunigung bei $N = 20$ äquidistanten Spline-Stützstellen . . . . .	40

Alle in dieser Arbeit enthaltenen Abbildungen wurden im beigefügten Python-Code mit den Python-Paketen `matplotlib` und `mpl_toolkits` oder LaTeX erstellt und sind keiner anderen Quelle entnommen.

## 9 Anhang

Im nachfolgenden Anhang finden sich ausgewertete Beispielmatrizen, weitere Abbildungen und die Plagiatserklärung. Der in dieser Arbeit verwendete Python-Code ist auf der beigefügten CD als Jupyter-Notebook hinterlegt.

### 9.1 Matrizen

Im Folgenden wurden einige der in dieser Arbeit definierten Matrizen explizit mit dem beigefügten Python-Code berechnet, um ihre Struktur zu veranschaulichen.

$$\tilde{M} = \begin{pmatrix} 1,0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0,332 & 0,511 & 0,147 & 0,01 & 0 & 0 & 0 & 0 & 0 \\ 0,057 & 0,445 & 0,421 & 0,078 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,169 & 0,568 & 0,262 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,017 & 0,386 & 0,58 & 0,016 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,098 & 0,694 & 0,208 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,004 & 0,354 & 0,587 & 0,055 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,055 & 0,587 & 0,354 & 0,004 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,208 & 0,694 & 0,098 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,016 & 0,58 & 0,386 & 0,017 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,262 & 0,568 & 0,169 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,078 & 0,421 & 0,445 & 0,057 \\ 0 & 0 & 0 & 0 & 0 & 0,01 & 0,147 & 0,511 & 0,332 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1,0 \end{pmatrix}$$

Abbildung 8: Matrix  $\tilde{M} \in \mathbb{R}^{K \times N}$  für  $K = 14$  äquidistante Daten- und  $N = 9$  äquidistante Spline-Stützstellen. Es wurden kubische B-Splines mit den Knoten aus Definition 14 verwendet.

$$\tilde{L}_1 = \begin{pmatrix} 0,72 & -0,453 & -0,227 & -0,04 & 0 & 0 & 0 & 0 & 0 \\ -0,453 & 0,56 & 0,037 & -0,139 & -0,004 & 0 & 0 & 0 & 0 \\ -0,227 & 0,037 & 0,347 & -0,029 & -0,122 & -0,005 & 0 & 0 & 0 \\ -0,04 & -0,139 & -0,029 & 0,517 & -0,14 & -0,163 & -0,005 & 0 & 0 \\ 0 & -0,004 & -0,122 & -0,14 & 0,533 & -0,14 & -0,122 & -0,004 & 0 \\ 0 & 0 & -0,005 & -0,163 & -0,14 & 0,517 & -0,029 & -0,139 & -0,04 \\ 0 & 0 & 0 & -0,005 & -0,122 & -0,029 & 0,347 & 0,037 & -0,227 \\ 0 & 0 & 0 & 0 & -0,004 & -0,139 & 0,037 & 0,56 & -0,453 \\ 0 & 0 & 0 & 0 & 0 & -0,04 & -0,227 & -0,453 & 0,72 \end{pmatrix}$$

Abbildung 9: Matrix  $\tilde{L}_1 \in \mathbb{R}^{N \times N}$  für  $N = 9$  äquidistante Spline-Stützstellen. Es wurden kubische B-Splines mit den Knoten aus Definition 14 verwendet.

$$\tilde{L}_2 = \begin{pmatrix} 0,768 & -1,109 & 0,213 & 0,128 & 0 & 0 & 0 & 0 & 0 \\ -1,109 & 1,792 & -0,597 & -0,142 & 0,057 & 0 & 0 & 0 & 0 \\ 0,213 & -0,597 & 0,683 & -0,391 & 0,028 & 0,064 & 0 & 0 & 0 \\ 0,128 & -0,142 & -0,391 & 1,067 & -0,768 & 0,043 & 0,064 & 0 & 0 \\ 0 & 0,057 & 0,028 & -0,768 & 1,365 & -0,768 & 0,028 & 0,057 & 0 \\ 0 & 0 & 0,064 & 0,043 & -0,768 & 1,067 & -0,391 & -0,142 & 0,128 \\ 0 & 0 & 0 & 0,064 & 0,028 & -0,391 & 0,683 & -0,597 & 0,213 \\ 0 & 0 & 0 & 0 & 0,057 & -0,142 & -0,597 & 1,792 & -1,109 \\ 0 & 0 & 0 & 0 & 0 & 0,128 & 0,213 & -1,109 & 0,768 \end{pmatrix}$$

