

Projektarbeit

**Evolution von Kurven  
unter dem Curve Shortening Flow  
und dem Elastic Flow**

Prof. Dr. Heiko von der Mosel

Christian Bagh (251189), Thomas Bedbur (251451),  
Michael Dahmen (251575), Tobias Hermes (252158),  
Jan Jongen (252365), Matthias Schlottbom (251942)

RWTH Aachen

November 2006

# Inhaltsverzeichnis

	<b>Vorwort</b>	<b>3</b>
<b>1</b>	<b>Curve Shortening Flow</b>	<b>4</b>
1.1	Einleitung . . . . .	4
1.2	Numerische Berechnung . . . . .	4
1.3	Beispiele . . . . .	9
1.3.1	Deltoid . . . . .	9
1.3.2	Spirale . . . . .	10
1.3.3	Limaçon . . . . .	10
<b>2</b>	<b>Elastic Flow</b>	<b>12</b>
2.1	Einleitung . . . . .	12
2.2	Numerische Berechnung . . . . .	12
2.3	Beispiele . . . . .	23
2.3.1	Sattel . . . . .	23
2.3.2	Lemniskate . . . . .	24
2.3.3	Verschlingungskurve . . . . .	25
<b>3</b>	<b>Implementierung und Methoden</b>	<b>26</b>
3.1	Implementierung . . . . .	26
3.2	Neuverteilung der Knoten . . . . .	29
3.3	Mass Lumping . . . . .	30
	<b>Literaturverzeichnis</b>	<b>31</b>

## Vorwort

In unserem Projekt im Studiengang Computermathematik an der RWTH Aachen haben wir den Curve Shortening Flow und den Elastic Flow untersucht, algorithmisiert und implementiert. Unsere Arbeit basiert dabei größtenteils auf den Arbeiten [1] „Computation of geometric partial differential equations and mean curvature flow“ und [2] „Evolution of elastic curves in  $\mathbb{R}^n$ : Existence and computation“.

Die Untersuchung der beiden obigen Flüsse fällt ins mathematische Gebiet der geometrischen Analysis und hat für praktische Anwendungen weitreichende Bedeutung. Typische Anwendungsbeispiele des Curve Shortening Flow finden sich in den Materialwissenschaften und in der Bildverarbeitung.

Der Elastic Flow beschreibt, in welcher Art und Weise sich die Oberflächenspannung, beschrieben durch die Krümmung, in der Zeit verändert. Dabei beruht der elastische Fluss auf der Minimierung eines Energiefunktional, so dass die elastische Energie von Kurven unter dem Elastic Flow minimiert wird.

In Kapitel 1 wird der Curve Shortening Flow behandelt. Dort finden sich die Herleitung des Algorithmus und Beispiele. In Kapitel 2 wird der elastische Fluss untersucht; auch hier werden die Algorithmen ausführlich hergeleitet und die Ergebnisse an Beispielen visualisiert. Kapitel 3 beschäftigt sich dann mit der Implementierung der Algorithmen und einer näheren Erläuterung von ausgewählten Methoden, die in den Algorithmen eine besondere Rolle spielen.

Dieser Projektarbeit liegt eine CD bei, auf der man den Quellcode der konkreten Implementierung der Algorithmen und Visualisierung der Flüsse in Maple und C++ findet. Die Programme zum Curve Shortening Flow und zum Elastic Flow haben wir im Maple-Paket `cf_low` zusammengefasst. Dort wird auf Hilfe-Seiten die Funktionalität der einzelnen Befehle erläutert. Zusätzlich befinden sich auf der CD auch zwei Maple-Worksheets mit Beispielen, in denen die einzelnen Flüsse auf verschiedene Kurven angewendet werden.

Wir möchten dem Institut für Mathematik der RWTH Aachen für die freundliche Unterstützung danken. Ein besonderer Dank gilt Herrn Prof. von der Mosel für das interessante Thema und die gute Betreuung.

Aachen, im November 2006

Christian Bagh, Thomas Bedbur, Michael Dahmen,  
Tobias Hermes, Jan Jongen, Matthias Schlottbom

# 1 Curve Shortening Flow

## 1.1 Einleitung

Der Curve Shortening Flow beschreibt die Entwicklung einer geschlossenen Kurve in der Zeit, wobei die Entwicklung der Kurve in jedem Punkt durch die zugehörige Normale und Krümmung beschrieben wird. Die Kurve zum Zeitpunkt  $t$  sei gegeben durch eine Abbildung  $X(\cdot, t) : S^1 \rightarrow \mathbb{R}^2$  mit  $0 \leq t < T$ , wobei  $X$  die folgende partielle Differentialgleichung löst:

$$\begin{cases} X_t - \frac{1}{|X_\theta|} \left( \frac{X_\theta}{|X_\theta|} \right)_\theta = 0 & \text{in } I \times (0, T) \\ X(\cdot, 0) = X_0 & \text{in } I \end{cases} \quad (1)$$

Hierbei ist  $I = [0, 2\pi]$  und  $X(\theta, t) = X(\theta + 2\pi, t)$  für  $0 \leq t < T$ ,  $\theta \in \mathbb{R}$ . Insbesondere gilt dann  $|X_\theta| > 0$  in  $I \times [0, T]$ .

## 1.2 Numerische Berechnung

Unser Ziel ist es nun, die Lösung der partiellen Differentialgleichung (1) zu diskretisieren und damit einen Algorithmus implementieren zu können. Hierzu betrachtet man zunächst die schwache Formulierung der Differentialgleichung (1):

Angenommen  $X$  ist eine Lösung von (1). Multipliziert man die erste Gleichung mit  $|X_\theta|$ , bildet das Skalarprodukt mit einer Testfunktion  $\varphi \in H_{per}^1(I; \mathbb{R}^2)$  und integriert über  $I$ , so erhält man mit partieller Integration die schwache Formulierung der Differentialgleichung

$$\begin{aligned} & \int_I X_t \cdot \varphi |X_\theta| \, d\theta - \int_I \left( \frac{X_\theta}{|X_\theta|} \right)_\theta \cdot \varphi \, d\theta \\ &= \int_I X_t \cdot \varphi |X_\theta| \, d\theta + \int_I \frac{X_\theta \cdot \varphi_\theta}{|X_\theta|} \, d\theta = 0 \quad \forall \varphi \in H_{per}^1(I; \mathbb{R}^2) \end{aligned} \quad (2)$$

Möchte man nun die räumliche Diskretisierung durchführen, führt man zur Vereinfachung ein konstantes Rasterfeld  $\theta_j = jh$  mit Länge  $h = \frac{2\pi}{N}$  ein ( $j = 0, \dots, N$ ). Außerdem sei

$$S_h = \{ \varphi_h \in C^0(I; \mathbb{R}^2) \mid \varphi_h|_{[\theta_{j-1}, \theta_j]} \in P_1^2, j = 1, \dots, N; \varphi_h(0) = \varphi_h(2\pi) \}.$$

Damit ergibt sich die schwache räumliche Diskretisierung

$$\int_I X_{ht} \cdot \varphi_h |X_{h\theta}| \, d\theta + \int_I \frac{X_{h\theta} \cdot \varphi_{h\theta}}{|X_{h\theta}|} \, d\theta = 0 \quad \forall \varphi_h \in S_h \quad (3)$$

Da  $S_h$  endlich dimensional ist, wähle man die Basis  $\{\phi_1, \dots, \phi_N\}$  mit

$$\phi_j := \frac{\theta - \theta_{j-1}}{\theta_j - \theta_{j-1}} \chi_{[\theta_{j-1}, \theta_j]} + \frac{\theta_{j+1} - \theta}{\theta_{j+1} - \theta_j} \chi_{[\theta_j, \theta_{j+1}]}$$

## 1.1 GRUNDLAGEN

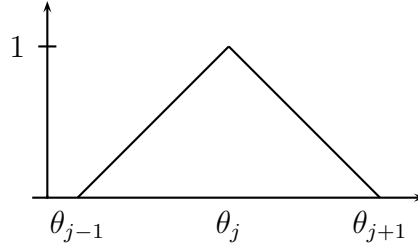


Abbildung 1: Darstellung des Basiselementes  $\phi_j$

Offensichtlich gilt

$$\phi_i \phi_j = 0, \text{ falls } i \notin \{j-1, j, j+1\} \quad (4)$$

und man erhält

$$X_h(\theta, t) = \sum_{j=1}^N X_j(t) \phi_j(\theta) \quad X_j \in \mathbb{R}^2$$

Für die Berechnung des Differenzschemas benötigt man folgende Ableitungen

$$X_{ht}(\theta, t) = \sum_{j=1}^N \dot{X}_j(t) \phi_j(\theta)$$

$$X_{h\theta}(\theta, t) = \sum_{j=1}^N X_j(t) \phi'_j(\theta)$$

$$(\phi_j)_\theta = \frac{1}{\theta_j - \theta_{j-1}} \chi_{[\theta_{j-1}, \theta_j)} - \frac{1}{\theta_{j+1} - \theta_j} \chi_{[\theta_j, \theta_{j+1})}$$

Wählt man nun  $\varphi_h = \phi_l e^k$ ,  $k = 1, 2$ ,  $l = 1, \dots, N$ , wobei  $e^k$  der  $k$ -te Einheitsvektor im  $\mathbb{R}^2$  ist, so ergibt sich durch Einsetzen in (3):

$$\underbrace{\int_I X_{ht} \cdot (\phi_l e^k) |X_{h\theta}| d\theta}_{(I)} + \underbrace{\int_I \frac{X_{h\theta} \cdot (\phi_l e^k)_\theta}{|X_{h\theta}|} d\theta}_{(II)} = 0$$

Zu (I):

$$\int_I X_{ht} \cdot (\phi_l e^k) |X_{h\theta}| d\theta = \int_I \left( \sum_{j=1}^N \dot{X}_j \phi_j \right) \cdot (\phi_l e^k) |X_{h\theta}| d\theta = \int_I \sum_{j=1}^N \dot{X}_j \cdot e^k \phi_j \phi_l |X_{h\theta}| d\theta$$

$$\stackrel{(4)}{=} \int_I (\dot{X}_{l-1} \cdot e^k \phi_{l-1} \phi_l + \dot{X}_l \cdot e^k \phi_l^2 + \dot{X}_{l+1} \cdot e^k \phi_{l+1} \phi_l) |X_{h\theta}| d\theta$$

## 1.1 GRUNDLAGEN

$$= \dot{X}_{l-1} \cdot e^k \underbrace{\int_I \phi_{l-1} \phi_l |X_{h\theta}| d\theta}_{(A)} + \dot{X}_l \cdot e^k \underbrace{\int_I \phi_l^2 |X_{h\theta}| d\theta}_{(C)} + \dot{X}_{l+1} \cdot e^k \underbrace{\int_I \phi_{l+1} \phi_l |X_{h\theta}| d\theta}_{(B)}$$

Man erhält für die einzelnen Integrale:

(A): Mit  $X_{h\theta}|_{[\theta_{l-1}, \theta_l]} = (X_{l-1} \phi'_{l-1} + X_l \phi'_l)|_{[\theta_{l-1}, \theta_l]} = \frac{X_l - X_{l-1}}{\theta_l - \theta_{l-1}}$  folgt

$$\begin{aligned} \int_I \phi_{l-1} \phi_l |X_{h\theta}| d\theta &= \int_{\theta_{l-1}}^{\theta_l} \phi_{l-1} \phi_l |X_{h\theta}| d\theta = \int_{\theta_{l-1}}^{\theta_l} \frac{(\theta_l - \theta)(\theta - \theta_{l-1})}{(\theta_l - \theta_{l-1})^2} \left| \frac{X_l - X_{l-1}}{\theta_l - \theta_{l-1}} \right| d\theta \\ &= \frac{|X_l - X_{l-1}|}{|\theta_l - \theta_{l-1}|^3} \left[ \frac{1}{2} \theta_l \theta^2 - \frac{1}{3} \theta^3 + \frac{1}{2} \theta_{l-1} \theta^2 - \theta_l \theta_{l-1} \theta \right]_{\theta_{l-1}}^{\theta_l} \\ &= \frac{|X_l - X_{l-1}|}{|\theta_l - \theta_{l-1}|^3} \left( \frac{1}{6} \theta_l^3 - \frac{1}{2} \theta_l^2 \theta_{l-1} + \frac{1}{2} \theta_l \theta_{l-1}^2 - \frac{1}{6} \theta_{l-1}^3 \right) = \frac{1}{6} |X_l - X_{l-1}| \end{aligned}$$

(B): Die Berechnung dieses Integrals verläuft analog.

Hierbei gilt:  $X_{h\theta}|_{[\theta_l, \theta_{l+1}]} = (X_l \phi'_l + X_{l+1} \phi'_{l+1})|_{[\theta_l, \theta_{l+1}]} = \frac{X_{l+1} - X_l}{\theta_{l+1} - \theta_l}$ .