Abbildung 10: Matrix  $\tilde{L}_2 \in \mathbb{R}^{K \times N}$  für  $N = 9$  äquidistante Spline-Stützstellen. Es wurden kubische B-Splines mit den Knoten aus Definition 14 verwendet.

## 9.2 Weitere Abbildungen zur Rekonstruktion der Zellbewegung

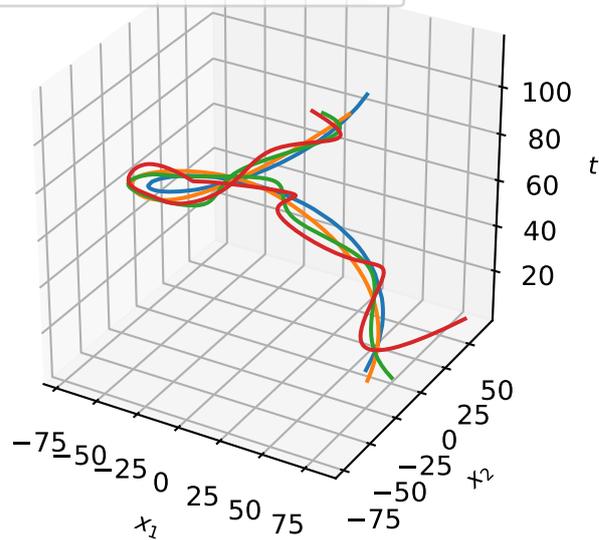
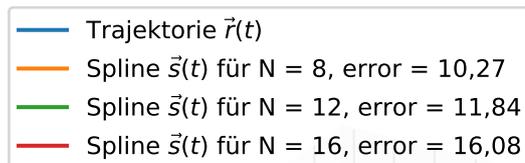
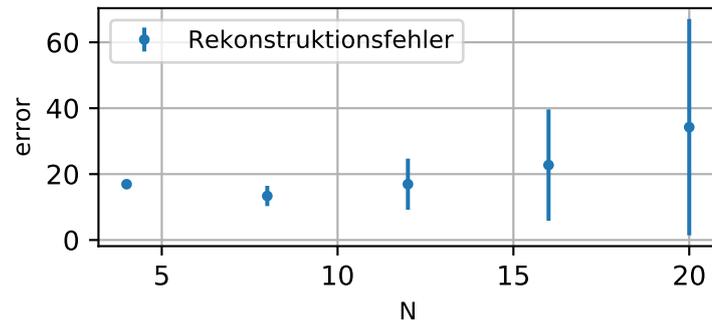


Abbildung 11: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung des Datenterms  $I(\vec{a})$  bei äquidistanten Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit von der Zahl  $N$  der Spline-Stützstellen. Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $N = 8$  (orange), 12 (grün) und 16 (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.

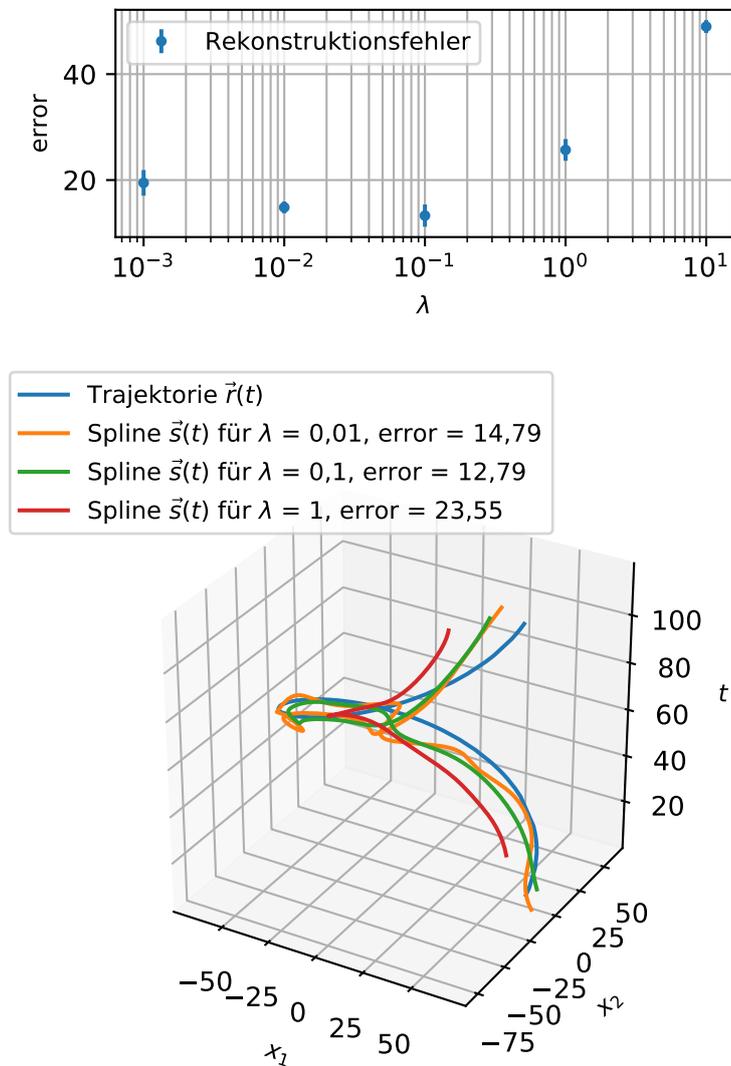


Abbildung 12: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung der Geschwindigkeit bei  $N = 20$  äquidistanten Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^{-3}, \dots, 10^1$ . Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 0,01$  (orange),  $0,1$  (grün) und  $1$  (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.

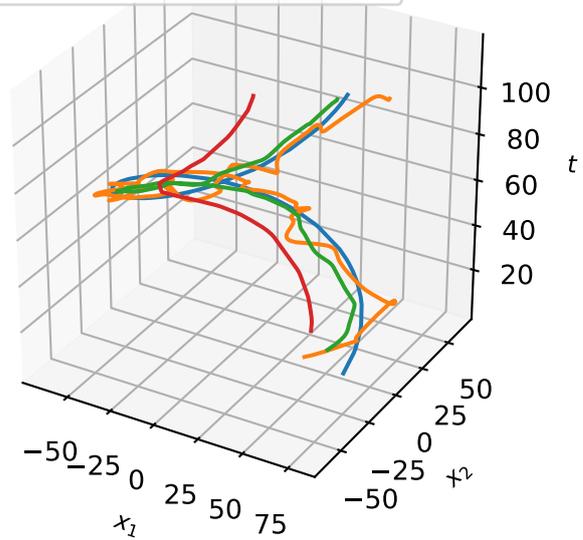
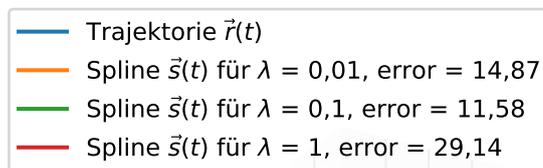
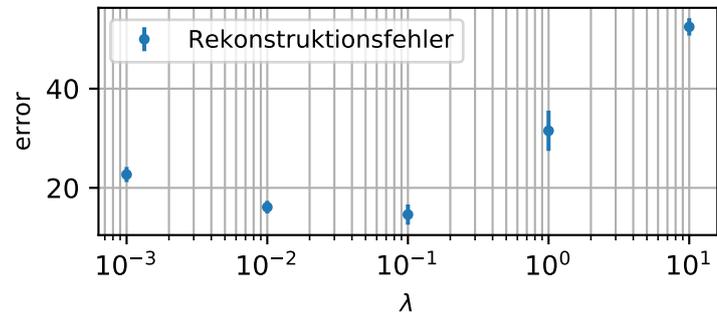


Abbildung 13: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung der Geschwindigkeit bei identischen Daten- und Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^{-3}, \dots, 10^1$ . Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 0,01$  (orange),  $0,1$  (grün) und  $1$  (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.