Damit ergibt sich

$$\begin{aligned} \int_I \phi_{l+1} \phi_l |X_{h\theta}| d\theta &= \int_{\theta_l}^{\theta_{l+1}} \frac{(\theta - \theta_l)(\theta_{l+1} - \theta)}{(\theta_{l+1} - \theta_l)^2} \left| \frac{X_{l+1} - X_l}{\theta_{l+1} - \theta_l} \right| d\theta \\ &= \frac{|X_{l+1} - X_l|}{|\theta_{l+1} - \theta_l|^3} \left[ \frac{1}{2} \theta_{l+1} \theta^2 - \frac{1}{3} \theta^3 + \frac{1}{2} \theta_l \theta^2 - \theta_l \theta_{l+1} \theta \right]_{\theta_l}^{\theta_{l+1}} \\ &= \frac{|X_{l+1} - X_l|}{|\theta_{l+1} - \theta_l|^3} \left( \frac{1}{6} \theta_{l+1}^3 - \frac{1}{2} \theta_{l+1}^2 \theta_l + \frac{1}{2} \theta_{l+1} \theta_l^2 - \frac{1}{6} \theta_l^3 \right) = \frac{1}{6} |X_{l+1} - X_l| \end{aligned}$$

(C):

$$\begin{aligned} \int_I \phi_l^2 |X_{h\theta}| d\theta &= \int_{\theta_{l-1}}^{\theta_{l+1}} \phi_l^2 |X_{h\theta}| d\theta \\ &= \int_{\theta_{l-1}}^{\theta_l} \left( \frac{\theta - \theta_{l-1}}{\theta_l - \theta_{l-1}} \right)^2 \left| \frac{X_l - X_{l-1}}{\theta_l - \theta_{l-1}} \right| d\theta + \int_{\theta_l}^{\theta_{l+1}} \left( \frac{\theta_{l+1} - \theta}{\theta_{l+1} - \theta_l} \right)^2 \left| \frac{X_{l+1} - X_l}{\theta_{l+1} - \theta_l} \right| d\theta \end{aligned}$$

## 1.1 GRUNDLAGEN

$$\begin{aligned}
&= \frac{|X_l - X_{l-1}|}{|\theta_l - \theta_{l-1}|^3} \left[ \frac{1}{3}\theta^3 - \theta_{l-1}\theta^2 + \theta_{l-1}^2\theta \right]_{\theta_{l-1}}^{\theta_l} + \frac{|X_{l+1} - X_l|}{|\theta_{l+1} - \theta_l|^3} \left[ \frac{1}{3}\theta^3 - \theta_{l+1}\theta^2 + \theta_{l+1}^2\theta \right]_{\theta_l}^{\theta_{l+1}} \\
&= \frac{1}{3}(|X_l - X_{l-1}| + |X_{l+1} - X_l|)
\end{aligned}$$

Also ergibt sich insgesamt

$$\begin{aligned}
&\int_I X_{ht} \cdot \varphi_h |X_{h\theta}| d\theta = \frac{1}{6} |X_l - X_{l-1}| \dot{X}_{l-1} \cdot e^k \\
&+ \frac{1}{3} (|X_l - X_{l-1}| + |X_{l+1} - X_l|) \dot{X}_l \cdot e^k + \frac{1}{6} |X_{l+1} - X_l| \dot{X}_{l+1} \cdot e^k
\end{aligned}$$

Zu (II):

$$\begin{aligned}
&\int_I \frac{X_{h\theta} \cdot (\phi_l e^k)_\theta}{|X_{h\theta}|} d\theta = \int_I \frac{1}{|X_{h\theta}|} \left( \sum_{j=1}^N X_j \phi'_j \right) \cdot (\phi'_l e^k) d\theta = \int_I \frac{1}{|X_{h\theta}|} \sum_{j=1}^N X_j \cdot e^k \phi'_j \phi'_l d\theta \\
&\stackrel{(4)}{=} \int_I \frac{1}{|X_{h\theta}|} (X_{l-1} \cdot e^k \phi'_{l-1} \phi'_l + X_l \cdot e^k \phi_l'^2 + X_{l+1} \cdot e^k \phi'_{l+1} \phi_l) d\theta \\
&= X_{l-1} \cdot e^k \underbrace{\int_I \frac{\phi'_{l-1} \phi'_l}{|X_{h\theta}|} d\theta}_{(A)} + X_l \cdot e^k \underbrace{\int_I \frac{\phi_l'^2}{|X_{h\theta}|} d\theta}_{(C)} + X_{l+1} \cdot e^k \underbrace{\int_I \frac{\phi'_{l+1} \phi'_l}{|X_{h\theta}|} d\theta}_{(B)}
\end{aligned}$$

(A):

$$\int_I \frac{\phi'_{l-1} \phi'_l}{|X_{h\theta}|} d\theta = \int_{\theta_{l-1}}^{\theta_l} -\frac{1}{|X_{h\theta}|(\theta_l - \theta_{l-1})^2} d\theta = -\frac{1}{|X_l - X_{l-1}|}$$

(B):

$$\int_I \frac{\phi'_{l+1} \phi'_l}{|X_{h\theta}|} d\theta = \int_{\theta_l}^{\theta_{l+1}} -\frac{1}{|X_{h\theta}|(\theta_{l+1} - \theta_l)^2} d\theta = -\frac{1}{|X_{l+1} - X_l|}$$

(C):

$$\begin{aligned}
&\int_I \frac{\phi_l'^2}{|X_{h\theta}|} d\theta = \int_{\theta_{l-1}}^{\theta_l} \frac{\phi_l'^2}{|X_{h\theta}|} d\theta + \int_{\theta_l}^{\theta_{l+1}} \frac{\phi_l'^2}{|X_{h\theta}|} d\theta = \int_{\theta_{l-1}}^{\theta_l} \frac{1}{|X_{h\theta}|(\theta_l - \theta_{l-1})^2} d\theta + \int_{\theta_l}^{\theta_{l+1}} \frac{1}{|X_{h\theta}|(\theta_{l+1} - \theta_l)^2} d\theta \\
&= \frac{1}{|X_l - X_{l-1}|} + \frac{1}{|X_{l+1} - X_l|}
\end{aligned}$$

(II) hat also die Form

$$\int_I \frac{X_{h\theta} \cdot \varphi_{h\theta}}{|X_{h\theta}|} d\theta = \left( -\frac{X_{l+1} - X_l}{|X_{l+1} - X_l|} + \frac{X_l - X_{l-1}}{|X_l - X_{l-1}|} \right) \cdot e^k$$

Setzt man nun  $q_l := |X_l - X_{l-1}|$ , so ergibt sich für  $l = 1, \dots, N$  das Differenzenschema

$$\frac{1}{6}q_l \dot{X}_{l-1} + \frac{1}{3}(q_l + q_{l+1}) \dot{X}_l + \frac{1}{6}q_{l+1} \dot{X}_{l+1} = \frac{X_{l+1} - X_l}{q_{l+1}} - \frac{X_l - X_{l-1}}{q_l} \quad (5)$$

Mittels Mass Lumping<sup>1</sup> ersetzt man die zum linearen Gleichungssystem gehörende Matrix durch eine Diagonalmatrix, wobei man für die Diagonalelemente die Zeilensummen der ursprünglichen Matrix wählt. Damit kann man (5) darstellen als

$$\frac{1}{2}(q_l + q_{l+1}) \dot{X}_l = \frac{X_{l+1} - X_l}{q_{l+1}} - \frac{X_l - X_{l-1}}{q_l} \quad (6)$$

Wählt man nun als Zeitschritt  $\tau > 0$  und betrachtet die Zeitdiskretisierung  $(X_h^m)_t = \frac{X_h^{m+1} - X_h^m}{\tau}$ , wobei  $m = 0, \dots, [\frac{T}{\tau}]$ ,  $t_m = m\tau$  und  $X_h^m$  die Approximation an  $X(\cdot, t_m)$  bezeichnen, so erhält man

$$\int_I \frac{X_h^{m+1} - X_h^m}{\tau} \cdot \varphi |X_{h\theta}^m| d\theta + \int_I \frac{X_{h\theta}^{m+1} \cdot \varphi_\theta}{|X_{h\theta}^m|} d\theta = 0 \quad \forall \varphi \in H_{per}^1(I; \mathbb{R}^2)$$

Analoge Rechnungen wie beim Differenzenschema (6) führen zu folgendem Algorithmus

**Algorithmus (Curve Shortening Flow)**

(i)  $X_j^0 = X_0(\theta_j) \quad , j = 0, \dots, N$

(ii) Berechne  $X_j^{m+1} \quad , j = 0, \dots, N$  durch

$$\frac{1}{2\tau}(q_l^m + q_{l+1}^m)(X_l^{m+1} - X_l^m) - \left( \frac{X_{l+1}^{m+1} - X_l^{m+1}}{q_{l+1}^m} - \frac{X_l^{m+1} - X_{l-1}^{m+1}}{q_l^m} \right) = 0$$

(iii) Ist  $\min_{j=1, \dots, N+1} q_j^{m+1} > 0$ , ersetze  $m$  durch  $m + 1$  und gehe zu (ii)

Zur genaueren Berechnung in Schritt (2) und zur Implementierung des Algorithmus betrachte folgende Matrixschreibweise des linearen Gleichungssystems:

$$\begin{aligned} & \frac{q_l^m + q_{l+1}^m}{2\tau} X_l^{m+1} + \frac{X_l^{m+1}}{q_{l+1}^m} + \frac{X_l^{m+1}}{q_l^m} - \frac{X_{l+1}^{m+1}}{q_{l+1}^m} - \frac{X_{l-1}^{m+1}}{q_l^m} = \frac{q_l^m + q_{l+1}^m}{2\tau} X_l^m \\ \iff & \left( \frac{q_l^m + q_{l+1}^m}{2\tau} + \frac{1}{q_{l+1}^m} + \frac{1}{q_l^m} \right) X_l^{m+1} - \frac{X_{l+1}^{m+1}}{q_{l+1}^m} - \frac{X_{l-1}^{m+1}}{q_l^m} = \underbrace{\frac{q_l^m + q_{l+1}^m}{2\tau} X_l^m}_{\text{bekannt}} \end{aligned} \quad (7)$$

<sup>1</sup>Für nähere Informationen bzgl. Mass Lumping siehe Kapitel 3.3



## 1.3 Beispiele

Aus der Theorie des Curve Shortening Flows ist bekannt, dass einfach geschlossene Kurven zu einem konvexen Gebiet übergehen und letztendlich zu einem „Punkt“ konvergieren. Im Folgenden wird die Entwicklung einiger charakteristischer Kurven unter dem Curve Shortening Flow betrachtet. Es ist noch anzumerken, dass sich im Allgemeinen keine Aussage über die Entwicklung von nicht einfach geschlossenen Kurven machen lässt. Es gibt also Kurven, die sich zu einem Kreis entwickeln, aber auch solche, die dies nicht tun, wie zum Beispiel die lemniscate of Bernoulli (symmetrische Acht).

### 1.3.1 Deltoid

Nach obiger Bemerkung erwarten wir, dass sich der Deltoid als einfach geschlossene Kurve unter dem Einfluss des Curve Shortening Flows zu einem konvexen Gebiet entwickelt und dieses immer kleiner wird.

```
> g1:=t->[2*cos(t)+cos(2*t),2*sin(t)-sin(2*t)]:
> CSF(g1,80,1/10000,7000,10, Plot);
```

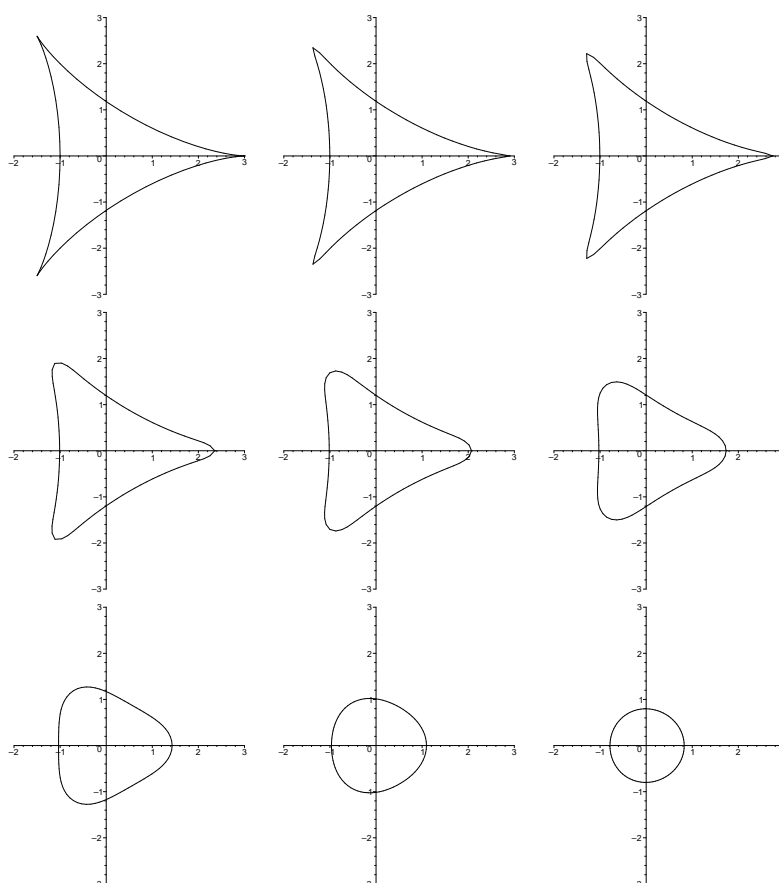


Abbildung 2: Deltoid zu den Zeitpunkten  $t = 0, 100, 240, 620, 1000, 1800, 3000, 5000, 7000$  (80 Knoten,  $\tau = \frac{1}{10000}$ )

## 1.3 BEISPIELE

### 1.3.2 Spirale

Das folgende Beispiel visualisiert die zeitliche Entwicklung einer Spirale unter Einfluss des Curve Shortening Flows. Die Spirale ist hierbei als Punktliste übergeben worden. Die genaue Konstruktion ist in dem beiliegenden Maple-Worksheet angegeben. Zu erwarten ist, dass sich die nicht konvexe Spirale in endlicher Zeit zu einem konvexen Gebiet entwickeln wird. Dies ist in der Tat der Fall, wie man an den folgenden Abbildungen erkennen kann:

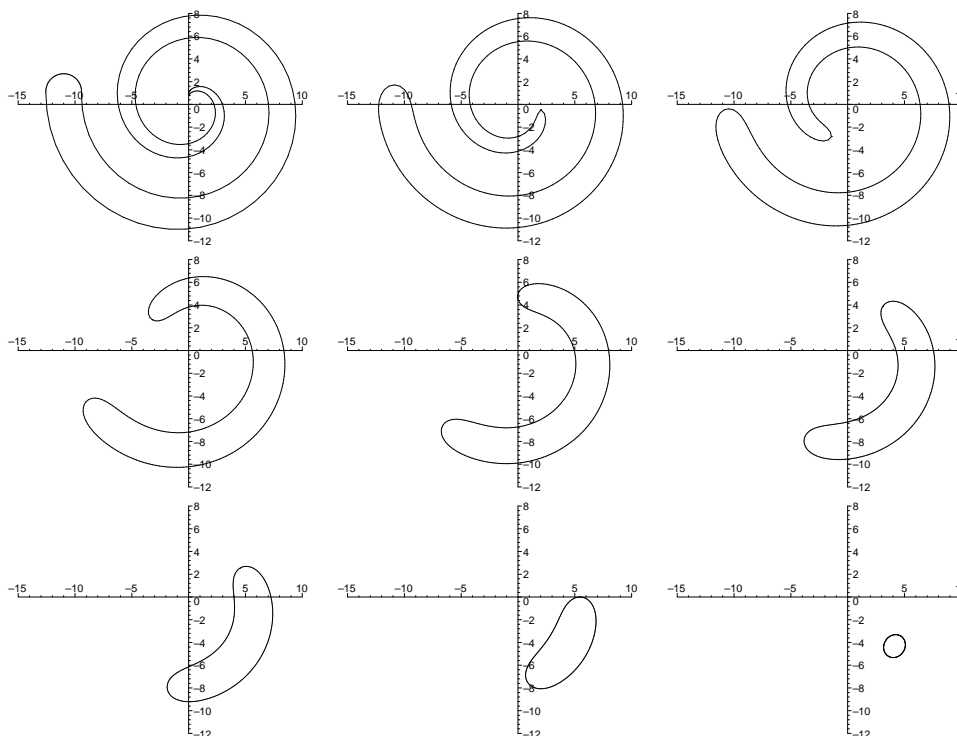


Abbildung 3: Spirale zu den Zeitpunkten  $t = 0, 50, 150, 350, 500, 650, 750, 900, 1100$  (250 Knoten,  $\tau = \frac{1}{50}$ )

### 1.3.3 Limaçon

Abschließend zur Darstellung der CSF-Funktion betrachten wir nun eine nicht einfach geschlossene Kurve. Interessant bei dieser Kurve ist die entstehende Singularität (ersichtlich zum Zeitpunkt  $t = 160$ ), die sofort wieder verschwindet. Danach verläuft dieses Beispiel analog zu den obigen. Für weitere Beispiele sei auf das beiliegende Maple-Worksheet verwiesen.

```
> g3 := t -> [6*sin(t)-6*sin(2*t), -6*cos(t)+6*cos(2*t)]:  
> CSF(g1,80,1/10000,7000,10, Plot);
```

### 1.3 BEISPIELE

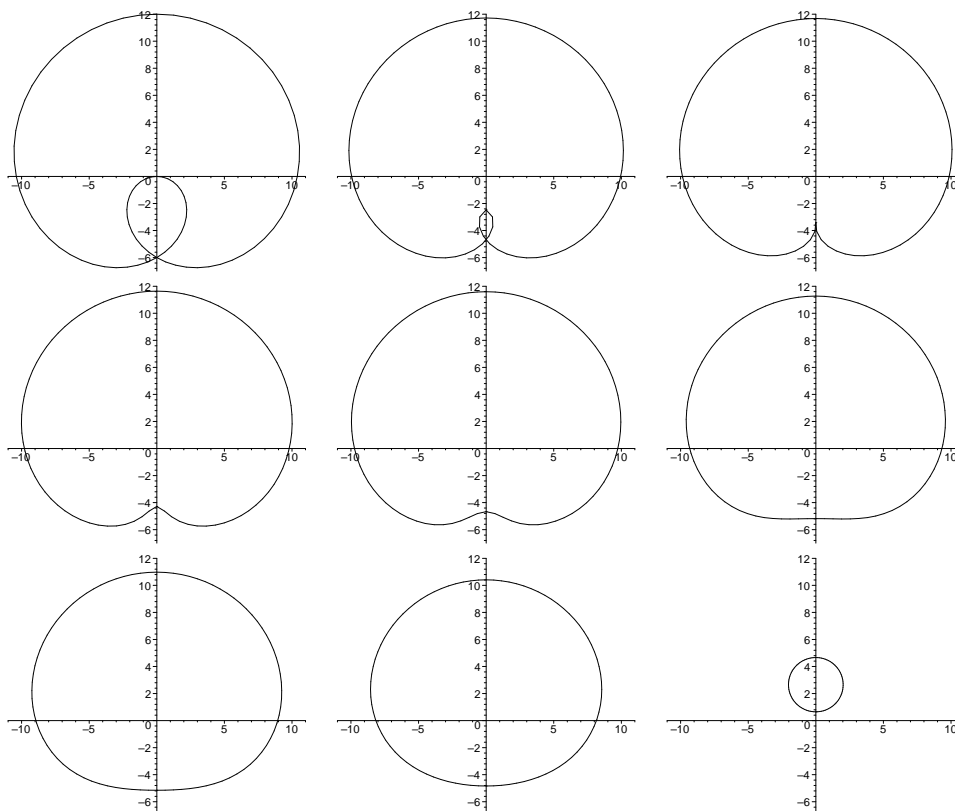


Abbildung 4: Limaçon zu den Zeitpunkten  $t = 0, 140, 160, 180, 200, 360, 500, 760, 2300$  (90 Knoten,  $\tau = \frac{1}{50}$ )

Eine weitere Funktion des Pakets ist die TCSF-Funktion. Mit Hilfe dieser lässt sich die Zeitentwicklung einer geschlossenen Kurve unter dem Einfluss des Curve Shortening Flows in einem Bild visualisieren. Man erkennt an dem von TCSF erzeugten Bild, über welche Kurven sich der Deltoid zu einem Kreis entwickelt. Für weitere Informationen zu dieser und anderen Funktionen des Pakets sei auf die Maple-Hilfeseiten verwiesen.

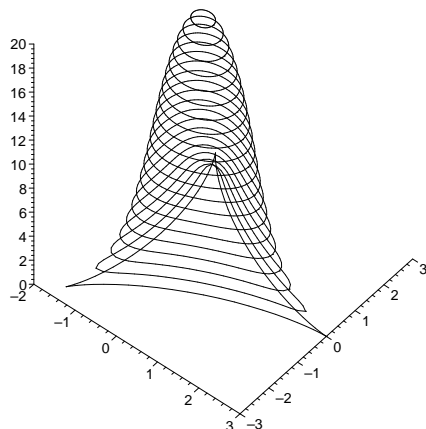


Abbildung 5: Zeitentwicklung des Deltoids mit 80 Knoten und  $\tau = \frac{1}{1000}$

## 2 Elastic Flow

### 2.1 Einleitung

Im Modell einer elastischen Kurve nach Bernoulli - beschrieben durch eine geschlossene Kurve  $f : I \rightarrow \mathbb{R}^n$  - wird die **elastische Energie** definiert durch

$$E(f) = \frac{1}{2} \int_I |\kappa|^2 ds. \quad (8)$$

Mit  $\gamma = |\partial_x f|$  bezeichnet man  $ds = \gamma dx$  als Bogenlängenelement und  $\partial_s = \gamma^{-1} \partial_x$  als Bogenlängenableitung. Dann sind  $\tau = \partial_s f$  die Einheitstangente und  $\kappa = \partial_s^2 f = \partial_s \tau$  der Krümmungsvektor von  $f$ . Der Euler-Lagrange-Operator von  $E$  ist

$$\text{grad}_{L^2} E(f) = \nabla_s^2 \kappa + \frac{1}{2} |\kappa|^2 \kappa. \quad (9)$$

Dabei bezeichnet  $\nabla_s \phi = \partial_s \phi - \langle \partial_s \phi, \tau \rangle \tau$  die Normalenkomponente von  $\partial_s \phi$ . Kritische Punkte von  $E$  bezogen auf konstante Länge werden **Elastika** genannt. Für jedes  $\lambda > 0$  sind die Elastika genau die kritischen Punkte der Energie

$$E_\lambda(f) = E(f) + \lambda L(f). \quad (10)$$

Nun kann man erwarten, dass der Gradient Flow für  $E_\lambda$ , der durch

$$\partial_t f = -\nabla_s^2 \kappa - \frac{1}{2} |\kappa|^2 \kappa + \lambda \kappa \quad (11)$$

definiert wird, eine gegebene Kurve in ein Elastika verwandelt. Betrachtet man alternativ den Gradient Flow von  $E$  bezogen auf konstante Länge, so erhält man wieder (11), wobei jetzt der Parameter  $\lambda$  nicht konstant ist. Es gilt hier

$$\lambda = \frac{\int_I \langle \nabla_s^2 \kappa + \frac{1}{2} |\kappa|^2 \kappa, \kappa \rangle ds}{\int_I |\kappa|^2 ds}. \quad (12)$$

Desweiteren kann man auch den Curve Diffusion Flow, definiert durch

$$\partial_t f = -\nabla_s^2 \kappa, \quad (13)$$

betrachten. Dieser hat für  $n = 2$  die Eigenschaft, dass die eingeschlossene Fläche konstant bleibt.

### 2.2 Numerische Berechnung

Der folgende Algorithmus basiert auf der schwachen Formulierung der Entwicklungsgleichungen. Aus (9) erhält man dabei

$$\nabla_s^2 \kappa + \frac{1}{2} |\kappa|^2 \kappa = \partial_s \left( \partial_s \kappa + \frac{3}{2} |\kappa|^2 \tau \right) \quad (14)$$

## 2.2 NUMERISCHE BERECHNUNG

wie folgt:

Zuerst stellt man fest, dass der Krümmungsvektor  $\kappa$  senkrecht auf der Einheitstangente  $\tau$  steht. Es gilt also

$$\kappa \perp \tau, \quad (15)$$

denn für  $\kappa = \frac{1}{|f_x|} \partial_x \left( \frac{f_x}{|f_x|} \right)$  und  $\tau = \frac{f_x}{|f_x|}$  erhält man

$$\langle \kappa, \tau \rangle = \left\langle \frac{1}{|f_x|} \partial_x \left( \frac{f_x}{|f_x|} \right), \frac{f_x}{|f_x|} \right\rangle = \frac{1}{|f_x|} \left\langle \partial_x \left( \frac{f_x}{|f_x|} \right), \frac{f_x}{|f_x|} \right\rangle = \frac{1}{2|f_x|} \partial_x \underbrace{\left\langle \frac{f_x}{|f_x|}, \frac{f_x}{|f_x|} \right\rangle}_{=1} = 0,$$

wenn man folgendes berücksichtigt:

$$\frac{1}{2} \partial_x \left\langle \frac{f_x}{|f_x|}, \frac{f_x}{|f_x|} \right\rangle = \frac{1}{2} \left( \left\langle \partial_x \left( \frac{f_x}{|f_x|} \right), \frac{f_x}{|f_x|} \right\rangle + \left\langle \frac{f_x}{|f_x|}, \partial_x \left( \frac{f_x}{|f_x|} \right) \right\rangle \right) = \left\langle \partial_x \left( \frac{f_x}{|f_x|} \right), \frac{f_x}{|f_x|} \right\rangle.$$

Wählt man nun  $\phi$  normal, d.h.  $\langle \phi, \tau \rangle = 0$ , so gilt  $\partial_s \langle \phi, \tau \rangle = \langle \partial_s \phi, \tau \rangle + \langle \phi, \partial_s \tau \rangle = 0$ . Hieraus ergibt sich nun die Beziehung

$$\langle \partial_s \phi, \tau \rangle = -\langle \phi, \partial_s \tau \rangle = -\langle \phi, \kappa \rangle$$

und somit letztendlich die Formel

$$\nabla_s \phi = \partial_s \phi - \langle \partial_s \phi, \tau \rangle \tau = \partial_s \phi + \langle \phi, \kappa \rangle \tau. \quad (16)$$

Betrachtet man nun  $\nabla_s^2 \kappa + \frac{1}{2} |\kappa|^2 \kappa$  und benutzt (15), (16) sowie  $\nabla_s \kappa \perp \tau$ , so erhält man

$$\begin{aligned} & \nabla_s^2 \kappa + \frac{1}{2} |\kappa|^2 \kappa \\ &= \partial_s (\nabla_s \kappa) + \langle \nabla_s \kappa, \kappa \rangle \tau + \frac{1}{2} |\kappa|^2 \kappa \\ &= \partial_s \left( \partial_s \kappa + \underbrace{\langle \kappa, \kappa \rangle}_{|\kappa|^2} \tau \right) + \langle \partial_s \kappa + \langle \kappa, \kappa \rangle \tau, \kappa \rangle \tau + \frac{1}{2} |\kappa|^2 \kappa \\ &= \partial_s^2 \kappa + \partial_s (|\kappa|^2 \tau) + \underbrace{\langle \partial_s \kappa, \kappa \rangle \tau}_{=\frac{1}{2} \partial_s (|\kappa|^2 \tau)} + \frac{1}{2} |\kappa|^2 \kappa \\ &= \partial_s^2 \kappa + \frac{3}{2} \partial_s (|\kappa|^2 \tau) \\ &= \partial_s \left( \partial_s \kappa + \frac{3}{2} |\kappa|^2 \tau \right), \end{aligned}$$

wenn man wieder

$$\begin{aligned} & \frac{1}{2} \partial_s (|\kappa|^2 \tau) \\ &= \frac{1}{2} (\partial_s (|\kappa|^2) \tau + |\kappa|^2 \partial_s \tau) = \frac{1}{2} (\partial_s (|\kappa|^2) \tau + |\kappa|^2 \kappa) \\ &= \frac{1}{2} (2 \langle \kappa, \partial_s \kappa \rangle \tau + |\kappa|^2 \kappa) = \langle \partial_s \kappa, \kappa \rangle \tau + \frac{1}{2} |\kappa|^2 \kappa \end{aligned}$$

## 2.2 NUMERISCHE BERECHNUNG

beachtet. Damit ist insgesamt (14) bewiesen.