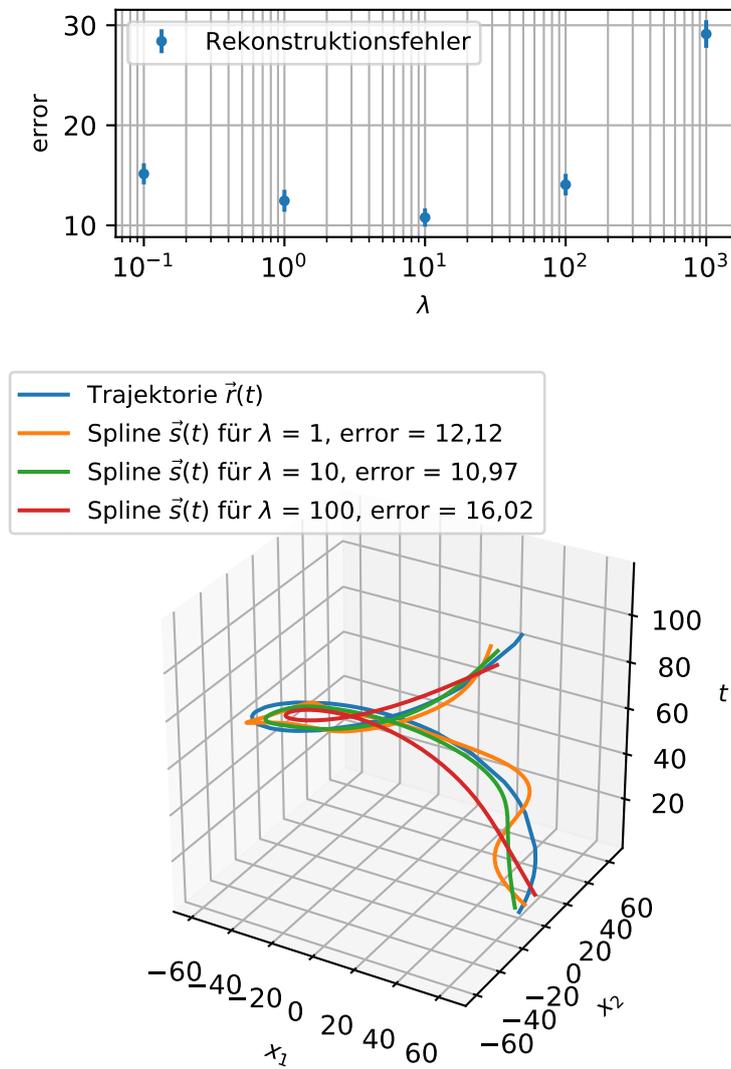


Abbildung 14: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung der Beschleunigung bei  $N = 20$  äquidistanten Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^{-1}, \dots, 10^3$ . Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 1$  (orange), 10 (grün) und 100 (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.