Um eine Diskretisierung von (11) mit differenzierbaren Funktionen zu vermeiden, schreibt man diese Gleichung um in ein System von Problemen zweiter Ordnung für den Positionsvektor  $f$  und den Krümmungsvektor  $\kappa$ , wobei hier der mittlere Krümmungsvektor verwendet wird. Formt man (14) um zu  $\partial_s^2 \kappa + \frac{3}{2} \partial_s (|\kappa|^2 \tau) - \nabla_s^2 \kappa - \frac{1}{2} |\kappa|^2 \kappa = 0$  und verwendet  $-\nabla_s^2 \kappa - \frac{1}{2} |\kappa|^2 \kappa = \partial_t f - \lambda \kappa$  aus (11) sowie die Definitionen von  $\tau = \partial_s f$  und  $\kappa = \partial_s^2 f$ , so ist (11) äquivalent zum System

$$\partial_t f + \partial_s^2 \kappa + \frac{3}{2} \partial_s (|\kappa|^2 \partial_s f) - \lambda \kappa = 0, \quad (17)$$

$$\kappa - \partial_s^2 f = 0. \quad (18)$$

Verwendet man  $\partial_s = \frac{\partial_x}{|\partial_x f|}$ , multipliziert die Gleichungen mit  $|\partial_x f|$  und integriert, so gilt für alle Testfunktionen  $\varphi, \psi \in H^1(I, \mathbb{R}^n)$

$$\begin{aligned} \int_I \partial_t f |\partial_x f| \varphi \, dx + \int_I \frac{\partial_x^2 \kappa}{|\partial_x f|} \varphi \, dx + \frac{3}{2} \int_I \partial_x \left( |\kappa|^2 \frac{\partial_x f}{|\partial_x f|} \right) \varphi \, dx - \lambda \int_I \kappa |\partial_x f| \varphi \, dx &= 0, \\ \int_I \kappa |\partial_x f| \psi \, dx - \int_I \frac{\partial_x^2 f}{|\partial_x f|} \psi \, dx &= 0. \end{aligned}$$

Wendet man darauf nun die Regel der partiellen Integration an, so erhält man die **schwache Formulierung** von (17), (18): Es gilt für alle Testfunktionen  $\varphi, \psi \in H^1(I, \mathbb{R}^n)$

$$\int_I \partial_t f |\partial_x f| \varphi \, dx - \int_I \frac{\partial_x \kappa}{|\partial_x f|} \partial_x \varphi \, dx - \frac{3}{2} \int_I |\kappa|^2 \frac{\partial_x f}{|\partial_x f|} \partial_x \varphi \, dx - \lambda \int_I \kappa |\partial_x f| \varphi \, dx = 0, \quad (19)$$

$$\int_I \kappa |\partial_x f| \psi \, dx + \int_I \frac{\partial_x f}{|\partial_x f|} \partial_x \psi \, dx = 0. \quad (20)$$

Um ein geeignetes Differenzenschema zu erhalten, wendet man jetzt auf diese schwache Form des Problems eine **räumliche Diskretisierung** an.

Dazu wird das Intervall  $I = \bigcup_{k=1}^N I_k$  zerlegt in Intervalle  $I_k = [x_{k-1}, x_k]$ . Hierbei sind  $x_k$  die Knoten,  $h_k = x_k - x_{k-1}$  die Intervalllängen und  $h = \max_{k=1, \dots, N} h_k$  die maximale Intervalllänge. Um nun (19), (20) zu diskretisieren, wird der Raum  $H^1(I, \mathbb{R}^n)$  der Testfunktionen durch den diskreten Raum der stetigen, stückweise linearen Funktionen  $X_h = \{\omega \in C^0(I, \mathbb{R}^n) \mid \omega|_{I_k} \in \mathbb{P}_1(I_k), k = 1, \dots, N\} = Y_h^n$  ersetzt. Dabei ist  $Y_h$  der Raum der skalaren stückweise linearen Funktionen. Benutzt man die **Basisfunktionen**  $\varphi_k \in Y_h$  mit  $\varphi_k(x_i) = \delta_{ik}$ , so gilt  $X_h = Y_h^n = (\text{span}\{\varphi_1, \dots, \varphi_N\})^n$ . Für die Basisfunktionen und deren stückweise definierten Ableitungen gilt

$$\varphi_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}} = \frac{1}{h_k}(x-x_{k-1}) & , x \in [x_{k-1}, x_k] \\ \frac{x_{k+1}-x}{x_{k+1}-x_k} = \frac{1}{h_{k+1}}(x_{k+1}-x) & , x \in [x_k, x_{k+1}] \\ 0 & , \text{sonst} \end{cases}, \quad (21)$$

$$\partial_x \varphi_k(x) = \begin{cases} \frac{1}{h_k} & , x \in (x_{k-1}, x_k) \\ -\frac{1}{h_{k+1}} & , x \in (x_k, x_{k+1}) \\ 0 & , \text{sonst} \end{cases}. \quad (22)$$

## 2.2 NUMERISCHE BERECHNUNG

Benutzt man nun die punktweise Interpolante  $I_h\omega$  einer Funktion  $\omega \in C^0(I, \mathbb{R}^n)$ , die durch  $I_h\omega \in X_h$  und  $I_h\omega(x_k) = \omega(x_k)$ ,  $k = 1, \dots, N$  eindeutig definiert wird, und setzt die räumlich diskrete Lösung von (19), (20)  $f_h : [0, T] \rightarrow X_h$ ,  $\kappa_h : [0, T] \rightarrow X_h$ ,

$$f_h(x, t) = \sum_{k=1}^N f_k(t)\varphi_k(x), \quad \kappa_h(x, t) = \sum_{k=1}^N \kappa_k(t)\varphi_k(t) \quad (23)$$

ein, so sind  $f_h(\cdot, t), \kappa_h(\cdot, t) \in X_h$ ,  $t \in [0, T]$  gesucht mit  $f_h(\cdot, 0) = f_{h0} = I_h f_0$  und so dass für alle Testfunktionen  $\varphi_h, \psi_h \in X_h$  gilt

$$\int_I I_h(\partial_t f_h \varphi_h) |\partial_x f_h| dx - \int_I \frac{\partial_x \kappa_h}{|\partial_x f_h|} \partial_x \varphi_h dx - \frac{3}{2} \int_I |\kappa_h|^2 \frac{\partial_x f_h}{|\partial_x f_h|} \partial_x \varphi_h dx \quad (24)$$

$$- \lambda \int_I I_h(\kappa_h \varphi_h) |\partial_x f_h| dx = 0,$$

$$\int_I I_h(\kappa_h \psi_h) |\partial_x f_h| dx + \int_I \frac{\partial_x f_h}{|\partial_x f_h|} \partial_x \psi_h dx = 0. \quad (25)$$

Nun setzt man  $\varphi_h = \varphi_j$  und  $\psi_h = \varphi_j$ ,  $j = 1, \dots, N$  einzeln für jede Komponente in diese Gleichungen ein und integriert. Dann liefern die Integrale über  $I$  nur in  $[x_{j-1}, x_j]$  und  $[x_j, x_{j+1}]$  einen Beitrag. Für die einzelnen Terme gilt in  $[x_{j-1}, x_j]$ :

$$f_h = \sum_{k=1}^N f_k \varphi_k = f_{j-1} \varphi_{j-1} + f_j \varphi_j, \quad \partial_x f_h = -\frac{1}{h_j} f_{j-1} + \frac{1}{h_j} f_j = \frac{1}{h_j} (f_j - f_{j-1}), \quad (26)$$

$$|\partial_x f_h| = \frac{1}{h_j} |f_j - f_{j-1}|, \quad \frac{\partial_x f_h}{|\partial_x f_h|} = \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|},$$

$$\kappa_h = \sum_{k=1}^N \kappa_k \varphi_k = \kappa_{j-1} \varphi_{j-1} + \kappa_j \varphi_j, \quad \partial_x \kappa_h = -\frac{1}{h_j} \kappa_{j-1} + \frac{1}{h_j} \kappa_j = \frac{1}{h_j} (\kappa_j - \kappa_{j-1}), \quad (27)$$

$$|\kappa_h|^2 = |\kappa_{j-1}|^2 \varphi_{j-1}^2 + 2\kappa_{j-1} \kappa_j \varphi_{j-1} \varphi_j + |\kappa_j|^2 \varphi_j^2.$$

Analog gilt in  $[x_j, x_{j+1}]$ :

$$f_h = \sum_{k=1}^N f_k \varphi_k = f_j \varphi_j + f_{j+1} \varphi_{j+1}, \quad \partial_x f_h = \frac{1}{h_{j+1}} (f_{j+1} - f_j), \quad (28)$$

$$|\partial_x f_h| = \frac{1}{h_{j+1}} |f_{j+1} - f_j|, \quad \frac{\partial_x f_h}{|\partial_x f_h|} = \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|},$$

$$\kappa_h = \sum_{k=1}^N \kappa_k \varphi_k = \kappa_j \varphi_j + \kappa_{j+1} \varphi_{j+1}, \quad \partial_x \kappa_h = \frac{1}{h_{j+1}} (\kappa_{j+1} - \kappa_j), \quad (29)$$

$$|\kappa_h|^2 = |\kappa_j|^2 \varphi_j^2 + 2\kappa_j \kappa_{j+1} \varphi_j \varphi_{j+1} + |\kappa_{j+1}|^2 \varphi_{j+1}^2.$$

Nun löst man die Ausdrücke für die Interpolante auf. Es gilt

$$\partial_t f_h \varphi_j = \sum_{k=1}^n (\partial_t f_k) \varphi_k \varphi_j = (\partial_t f_{j-1}) \varphi_{j-1} \varphi_j + (\partial_t f_j) \varphi_j^2 + (\partial_t f_{j+1}) \varphi_{j+1} \varphi_j,$$

## 2.2 NUMERISCHE BERECHNUNG

also liefert das Anwenden der Interpolante  $I_h(\partial_t f_h \varphi_j) = (\partial_t f_j) \varphi_j$ . Weiter gilt

$$\kappa_h \varphi_j = \sum_{k=1}^n \kappa_k \varphi_k \varphi_j = \kappa_{j-1} \varphi_{j-1} \varphi_j + \kappa_j \varphi_j^2 + \kappa_{j+1} \varphi_{j+1} \varphi_j,$$

also liefert hier das Anwenden der Interpolante  $I_h(\kappa_h \varphi_j) = \kappa_j \varphi_j$ .

Damit erhält man in (24) für den ersten Summanden

$$\begin{aligned} & \int_I I_h(\partial_t f_h \varphi_j) |\partial_x f_h| dx \\ &= \int_I (\partial_t f_j) |\partial_x f_h| \varphi_j dx \\ &= \int_{x_{j-1}}^{x_j} (\partial_t f_j) |\partial_x f_h| \varphi_j dx + \int_{x_j}^{x_{j+1}} (\partial_t f_j) |\partial_x f_h| \varphi_j dx \\ &= \int_{x_{j-1}}^{x_j} (\partial_t f_j) \frac{1}{h_j} |f_j - f_{j-1}| \varphi_j dx + \int_{x_j}^{x_{j+1}} (\partial_t f_j) \frac{1}{h_{j+1}} |f_{j+1} - f_j| \varphi_j dx \\ &= \partial_t f_j \left( \frac{|f_j - f_{j-1}|}{h_j^2} \int_{x_{j-1}}^{x_j} (x - x_{j-1}) dx + \frac{|f_{j+1} - f_j|}{h_{j+1}^2} \int_{x_j}^{x_{j+1}} (x_{j+1} - x) dx \right) \\ &= \partial_t f_j \left( \frac{|f_j - f_{j-1}|}{h_j} \left( \frac{1}{2} x_j + \frac{1}{2} x_{j-1} - x_{j-1} \right) + \frac{|f_{j+1} - f_j|}{h_{j+1}} \left( x_{j+1} - \frac{1}{2} x_{j+1} - \frac{1}{2} x_j \right) \right) \\ &= \frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) \partial_t f_j. \end{aligned}$$

Hätte man mit dem Ausdruck  $\partial_t f_h \varphi_j$  das obige Integral berechnet, ohne die Interpolante anzuwenden, so hätte man nach längerer Rechnung das Ergebnis

$$\begin{aligned} & \int_I (\partial_t f_h \varphi_j) |\partial_x f_h| dx \\ &= |f_j - f_{j-1}| \left( \frac{1}{6} \partial_t f_{j-1} + \frac{1}{3} \partial_t f_j \right) + |f_{j+1} - f_j| \left( \frac{1}{3} \partial_t f_j + \frac{1}{6} \partial_t f_{j+1} \right) \end{aligned}$$

erhalten. Wendet man hierauf nun Mass Lumping an und ersetzt  $f_{j-1}$  bzw.  $f_{j+1}$  durch  $f_j$ , so erhält man das gleiche Ergebnis  $\frac{1}{2}(|f_j - f_{j-1}| + |f_{j+1} - f_j|) \partial_t f_j$  wie oben. Also entspricht die Anwendung der Interpolante zu Beginn der Rechnung der Verwendung von Mass Lumping.