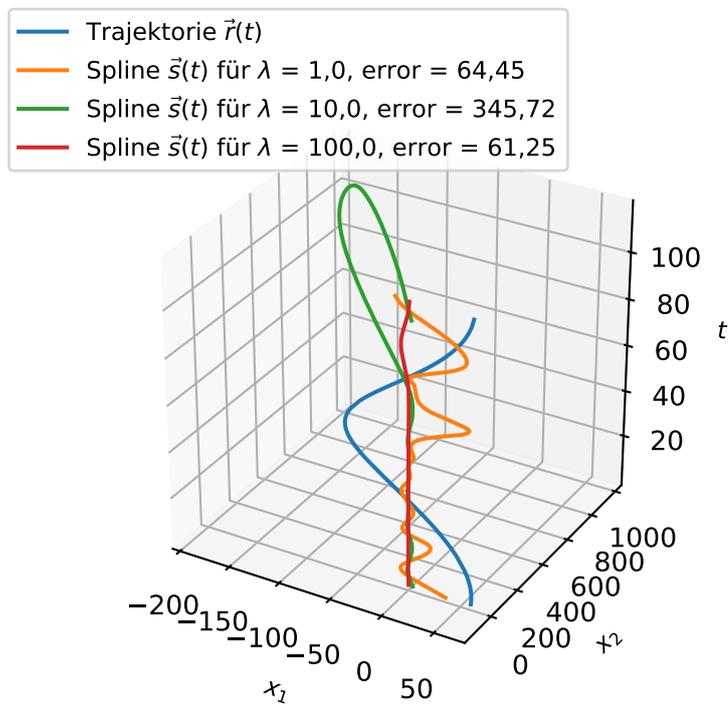
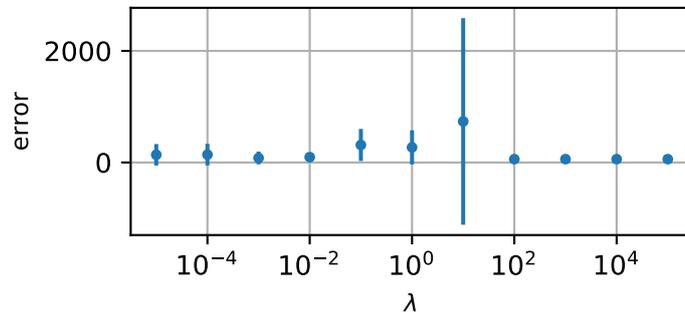


Abbildung 15: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung der Beschleunigung bei identischen Daten- und Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^{-5}, \dots, 10^5$ . Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 1$  (orange), 10 (grün) und 100 (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.

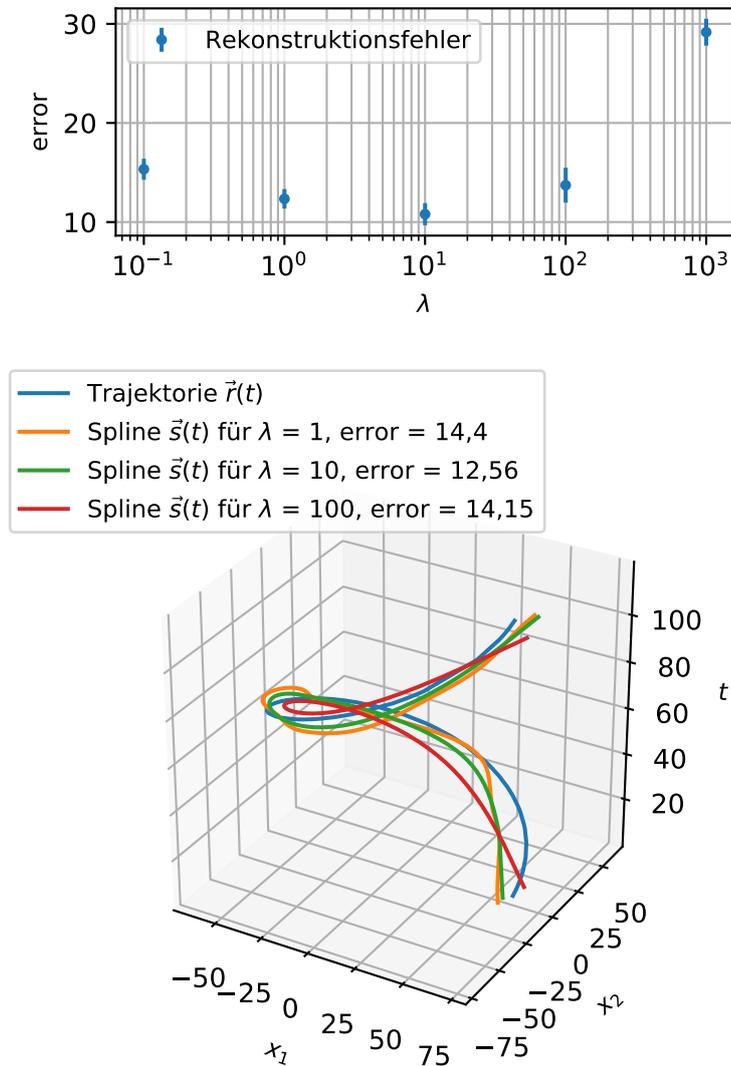
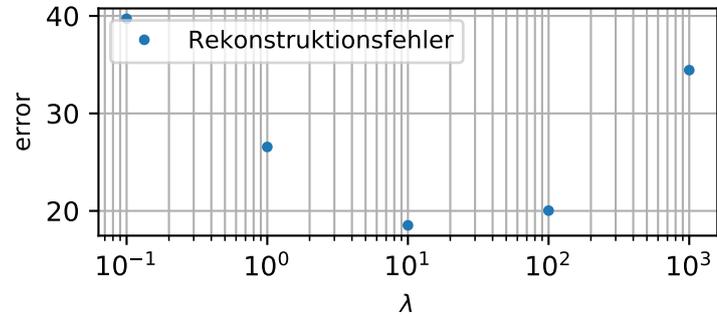