Mit ähnlichen Rechnungen wie oben lassen sich die anderen Summanden in (24) behandeln:

$$\begin{aligned} & - \int_I \frac{\partial_x \kappa_h}{|\partial_x f_h|} \partial_x \varphi_j dx \\ &= - \frac{\kappa_j - \kappa_{j-1}}{|f_j - f_{j-1}|} \int_{x_{j-1}}^{x_j} \frac{1}{h_j} dx - \frac{\kappa_{j+1} - \kappa_j}{|f_{j+1} - f_j|} \int_{x_j}^{x_{j+1}} \left( -\frac{1}{h_{j+1}} \right) dx \\ &= - \left( \frac{\kappa_j - \kappa_{j-1}}{|f_j - f_{j-1}|} - \frac{\kappa_{j+1} - \kappa_j}{|f_{j+1} - f_j|} \right), \end{aligned}$$



## 2.2 NUMERISCHE BERECHNUNG

$$\begin{aligned}
& -\frac{3}{2} \int_I |\kappa_h|^2 \frac{\partial_x f_h}{|\partial_x f_h|} \partial_x \varphi_j dx \\
& = -\frac{3}{2} \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} \frac{1}{h_j} \int_{x_{j-1}}^{x_j} \left( |\kappa_{j-1}|^2 \frac{(x_j - x)^2}{h_j^2} + 2\kappa_{j-1}\kappa_j \frac{(x_j - x)(x - x_{j-1})}{h_j^2} \right. \\
& \quad \left. + |\kappa_j|^2 \frac{(x - x_{j-1})^2}{h_j^2} \right) dx - \frac{3}{2} \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|} \left( -\frac{1}{h_{j+1}} \right) \int_{x_j}^{x_{j+1}} \left( |\kappa_j|^2 \frac{(x_{j+1} - x)^2}{h_{j+1}^2} \right. \\
& \quad \left. + 2\kappa_j\kappa_{j+1} \frac{(x_{j+1} - x)(x - x_j)}{h_{j+1}^2} + |\kappa_{j+1}|^2 \frac{(x - x_j)^2}{h_{j+1}^2} \right) dx \\
& = -\frac{3}{2} \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} \frac{1}{h_j} \frac{1}{3} h_j (|\kappa_{j-1}|^2 + \kappa_{j-1}\kappa_j + |\kappa_j|^2) \\
& \quad + \frac{3}{2} \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|} \frac{1}{h_{j+1}} \frac{1}{3} h_{j+1} (|\kappa_j|^2 + \kappa_j\kappa_{j+1} + |\kappa_{j+1}|^2) \\
& = -\frac{1}{2} (|\kappa_{j-1}|^2 + \kappa_{j-1}\kappa_j + |\kappa_j|^2) \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} + \frac{1}{2} (|\kappa_j|^2 + \kappa_j\kappa_{j+1} + |\kappa_{j+1}|^2) \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|}, \\
& -\lambda \int_I I_h(\kappa_h \varphi_j) |\partial_x f_h| dx \\
& = -\lambda \int_I \kappa_j |\partial_x f_h| \varphi_j dx \\
& = -\lambda \int_{x_{j-1}}^{x_j} \kappa_j |\partial_x f_h| \varphi_j dx - \lambda \int_{x_j}^{x_{j+1}} \kappa_j |\partial_x f_h| \varphi_j dx \\
& = -\lambda \kappa_j \left( \frac{|f_j - f_{j-1}|}{h_j^2} \int_{x_{j-1}}^{x_j} (x - x_{j-1}) dx + \frac{|f_{j+1} - f_j|}{h_{j+1}^2} \int_{x_j}^{x_{j+1}} (x_{j+1} - x) dx \right) \\
& = -\frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) \lambda \kappa_j,
\end{aligned}$$

wenn man im letzten Summanden die oben berechnete Interpolante einsetzt. Hätte man hier ohne Anwendung der Interpolante gerechnet, so hätte man den komplizierteren Ausdruck

$$-\lambda \frac{|f_j - f_{j-1}|}{h_j} \left( \frac{1}{6} h_j \kappa_{j-1} + \frac{1}{3} h_j \kappa_j \right) - \lambda \frac{|f_{j+1} - f_j|}{h_{j+1}} \left( \frac{1}{3} h_{j+1} \kappa_j + \frac{1}{6} h_{j+1} \kappa_{j+1} \right)$$

erhalten, der sich erst durch die Anwendung von Mass Lumping zum obigen Ergebnis  $-\frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) \lambda \kappa_j$  vereinfacht hätte.

Analog erhält man für die beiden Summanden in (25), wenn man nochmal die gerade verwendete Interpolante einsetzt:

## 2.2 NUMERISCHE BERECHNUNG

$$\begin{aligned}
& \int_I I_h(\kappa_h \varphi_j) |\partial_x f_h| dx \\
&= \int_I \kappa_j |\partial_x f_h| \varphi_j dx \\
&= \kappa_j \left( \frac{|f_j - f_{j-1}|}{h_j^2} \int_{x_{j-1}}^{x_j} (x - x_{j-1}) dx + \frac{|f_{j+1} - f_j|}{h_{j+1}^2} \int_{x_j}^{x_{j+1}} (x_{j+1} - x) dx \right) \\
&= \frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) \kappa_j, \\
& \int_I \frac{\partial_x f_h}{|\partial_x f_h|} \partial_x \varphi_j dx \\
&= \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} \int_{x_{j-1}}^{x_j} \frac{1}{h_j} dx + \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|} \int_{x_j}^{x_{j+1}} \left( -\frac{1}{h_{j+1}} \right) dx \\
&= \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} - \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|}.
\end{aligned}$$

Fasst man die obigen Ergebnisse zusammen, so sind insgesamt die diskreten Gleichungen in schwacher Form (24), (25) äquivalent zu dem folgenden System von  $n \cdot N$  ODEs

$$\frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) (\partial_t f_j - \lambda \kappa_j) - \left( \frac{\kappa_j - \kappa_{j-1}}{|f_j - f_{j-1}|} - \frac{\kappa_{j+1} - \kappa_j}{|f_{j+1} - f_j|} \right) \quad (30)$$

$$\begin{aligned}
& - \frac{1}{2} (|\kappa_{j-1}|^2 + \kappa_{j-1} \kappa_j + |\kappa_j|^2) \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} \\
& + \frac{1}{2} (|\kappa_j|^2 + \kappa_j \kappa_{j+1} + |\kappa_{j+1}|^2) \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|} = 0,
\end{aligned}$$

$$\frac{1}{2} (|f_j - f_{j-1}| + |f_{j+1} - f_j|) \kappa_j + \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} - \frac{f_{j+1} - f_j}{|f_{j+1} - f_j|} = 0, \quad (31)$$

$j = 1, \dots, N$ , mit  $f_0 = f_N, f_{N+1} = f_1, \kappa_0 = \kappa_N, \kappa_{N+1} = \kappa_1$  und den Anfangswerten  $f_j(0) = f_0(x_j), j = 1, \dots, N$ .

Für die **zeitliche Diskretisierung** von (30), (31) wird nun die Notation  $\omega^m = \omega(\cdot, m\tau)$  für die Entwicklung einer Funktion im  $m$ -ten Zeitschritt verwendet. Dabei ist  $\tau$  die Zeitschrittweite mit  $\tau M = T$ . Man erhält

$$\frac{1}{2} (|f_j^m - f_{j-1}^m| + |f_{j+1}^m - f_j^m|) \left( \frac{1}{\tau} (f_j^{m+1} - f_j^m) - \lambda \kappa_j^{m+1} \right) \quad (32)$$

$$- \left( \frac{\kappa_j^{m+1} - \kappa_{j-1}^{m+1}}{|f_j^m - f_{j-1}^m|} - \frac{\kappa_{j+1}^{m+1} - \kappa_j^{m+1}}{|f_{j+1}^m - f_j^m|} \right)$$

$$\begin{aligned}
& - \frac{1}{2} (|\kappa_{j-1}^m|^2 + \kappa_{j-1}^m \kappa_j^m + |\kappa_j^m|^2) \frac{f_j^{m+1} - f_{j-1}^{m+1}}{|f_j^m - f_{j-1}^m|} \\
& + \frac{1}{2} (|\kappa_j^m|^2 + \kappa_j^m \kappa_{j+1}^m + |\kappa_{j+1}^m|^2) \frac{f_{j+1}^{m+1} - f_j^{m+1}}{|f_{j+1}^m - f_j^m|} = 0,
\end{aligned}$$

$$\frac{1}{2} (|f_j^m - f_{j-1}^m| + |f_{j+1}^m - f_j^m|) \kappa_j^{m+1} + \frac{f_j^{m+1} - f_{j-1}^{m+1}}{|f_j^m - f_{j-1}^m|} - \frac{f_{j+1}^{m+1} - f_j^{m+1}}{|f_{j+1}^m - f_j^m|} = 0. \quad (33)$$

## 2.2 NUMERISCHE BERECHNUNG

Mit den abkürzenden Schreibweisen  $h_j^m = |f_j^m - f_{j-1}^m|$ ,  $\beta_j^m = |\kappa_j^m|^2 + \kappa_{j-1}^m \kappa_j^m + |\kappa_j^m|^2$ ,  $\lambda_j^m = \lambda$  gilt weiter

$$\frac{1}{2}(h_j^m + h_{j+1}^m) \left( \frac{1}{\tau} (f_j^{m+1} - f_j^m) - \lambda_j^m \kappa_j^{m+1} \right) - \left( \frac{\kappa_j^{m+1} - \kappa_{j-1}^{m+1}}{h_j^m} - \frac{\kappa_{j+1}^{m+1} - \kappa_j^{m+1}}{h_{j+1}^m} \right) \quad (34)$$

$$- \frac{1}{2} \beta_j^m \frac{f_j^{m+1} - f_{j-1}^{m+1}}{h_j^m} + \frac{1}{2} \beta_{j+1}^m \frac{f_{j+1}^{m+1} - f_j^{m+1}}{h_{j+1}^m} = 0,$$

$$\frac{1}{2} (h_j^m + h_{j+1}^m) \kappa_j^{m+1} + \frac{f_j^{m+1} - f_{j-1}^{m+1}}{h_j^m} - \frac{f_{j+1}^{m+1} - f_j^{m+1}}{h_{j+1}^m} = 0. \quad (35)$$

Durch Umstellung von (34), (35) lässt sich damit der Algorithmus für den Elastic Flow mit festem Parameter  $\lambda$  für Kurven im  $\mathbb{R}^n$  formulieren.

### Algorithmus 1 (Elastic Flow With Fixed Parameter).

Für  $j = 1, \dots, N$  setze  $f_j^0 = f_0(x_j)$ ,  $h_j^0 = |f_j^0 - f_{j-1}^0|$  und

$$\kappa_j^0 = \frac{2}{h_{j+1}^0 (h_j^0 + h_{j+1}^0)} f_{j+1}^0 - \frac{2}{h_j^0 h_{j+1}^0} f_j^0 + \frac{2}{h_j^0 (h_j^0 + h_{j+1}^0)} f_{j-1}^0.$$

Für  $m = 0, \dots, M-1$  berechne

$$h_j^m = |f_j^m - f_{j-1}^m|, \quad \beta_j^m = |\kappa_{j-1}^m|^2 + \kappa_{j-1}^m \kappa_j^m + |\kappa_j^m|^2, \quad \lambda_j^m = \lambda \quad (36)$$

und berechne  $f_j^{m+1} \in \mathbb{R}^n$  und  $\kappa_j^{m+1} \in \mathbb{R}^n$  ( $j = 1, \dots, N$ ) aus dem linearen System

$$\frac{\beta_j^m}{2h_j^m} f_{j-1}^{m+1} + \left( \frac{h_j^m + h_{j+1}^m}{2\tau} - \frac{\beta_j^m}{2h_j^m} - \frac{\beta_{j+1}^m}{2h_{j+1}^m} \right) f_j^{m+1} + \frac{\beta_{j+1}^m}{2h_{j+1}^m} f_{j+1}^{m+1} \quad (37)$$

$$+ \frac{1}{h_j^m} \kappa_{j-1}^{m+1} - \left( \frac{1}{h_j^m} + \frac{1}{h_{j+1}^m} + \frac{\lambda_j^m}{2} (h_j^m + h_{j+1}^m) \right) \kappa_j^{m+1} + \frac{1}{h_{j+1}^m} \kappa_{j+1}^{m+1}$$

$$= \frac{h_j^m + h_{j+1}^m}{2\tau} f_j^m,$$

$$- \frac{1}{h_j^m} f_{j-1}^{m+1} + \left( \frac{1}{h_j^m} + \frac{1}{h_{j+1}^m} \right) f_j^{m+1} - \frac{1}{h_{j+1}^m} f_{j+1}^{m+1} \quad (38)$$

$$+ \frac{1}{2} (h_j^m + h_{j+1}^m) \kappa_j^{m+1} = 0$$

( $j = 1, \dots, N$ ) mit den periodischen Randbedingungen  $f_{i+N} = f_i$ ,  $\kappa_{i+N} = \kappa_i$  ( $i \in \mathbb{Z}$ ).