Abbildung 16: Rekonstruktion der Trajektorie  $\vec{r}(t)$  mit Minimierung der Beschleunigung bei  $N = 100$  äquidistanten Spline-Stützstellen. Oben: Mittelwert des Rekonstruktionsfehlers in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^{-1}, \dots, 10^3$ . Es wurde jeweils für 10 verschiedene PET-Datensätze ein Spline  $\vec{s}(t)$  rekonstruiert. Die Balken geben die Standardabweichung des Rekonstruktionsfehlers an. Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 1$  (orange), 10 (grün) und 100 (rot) zum selben PET-Datensatz im  $x_1$ - $x_2$ - $t$ -Raum.



- Trajektorie  $\vec{r}(t)$
- Spline  $\vec{s}(t)$  für  $\lambda = 1$ , error = 26,56
- Spline  $\vec{s}(t)$  für  $\lambda = 10$ , error = 18,52
- Spline  $\vec{s}(t)$  für  $\lambda = 100$ , error = 20,03

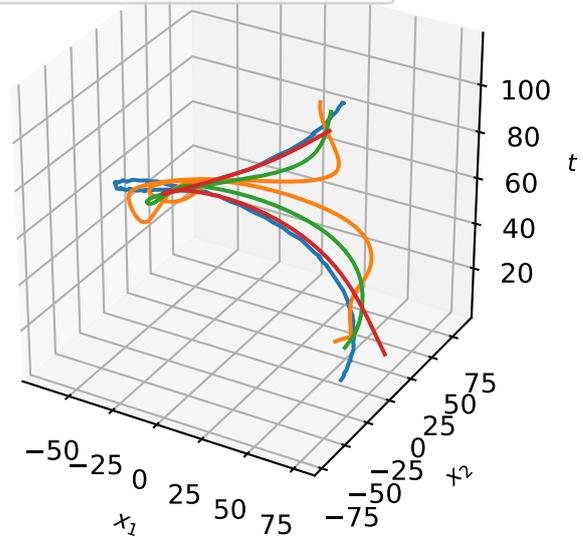


Abbildung 17: Rekonstruktion der Trajektorie aus externen PET-Daten  $\vec{r}(t)$  mit Minimierung der Beschleunigung bei  $N = 20$  äquidistanten Spline-Stützstellen. Oben: Rekonstruktionsfehler in Abhängigkeit vom Relaxationsparameter  $\lambda = 10^0, \dots, 10^4$ . Unten: Die reale Trajektorie  $\vec{r}(t)$  der Zelle (blau) und drei rekonstruierte Splines  $\vec{s}(t)$  für  $\lambda = 10$  (orange), 100 (grün) und 1000 (rot) im  $x_1$ - $x_2$ - $t$ -Raum.

## Plagiatserklärung der / des Studierenden

Hiermit versichere ich, dass die vorliegende Arbeit über „**Verfolgung von Einzelzellen mit PET und kubischer Spline-Interpolation**“ 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)