Im Folgenden soll nun die Länge  $L(f)$  der Kurve konstant gehalten werden. Hierzu wird der obige Algorithmus leicht abgeändert. Dazu muss  $\lambda$  in jedem Zeitschritt gemäß (12) berechnet werden. Wir bedienen uns dazu der schon hergeleiteten schwachen Form (24),

## 2.2 NUMERISCHE BERECHNUNG

(25) und setzen  $\varphi_h = \kappa_h$  und  $\psi_h = \partial_t f_h = f_{ht}$ . Also gilt  $\partial_x \varphi_h = \varphi_{hx} = \kappa_{hx}$ . Die Formel (24) liefert

$$- \int_I \frac{|\kappa_{hx}|^2}{|f_{hx}|} dx - \frac{3}{2} \int_I |\kappa_h|^2 \frac{f_{hx}}{|f_{hx}|} \kappa_{hx} dx - \lambda \int_I |\kappa_h|^2 |f_{hx}| dx = - \int_I I_h(f_{ht} \kappa_h) |f_{hx}| dx. \quad (39)$$

Wendet man nun (25) auf die rechte Seite an und differenziert unter dem Integral unter Beachtung von  $\partial_t(|f_{hx}|) = \frac{f_{hx}}{|f_{hx}|} f_{hxt}$ , so erhält man

$$\int_I I_h(f_{ht} \kappa_h) |f_{hx}| dx = \frac{d}{dt} \int_I |f_{hx}| dx.$$

Man berechnet

$$\int_I |f_{hx}| dx = \sum_{i=1}^N |f_i - f_{i-1}| = L(f).$$

Da im folgenden die Länge  $L(f)$  konstant gehalten werden soll bzgl. der Zeit, ist  $\frac{d}{dt} L(f) = 0$ . So kann man jetzt (39) nach  $\lambda$  auflösen:

$$\lambda = - \left( \int_I \frac{|\kappa_{hx}|^2}{|f_{hx}|} dx + \frac{3}{2} \int_I |\kappa_h|^2 \frac{f_{hx}}{|f_{hx}|} \kappa_{hx} dx \right) / \int_I |\kappa_h|^2 |f_{hx}| dx.$$

Nun wertet man diese Integrale einzeln aus unter Beachtung der obigen Definitionen von  $\kappa_h$ ,  $f_h$  und den Basisfunktionen  $\varphi_j$ . Dabei ist  $I = \bigcup_{j=1}^N I_j = \bigcup_{j=1}^N [x_{j-1}, x_j]$ , so dass man erhält:

$$\begin{aligned} & \int_I \frac{|\kappa_{hx}|^2}{|f_{hx}|} dx \\ &= \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \frac{|\sum_{i=1}^N \kappa_i \varphi_{ix}|^2}{|\sum_{i=1}^N f_i \varphi_{ix}|} dx \\ &= \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \frac{|\kappa_j \frac{1}{h_j} - \kappa_{j-1} \frac{1}{h_j}|^2}{|f_j \frac{1}{h_j} - f_{j-1} \frac{1}{h_j}|} dx \\ &= \sum_{j=1}^N \frac{1}{h_j} \frac{|\kappa_j - \kappa_{j-1}|^2}{|f_j - f_{j-1}|} h_j. \end{aligned}$$

Diskretisiert man diesen Term zeitlich, so erhält man

$$\sum_{j=1}^N \frac{|\kappa_j^m - \kappa_{j-1}^m|^2}{h_j^m}.$$

## 2.2 NUMERISCHE BERECHNUNG

Weiter gilt

$$\begin{aligned}
& \int_I |\kappa_h|^2 \frac{f_{hx}}{|f_{hx}|} \kappa_{hx} dx \\
&= \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \left| \sum_{i=1}^N \kappa_i \varphi_i \right|^2 \frac{\sum_{i=1}^N f_i \varphi_{ix}}{\left| \sum_{i=1}^N f_i \varphi_{ix} \right|} \sum_{i=1}^N \kappa_i \varphi_{ix} dx \\
&= \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \left| \kappa_j (x - x_{j-1}) \frac{1}{h_j} + \kappa_{j-1} (x_j - x) \frac{1}{h_j} \right|^2 \frac{f_j \frac{1}{h_j} - f_{j-1} \frac{1}{h_j}}{\left| f_j \frac{1}{h_j} - f_{j-1} \frac{1}{h_j} \right|} \left( \kappa_j \frac{1}{h_j} - \kappa_{j-1} \frac{1}{h_j} \right) dx \\
&= \sum_{j=1}^N \frac{1}{h_j^3} (\kappa_j - \kappa_{j-1}) \frac{f_j - f_{j-1}}{|f_j - f_{j-1}|} \int_{x_{j-1}}^{x_j} \sum_{i=1}^N \left( \kappa_j^i (x - x_{j-1}) - \kappa_{j-1}^i (x - x_j) \right)^2 dx.
\end{aligned}$$

In der inneren Summe werden nun die quadratischen Summanden mit der binomischen Formel ausgewertet. Betrachtet man nur die Summe und integriert diese, so erhält man

$$\begin{aligned}
& \int_{x_{j-1}}^{x_j} \sum_{i=1}^N (\kappa_j^i)^2 (x - x_{j-1})^2 - 2 (x^2 - (x_{j-1} + x_j)x + x_{j-1}x_j) \kappa_j^i \kappa_{j-1}^i + (\kappa_{j-1}^i)^2 (x - x_j)^2 dx \\
&= \sum_{i=1}^N \frac{(\kappa_j^i)^2}{3} (x_j - x_{j-1})^3 - 2 \kappa_j^i \kappa_{j-1}^i \left( \frac{1}{3} x_j^3 - (x_{j-1} + x_j) \frac{x_j^2}{2} + x_{j-1} x_j^2 \right) \\
&\quad - 2 \kappa_j^i \kappa_{j-1}^i \left( -\frac{1}{3} x_{j-1}^3 + (x_{j-1} + x_j) \frac{x_{j-1}^2}{2} - x_{j-1}^2 x_j \right) - \frac{(\kappa_{j-1}^i)^2}{3} (x_{j-1} - x_j)^3
\end{aligned}$$

Nun benutzt man zur Vereinfachung  $h_j = x_j - x_{j-1}$  und verwendet weiter binomische Formeln.

$$\begin{aligned}
&= \sum_{i=1}^N \frac{(\kappa_j^i)^2 + (\kappa_{j-1}^i)^2}{3} h_j^3 - 2 \kappa_j^i \kappa_{j-1}^i \left( \frac{1}{3} h_j (x_j^2 + x_j x_{j-1} + x_{j-1}^2) \right. \\
&\quad \left. - \frac{1}{2} h_j (x_j + x_{j-1})^2 + x_{j-1} x_j^2 - x_{j-1}^2 x_j \right) \\
&= \sum_{i=1}^N \frac{(\kappa_j^i)^2 + (\kappa_{j-1}^i)^2}{3} h_j^3 - 2 \kappa_j^i \kappa_{j-1}^i h_j \frac{1}{6} (-x_j^2 + 2x_j x_{j-1} - x_{j-1}^2) \\
&= \sum_{i=1}^N \frac{(\kappa_j^i)^2 + (\kappa_{j-1}^i)^2}{3} h_j^3 - 2 \kappa_j^i \kappa_{j-1}^i h_j \left( -\frac{1}{6} h_j^2 \right).
\end{aligned}$$

Damit erhält man für das obige Integral insgesamt

$$\begin{aligned}
& \int_I |\kappa_h|^2 \frac{f_{hx}}{|f_{hx}|} \kappa_{hx} dx \\
&= \sum_{j=1}^N (\kappa_j - \kappa_{j-1}) \frac{(f_j - f_{j-1})}{|f_j - f_{j-1}|} \frac{1}{3} \sum_{i=1}^N (\kappa_j^i)^2 + \kappa_j^i \kappa_{j-1}^i + (\kappa_{j-1}^i)^2.
\end{aligned}$$

Diskretisiert man auch hier zeitlich, so erhält man

$$\sum_{j=1}^N (\kappa_j^m - \kappa_{j-1}^m) \frac{(f_j^m - f_{j-1}^m)}{h_j^m} \frac{1}{3} \beta_j^m.$$

Schließlich betrachtet man noch den Nenner:

$$\begin{aligned} & \int_I |\kappa_h|^2 |f_{hx}| dx \\ &= \sum_{j=1}^N \int_{x_{j-1}}^{x_j} \left| \kappa_j (x - x_{j-1}) \frac{1}{h_j} - \kappa_{j-1} (x - x_j) \frac{1}{h_j} \right|^2 |f_j - f_{j-1}| \frac{1}{h_j} dx. \end{aligned}$$

Dabei erkennt man denselben quadratischen Term wie oben und erhält die zeitlich diskretisierte Form

$$\sum_{j=1}^N \frac{1}{3} |f_j^m - f_{j-1}^m| \beta_j^m = \frac{1}{3} \sum_{j=1}^N h_j^m \beta_j^m.$$

Setzt man die einzelnen Integrale zusammen, so erhält man den Algorithmus für den Elastic Flow mit Längenerhaltung.

**Algorithmus 2 (Length Preserving Elastic Flow).**

Ersetze die Berechnung von  $\lambda_j^m$  in (36) in Algorithmus 1 durch

$$\lambda_j^m = -3 \left( \sum_{i=1}^N \frac{|\kappa_i^m - \kappa_{i-1}^m|^2}{h_i^m} + \frac{1}{2} \sum_{i=1}^N (f_i^m - f_{i-1}^m) (\kappa_i^m - \kappa_{i-1}^m) \frac{\beta_i^m}{h_i^m} \right) / \sum_{i=1}^N h_i^m \beta_i^m. \quad (40)$$

Bis auf einen Fehler der Größe  $\tau$  durch die zeitliche Diskretisierung von  $\lambda$  bleibt die Länge des Polygons bei diesem Algorithmus erhalten.

Betrachtet man das obige Beispiel (13)  $\partial_t f = -\nabla_s^2 \kappa$  und vergleicht mit dem Ausdruck (11)  $\partial_t f = -\nabla_s^2 \kappa - \frac{1}{2} |\kappa|^2 \kappa + \lambda \kappa$ , so folgt sofort  $\lambda = \frac{1}{2} |\kappa|^2$ . Damit ist das Problem der Curve Diffusion (13) äquivalent zu dem System

$$\partial_t f + \partial_s^2 \kappa + \frac{3}{2} \partial_s (|\kappa|^2 \partial_s f) - \frac{1}{2} |\kappa|^2 \kappa = 0, \quad (41)$$

$$\kappa - \partial_s^2 f = 0. \quad (42)$$

Mit einer analogen Diskretisierung wie vorher für den Elastic Flow mit festem Parameter und den Elastic Flow mit Längenerhaltung kommt man auch hier zum Algorithmus für Curve Diffusion.

**Algorithmus 3 (Curve Diffusion).**

Ersetze die Berechnung von  $\lambda_j^m$  in (36) in Algorithmus 1 durch

$$\lambda_j^m = \frac{1}{2} |\kappa_j^m|^2. \quad (43)$$

## 2.3 Beispiele

Die Wirkung des Elastic Flow auf Kurven im  $\mathbb{R}^3$  in den Fällen Curve Diffusion, Längenerhaltung und fester Parameter  $\lambda$  soll nun anhand von drei einfachen Beispielen verdeutlicht werden. Auch hier kann man schon typische Effekte ablesen. Für weitere Beispiele sei auf das beiliegende Maple-Worksheet verwiesen. Dort werden teils sehr einfache Beispiele behandelt, wie der Kreis, dessen Verhalten von der Theorie berechnet werden kann und an dem abgelesen werden kann, ob die Implementierung Fehler aufweist. Teils werden auch kompliziertere Beispiele, wie z.B. Hypozykloiden, ein Double Circle oder weitere Kurven mit vielen Verschlingungen, behandelt.

Bei der Anwendung des Programms EF im Maple-Paket muss man auf die Wahl der Parameter achten. Es gibt Konstellationen, bei denen der Algorithmus falsche Ergebnisse liefert. Dies kann u.a. dann auftreten, wenn die Zeitschrittweite  $\tau$  zu groß gewählt wurde. Die Bedeutung der Parameter, die dem Programm EF übergeben werden, wird auf den Hilfeseiten des Maple-Pakets ausführlich erläutert.

### 2.3.1 Sattel

Beim Sattel erkennt man, wie durch den elastischen Fluss die elastische Energie minimiert wird, d.h. der Sattel wird in vertikaler Richtung gerade gebogen und erreicht den Endzustand des Kreises. Der folgende Aufruf zeigt die Entwicklung des Sattels als Animation in Maple.

```
> g1:=t->[sin(t),cos(t),sin(t)*cos(t)]:
> EF(g1, 150, 1/2000, 1500, 10, -2, Plot);
```

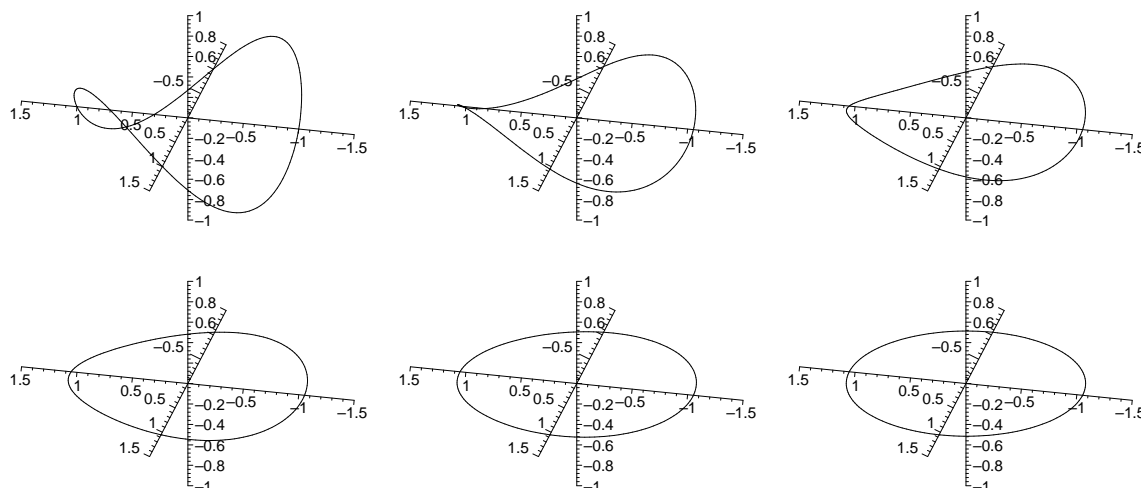


Abbildung 6: Sattel zu den Zeitpunkten  $t = 0, 150, 300, 450, 700, 1200$  (150 Knoten, Curve Diffusion,  $\tau = \frac{1}{2000}$ )

## 2.3 BEISPIELE

### 2.3.2 Lemniskate

Die gekrümmte Lemniskate verhält sich zunächst ähnlich wie der Sattel und biegt sich gerade. Dann wird ein stationärer Punkt erreicht und die Veränderungen zwischen den Zeitpunkten  $t = 500$  und  $t = 3500$  sind minimal. Der stationäre Punkt ist aber nicht stabil, denn allmählich beginnt die Lemniskate, sich zu entknoten und aufzubiegen, bis schließlich wieder der Endzustand des Kreises erreicht ist. Die Entwicklung lässt sich verfolgen durch den folgenden Aufruf.

```
> g2:=t->[sin(sin(t))^2*cos(t),sin(t)*cos(t)^3]:
> EF(g2, 150, 1/1000, 5000, 50, -1, Plot);
```

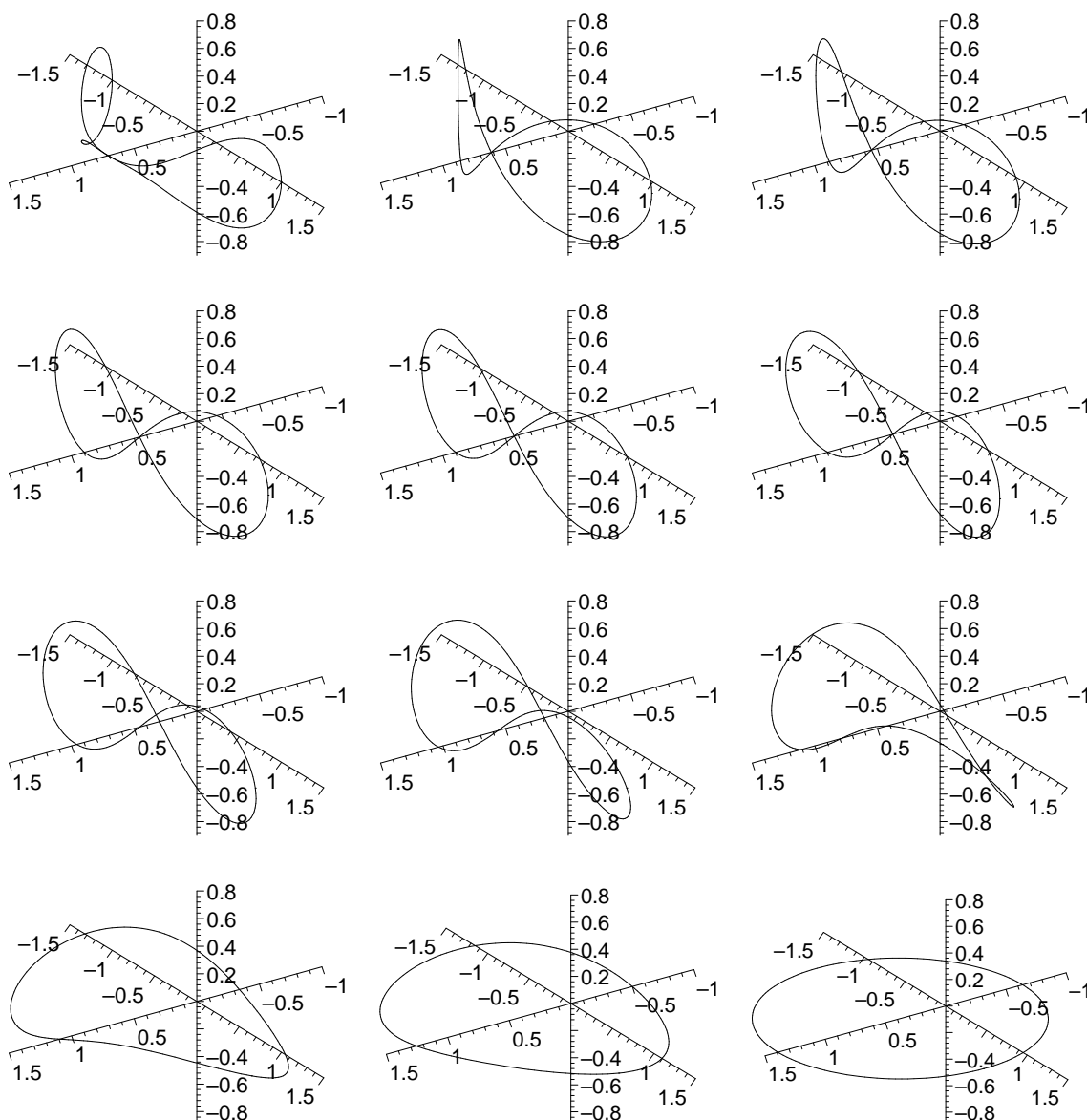


Abbildung 7: Lemniskate zu den Zeitpunkten  $t = 0, 100, 200, 400, 500, 2000, 4000, 4150, 4300, 4450, 4550, 5000$  (150 Knoten, Längenerhaltung,  $\tau = \frac{1}{1000}$ )



## 2.3 BEISPIELE

### 2.3.3 Verschlingungskurve

Auch bei dieser verschlungenen Kurve erkennt man, wie durch die Minimierung der elastischen Energie die Kurve gerade gebogen und dadurch entknotet wird. Im Unterschied zum Sattel lässt sich hier kein stationärer Zustand beobachten. Durch den folgenden Aufruf erhält man die Entwicklung und kontinuierliche Entknotung der Verschlingungskurve.

```
> g3:=t->[sin(7*t),cos(5*t),3*sin(t)*cos(t)]:  
> EF(g3, 150, 1/1000, 2500, 20, 1.2, Plot);
```

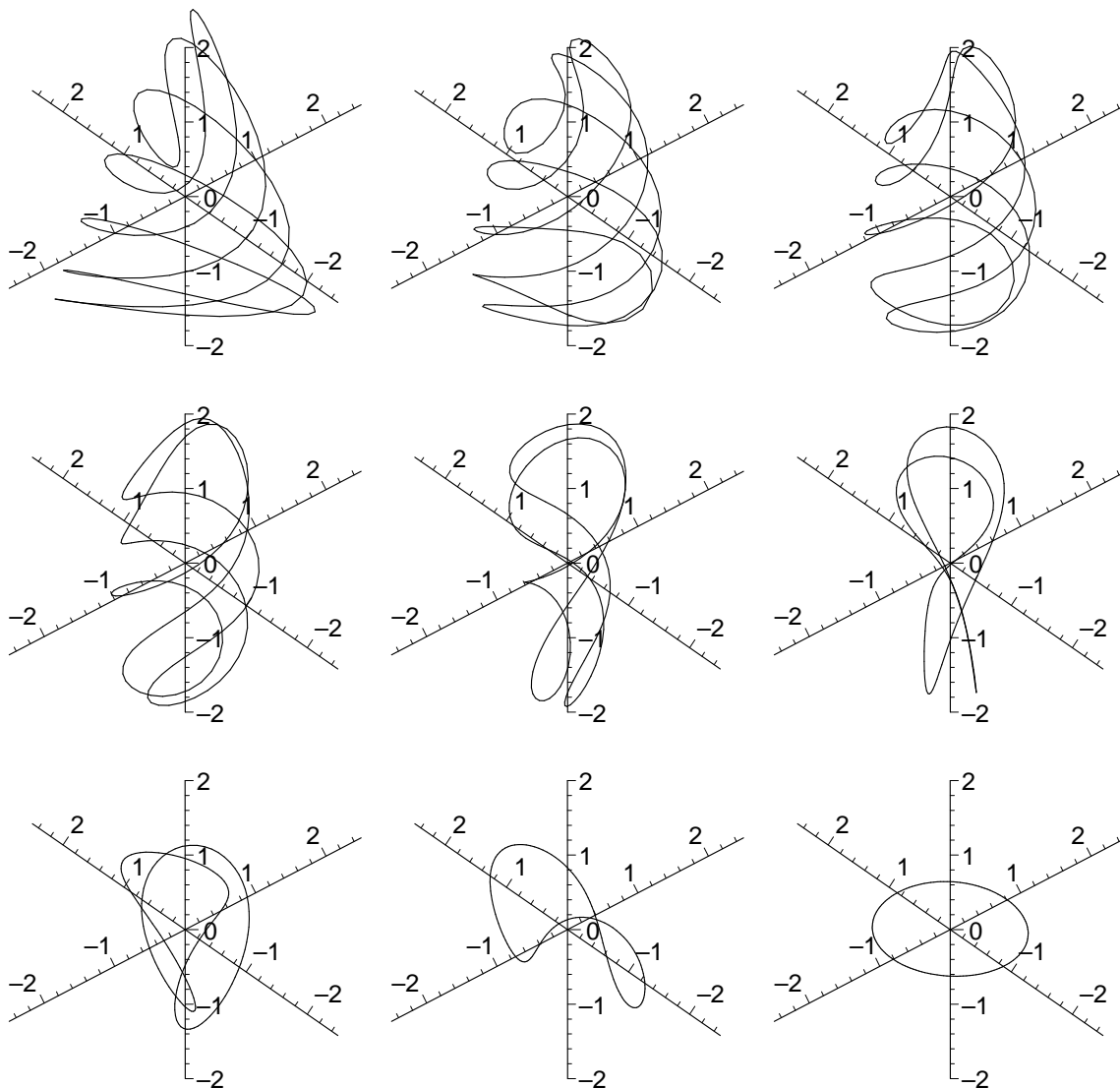


Abbildung 8: Verschlingungskurve zu den Zeitpunkten  $t = 0, 80, 180, 320, 500, 700, 1140, 1580, 2000$  (150 Knoten, fester Parameter  $\lambda = 1.2$ ,  $\tau = \frac{1}{1000}$ )

## 3 Implementierung und Methoden

### 3.1 Implementierung

Um die oben beschriebenen Algorithmen zum Curve Shortening Flow und Elastic Flow nun zu implementieren, haben wir Maple und C++ als Programme bzw. Programmiersprachen benutzt. Alle Dateien, die wir erstellt haben, haben wir in dem Maple-Paket `cf1ow` zusammengefasst. Der Benutzer kann das komplette Programm von Maple aus bedienen. Dabei ruft Maple ein C++-Programm auf, um alle aufwendigen Rechnungen durchzuführen. Diese sind insbesondere das Lösen der Gleichungssysteme (7) bzw. (37), (38) sowie die Aktualisierung der im Algorithmus auftretenden Variablen. Maple wird nur benutzt, um Parameter und Startkonfigurationen an das C++-Programm zu übergeben und abschließend die erzeugten Daten graphisch darzustellen. Es ist unerheblich, ob die Kurve als Parametrisierung vorliegt, oder ob eine Punktmenge zur Festlegung der Kurve eingegeben wird. Die einzelnen Funktionen, die das Maple-Paket `cf1ow` zum Curve Shortening Flow und zum Elastic Flow bietet, werden näher in den Hilfeseiten zu dem Paket beschrieben und dort auch mit Beispielen erläutert.

Ein zentraler Bestandteil zur numerischen Behandlung der beiden Flüsse ist das Lösen von linearen Gleichungssystemen. Dabei spielen die Faktoren

- Stabilität und
- Geschwindigkeit

eine wesentliche Rolle.

Da die auftretenden Probleme in der Implementierung des Curve Shortening Flows analog zu bearbeiten sind, wird im folgenden nur noch der elastische Fluss von räumlichen Kurven betrachtet.

Dies hat zur Folge, dass man 3 Gleichungssysteme der Größe  $2N \times 2N$  lösen muss, wobei  $N$  die Anzahl der Stützstellen ist, wie man an den Gleichungen (37), (38) erkennen kann. Die Effizienz des Löser spielt bei Gleichungssystemen dieser Größenordnung also eine wichtige Rolle.

Wir haben zu Beginn eine LR-Zerlegung mit Zeilenskalierung und Spaltenpivotsuche eingesetzt, die einen Aufwand von  $O(N^3)$  hat und einmalig durchgeführt werden muss in jedem Schritt des Algorithmus. Das Lösen der 3 Systeme geschieht durch Vorwärts- und Rückwärtseinsetzen, was jeweils einen Aufwand von  $O(N^2)$  hat.

Der Vorteil hierbei ist die Stabilität des Verfahrens, auf die gerade in der Anfangsphase des Projektes Wert gelegt wurde, um die Fehlersuche zu erleichtern. Implementiert man den Algorithmus mit dem oben beschriebenen Verfahren, ergibt sich jedoch ein weiteres Problem mit der Stabilität der berechneten Lösungen. Die Kurven deformieren sich wider Erwarten, d.h. sie entwickeln sich nicht so, wie von der Theorie vorhergesagt. Es ergibt

### 3.1 IMPLEMENTIERUNG

sich, dass einige Punkte der Kurve sich aufeinander zu bewegen. Dies hat zur Folge, dass die  $h_j^m$  aus (36), die gerade den Abstand der Kurvenpunkte beschreiben, sehr klein werden und Instabilitäten verursachen.

Eine Lösung dieses Problems ergibt sich, wenn man in jedem Schritt des Algorithmus die Kurve nach Bogenlänge parametrisiert. Ein ähnliches Ergebnis erhält man, indem man die Knoten auf der Kurve in jedem Schritt neu verteilt, und zwar in der Form, dass der Abstand der Knoten immer der gleiche ist. Man beachte dabei jedoch, dass in der diskretisierten Form nicht mehr die ursprüngliche Kurve vorliegt, sondern ein Polygon. Also liegen die neu verteilten Knoten nicht mehr zwangsläufig auf der eigentlichen Kurve. Man verändert also die Kurve in jedem Schritt des Algorithmus. Aber auch bei der Parametrisierung nach Bogenlänge ergibt sich das Problem, dass man in der diskretisierten Form nicht mehr die ursprüngliche Kurve hat, sondern ein Polygon, welches allein durch seine Eckpunkte bestimmt ist. Ändert man diese ab, so verändert man das Polygon zwangsläufig.

Man kann sich leicht klar machen, dass sich, selbst bei der Variante der konstanten Länge (40), die Länge des Polygons etwas verkürzt. Dazu betrachte man z.B. den folgenden zweidimensionalen Fall, bei dem das durch Neuverteilung entstandene Polygon immer innerhalb des vorherigen liegt, die Knotenpunkte jedoch auf dem vorherigen Polygon liegen.

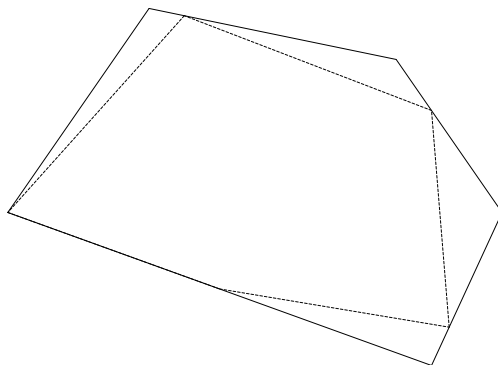


Abbildung 9: Veränderung eines Polygons bei der Neuverteilung der Knoten

Wir haben experimentell festgestellt, dass diese Verkürzung und damit auch die Veränderung des Polygons relativ klein ist, wenn die Neuverteilung erstens in jedem Schritt stattfindet und zweitens eine genügend große Anzahl von Knoten verwendet wird. Man betrachte dazu den Kreis mit Längenerhaltung. Hier lässt sich die Länge leicht exakt berechnen und der relative Fehler liegt im Bereich von  $10^{-4}$  nach 25 Schritten und bei  $10^{-3}$  nach 50 Schritten, wenn man 50 Knoten zu Grunde legt. Wählt man z.B. 100 Knoten, so ist der Fehler auch nach 100 Schritten bei  $5 \cdot 10^{-4}$ ; als Zeitschrittweite wurde  $\tau = \frac{1}{100}$  gewählt.

Durch die Neuverteilung der Knoten hat sich nun die Kondition der Systemmatrix soweit verbessert, dass die LR-Zerlegung auch ohne Spaltenpivotsuche auskommt, d.h. es müssen keine Zeilen mehr getauscht werden. Dies hat den großen Vorteil, dass ohne Zeilentauschen die Struktur der Systemmatrix erhalten bleibt und nun ausgenutzt werden

### 3.1 IMPLEMENTIERUNG

kann, dass die Matrix viele Einträge besitzt, die identisch 0 sind. Durch Umsortierung der Gleichungen und Variablen erhält man im wesentlichen eine (3,3)-Bandmatrix als Systemmatrix, die aufgrund der Periodizitätsbedingung zusätzlich noch je 4 Einträge in der rechten oberen und linken unteren Ecke enthält. Insgesamt besitzt jede Zeile nur 6 von 0 verschiedene Einträge. Dabei ist mit einer  $(p,q)$ -Bandmatrix eine Diagonalmatrix gemeint, bei der auch die  $p$  Nebendiagonalen unterhalb und die  $q$  Nebendiagonalen oberhalb der Diagonalen noch mit Werten ungleich 0 belegt sind.

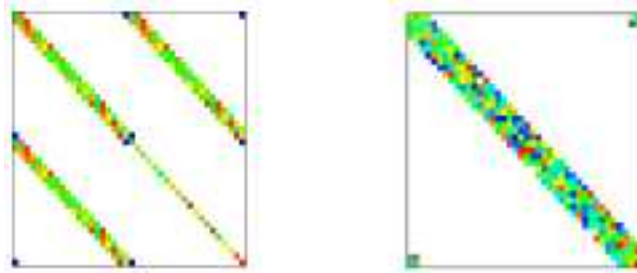


Abbildung 10: Systemmatrix vor und nach der Umordnung

Die besondere Struktur der Systemmatrix ermöglicht nun die Optimierung des Lösers. Bezeichnet man die linke untere Dreiecksmatrix der LR-Zerlegung mit  $L$  und die rechte obere Dreiecksmatrix mit  $R$ , so ist die Matrix  $L$  eine (3,0)-Bandmatrix, bei der die letzten beiden Zeilen auch noch mit Werten belegt sind. Die Matrix  $R$  ist eine (0,3)-Bandmatrix, bei der zusätzlich die letzten beiden Spalten mit Werten belegt sind. Dies resultiert wiederum aus der Bedingung der Periodizität. Die besondere Struktur von  $L$  und  $R$  als Bandmatrizen erlaubt es, bei der Berechnung der LR-Zerlegung nur die von 0 verschiedenen Einträge zu berechnen bzw. zu berücksichtigen. Die Einträge, die identisch 0 sind, werden außerdem bei der Zeilenskalierung, d.h. bei der Berechnung der Zeilensummen, sowie dem Vorwärts- und dem Rückwärtseinsetzen ausgenutzt. Man erhält damit insgesamt eine Laufzeit von  $O(N)$ .

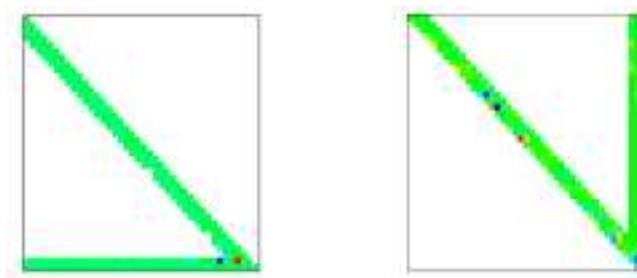


Abbildung 11: LR-Zerlegung der umsortierten Systemmatrix

Die Optimierung der Laufzeit zeigt sich z.B. an der gekrümmten Lemniskate aus Beispiel 2.3.2. Hier verringert sich die Berechnungszeit für 10000 Schritte von 7 Minuten auf 30 Sekunden. Der Geschwindigkeitsvorteil nimmt mit wachsender Knoten- und Zeitschrittzahl zu.

## 3.2 Neuverteilung der Knoten

Um die Implementierung der Algorithmen exemplarisch vorzustellen, soll im folgenden die oben beschriebene Neuverteilung der Knoten erläutert werden. In der konkreten Implementierung in C++ haben wir die Neuverteilung der Knoten beim Elastic Flow wie folgt realisiert:

```
// Neuverteilung der Knoten entlang eines Polygons gegeben durch f
void neuverteilung(double** f, double** fe){
    double sw_tmp;
    double sw_aktuell;
    double norm;

    double sw = laenge(fe)/N; // Schrittweite
    int i = 1; // mit i werden neue Knoten durchlaufen
    int j = 1; // mit j werden alte Knoten durchlaufen

    while (i < N) {
        sw_tmp = sw; // restliche Schrittweite
        sw_aktuell = sqrt( pow((f[i-1][0] - fe[j][0]),2) +
            pow((f[i-1][1] - fe[j][1]),2) + pow((f[i-1][2] - fe[j][2]),2));
        // Abstand zum nächsten alten Knoten
        f[i][0] = f[i-1][0]; // i. neuen Knoten vorinitialisieren
        f[i][1] = f[i-1][1];
        f[i][2] = f[i-1][2];

        while (sw_tmp > sw_aktuell) {
            sw_tmp -= sw_aktuell;
            f[i][0] = fe[j][0]; // i. neuen Knoten zwischenzeitlich neusetzen
            f[i][1] = fe[j][1];
            f[i][2] = fe[j][2];
            j=(j+1)%N;
            sw_aktuell = sqrt( pow((fe[j][0] - f[i][0]),2) +
                pow((fe[j][1] - f[i][1]),2) + pow((fe[j][2] - f[i][2]),2));
        }

        norm = sqrt( pow(fe[j][0] - f[i][0],2) + pow((fe[j][1] - f[i][1]),2) +
            pow((fe[j][2] - f[i][2]),2));
        f[i][0] = f[i][0] + sw_tmp/norm * (fe[j][0] - f[i][0]);
        // i. neuen Knoten endgültig neusetzen
        f[i][1] = f[i][1] + sw_tmp/norm * (fe[j][1] - f[i][1]);
        f[i][2] = f[i][2] + sw_tmp/norm * (fe[j][2] - f[i][2]);
        i++;
    }
}
```

Das Polygon und eine Kopie werden in den Doppelarrays `f` und `fe` übergeben. Dividiert man die Länge des Polygons, die mit einer separaten Funktion in C++ berechnet wird, durch die Anzahl der Knoten  $N$ , so erhält man die Schrittweite `sw`. Diese Schrittweite gibt die Strecke an, die jeweils entlang des alten Polygons gelaufen werden muss, bis ein neuer Knoten gesetzt wird. Nun beginnt man im ersten alten Knoten, der gleichzeitig der erste neue Knoten ist, das alte Polygon zu durchlaufen. Die alten Knoten werden dabei

### 3.3 MASS LUMPING

in der Variable  $i$ , die neuen in der Variable  $j$  durchgezählt. Die Schrittweite  $sw$  wird in der restlichen Schrittweite  $sw\_tmp$  gespeichert. In  $sw\_aktuell$  wird der Abstand entlang des alten Polygons zum nächsten alten Knoten gespeichert.

Nun sind zwei Fälle zu unterscheiden: Ist die restliche Schrittweite  $sw\_tmp$  größer als der Abstand  $sw\_aktuell$  zum nächsten alten Knoten, so zieht man  $sw\_aktuell$  von der restlichen Schrittweite  $sw\_tmp$  ab und läuft entlang des alten Polygons zum nächsten alten Knoten. Dieses Vorgehen wiederholt sich so lange, bis der zweite Fall eintritt, dass nämlich  $sw\_tmp$  kleiner gleich  $sw\_aktuell$  ist. Dann muss der neue Knoten auf dem Abschnitt des alten Polygons bis zum nächsten alten Knoten gesetzt werden. Dies geschieht nun, indem man auf dem alten Polygon noch um eine gewisse Strecke  $d$  weiterläuft. Dabei bezeichnet  $d$  den Quotienten aus restlicher Schrittweite  $sw\_tmp$  und Abstand  $norm$  zum nächsten alten Knoten. Die Idee zum Setzen des neuen Knotens in dieser Weise ergibt sich aus der Konstruktion einer Gerade durch Stütz- und Richtungsvektor. Die Einzelheiten zur Setzung des neuen Knotens sind am obigen Quellcode abzulesen; aus technischen Gründen muss der Knoten schon vorher vorinitialisiert bzw. zwischenzeitlich neugesetzt werden. Hat man den neuen Knoten nun endgültig neugesetzt, wiederholt sich das Vorgehen, bis alle alten Knoten durchlaufen sind. Die neuen Knoten werden sofort im Doppelarray  $f$  gespeichert.

Die Neuverteilung der Knoten wird im Algorithmus nach jedem Zeitschritt aufgerufen und auf das Polygon angewendet.

### 3.3 Mass Lumping

Sowohl beim Curve Shortening Flow als auch beim Elastic Flow wurde zur Herleitung der Algorithmen das sogenannte Mass Lumping verwendet. Mass Lumping wird häufig im Zusammenhang mit Finite-Elemente-Methoden zur Diagonalisierung eines vorhandenen linearen Gleichungssystems eingesetzt. Das hier verwendete Verfahren zur Diagonalisierung ist das Zeilensummenverfahren. Dabei wird die Massenmatrix  $M = (m_{ij})_{ij} \in \mathbb{R}^{n \times n}$  ersetzt durch die Diagonalmatrix  $\tilde{M} = (\tilde{m}_{ij})_{ij} \in \mathbb{R}^{n \times n}$  mit  $m_{ii} = \sum_{j=1}^n m_{ij}$ . Dadurch lässt sich die Inverse der Matrix und somit auch die Lösung direkt angeben. Falls bei dem Zeilensummenverfahren Nullzeilen auftauchen, muss ein anderes Verfahren (z.B. Quadraturverfahren) angewendet werden.

In dieser Arbeit wurde dieses Verfahren auf lineare Differentialgleichungssysteme anstatt auf lineare Gleichungssysteme angewendet.

## Literatur

- [1] Klaus Deckelnick, Gerhard Dziuk, and Charles M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, pages 1 – 94, 2005.
- [2] Gerhard Dziuk, Ernst Kuwert, and Reiner Schätzle. Evolution of elastic curves in  $\mathbb{R}^n$ : Existence and computation. *SIAM J. Math. Anal.*, 33(5):1228 – 1245, 2002.
- [3] Michael E. Gage. An isoperimetric inequality with applications to curve shortening. *Duke Mathematical Journal*, 50(4), December 1983.