Simulation von hängenden Textilien für eine automatische Bilderkennung

Ausarbeitung der Bachelorarbeit
Vorgelegt an der Universität Stuttgart

Autor:

Marius Stach

Datum:

27.12.2019

Betreuer:

Benjamin Maier

Prüferin:

Professorin Miriam Mehl

IPVS

Universität Stuttgart

Inhaltsverzeichnis

1 Kurzfassung **1**

2 Einleitung (Propaedeuticum) **2**
2.1 Motivation 2
2.2 Aufbau der Arbeit 2
2.3 Verwandte Arbeiten 2

3 Modellierung und Lösung **11**
3.1 Diskretisierung des Textils 11
3.2 Semi-Implizite Lösung des Modells 20
3.3 Kollisionsfreiheit 26

4 Implementierung **32**
4.1 Ablauf des gesamten Tools 32
4.2 Wahl einer geeigneten Programmiersprache 34
4.3 Zeitschrittverfahren zur Lösung der Differentialgleichung 34
4.4 Löser für das Lineare Gleichungssystem 35
4.5 Dynamische Zeitschrittweite des Zeitschrittverfahrens 35
4.6 Parallelisierung der Berechnungen 35
4.7 Speicherung der Gitterpositionen in einer Matrix 36
4.8 Fortsetzen einer begonnenen Simulation 37
4.9 Bilderkennung mit Matlab 37

5 Ergebnisse **39**
5.1 Entwicklung der Laufzeit 39
5.2 Effizienz der adaptiven Zeitschrittweitenregelung 40
5.3 Auswertung des Bildervergleichs in Matlab 41
5.4 Numerische Experimente 42

6 Zusammenfassung und Ausblick **51**

7 Anhang **52**

8 Literatur **58**

Das zweite Kapitel dieser Ausarbeitung ist als Propaedeuticum der Bachelorarbeit entstanden und getrennt zu bewerten.

Eigenständigkeitserklärung

Hiermit bestätige ich, Marius Stach,

- dass ich diese Ausarbeitung selbstständig verfasst habe,
- dass ich keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe,
- dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist und
- dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Stuttgart, den 27. Dezember 2019

Marius Stach

Notation

$\mathbf{x}_r$	Position des r-ten Gitterpunktes
$\mathbf{v}_r$	Geschwindigkeit des r-ten Gitterpunktes
$\mathbf{a}_r$	Beschleunigung des r-ten Gitterpunktes
$\mathbf{x}_n$	Vektor der Positionen aller Gitterpunkte zum Zeitpunkt t_n
$\mathbf{v}_n$	Vektor der Geschwindigkeiten aller Gitterpunkte zum Zeitpunkt t_n
p_0	Position des aktuell betrachteten Gitterpunktes
$p_j, \quad j = 1, ..., 4$	Positionen der Nachbarpunkte von p_0
p_{j+1}	Position des gegen den Uhrzeigersinn nächsten Nachbarn
p_{j+2}	Position des gegen den Uhrzeigersinn übernächsten Nachbarn
r_{0j}	Abstand zwischen p_0 und p_j
$\phi_j, j \in \{1, 2, 3, 4\}$	Winkel zwischen p_j und p_{j+1}
$\Phi_j, j \in \{1, 2\}$	Winkel zwischen p_j und p_{j+2}
$\phi(\mathbf{k}_1, \mathbf{k}_2)$	Winkelfunktion zwischen den Vektoren $\mathbf{k}_1, \mathbf{k}_2$
$\mathbf{k}_1$	Erste Komponente der Winkelfunktion
$\mathbf{k}_2$	Zweite Komponente der Winkelfunktion
$E_{r,j}$	Energiefunktion des r-ten Gitterpunkts bezogen auf Nachbar j
$\otimes$	Dyadenprodukt zweier Vektoren mit $[a \otimes b](i, j) = a_i b_j$
$\mathbf{I}$	Einheitsmatrix
C^0	Koeffizient zur Skalierung der Spannungsenergie
C^s	Koeffizient zur Skalierung der Scherenergie
C^b	Koeffizient zur Skalierung der Biegungsenergie
ρ	Dichte des Fluids, durch welches das Textil fällt
c_w	Widerstandsbeiwert
A_r	Fläche des von Reibung behafteten Teils des Textils
m_r	Masse des r-ten Gitterpunktes
$\mathbf{f}_n$	Vektor der Werte der Bewegungsgleichung für alle Gitterpunkte zum Zeitpunkt t_n

1 Kurzfassung

Das Einführen von Textilien in Wasch- und Mangelanlagen muss in der industriellen Textilproduktion bisher von Hand erfolgen und stellt einen ungesunden Arbeitsplatz in heißer und feuchter Umgebung dar. Um diese Aufgabe zu automatisieren sollen Roboter Abhilfe schaffen, welche die Textilien greifen und dabei automatisch erkennen und richtig weiterverarbeiten. Das Ziel dieser Arbeit ist es deshalb, die Form und Lage eines hängenden Textils anhand von Kameradaten zu erkennen. Um dies zu erreichen, werden die Kameradaten mit einer vorher erstellten Datenbank von simulierten Lagen deformierter Textilien abgeglichen.
Zuerst wird dazu ein partikelbasierter Modellansatz für das Verhalten von Textilien nach [Eberhardt et al., 1996] vorgestellt. Anhand der dort definierten Energiegleichungen erfolgt mithilfe der Lagrangeschen Betrachtungsweise die Aufstellung der Bewegungsgleichung des Systems in Eigenarbeit. Um diese Bewegungsgleichung effizienter zu lösen, wird anschließend eine Semi-Impliziter Formulierung nach einem Ansatz aus [Baraff und Witkin, 1998] gewählt. Die so berechnete Bewegungsgleichung wird zur Lösung in ein System von Differentialgleichungen erster Ordnung umgeschrieben. Als nächstes wird der Ablauf des fertigen Tools zur automatischen Bilderkennung im Gesamten beschrieben. Danach wird die Implementierung der Simulation sowie des zum Abgleich von Kameradaten und Simulationsergebnissen nötigen Programms vorgestellt. Als letztes werden Ergebnisse aus Laufzeit und numerischen Experimenten vorgestellt sowie ein Ausblick auf mögliche weiterführende Arbeiten gegeben.

2 Einleitung (Propaedeuticum)

2.1 Motivation

Das Waschen und Mangeln von Textilen ist ein Prozess, der in industriellen Anlagen gut automatisiert werden kann. Nur das Einführen der Textilien in die Anlage muss bisher von Hand erfolgen und stellt einen ungesunden Arbeitsplatz in heißer und feuchter Umgebung dar. Um diese Aufgabe zu automatisieren, sollen Roboter zum Einsatz kommen.
Die Aufgabe des Roboters ist es dabei, aus einem ungeordneten Haufen Kleidungsstücke und andere Textilien zu entnehmen und korrekt in die Maschine einzulegen. Der Vorgang des Einlegens wird dabei für im Vorhinein festgelegte Greifpositionen einprogrammiert. Der Roboter muss also die ungefähre Form der Textilien erkennen, um daraus auf die Positionen der Greifpositionen zu schließen.
Ziel dieser Arbeit ist es deshalb, die Form und Lage eines hängenden Textils anhand von Kameradaten zu erkennen. Dazu wird ein Simulationsprogramm für das Verhalten hängender Textilien entwickelt, mit dessen Ergebnissen dann Bilddaten hängender Textilien abgeglichen werden können. Für diesen Abgleich wird ein Algorithmus in Matlab entworfen. So kann die Form des hängenden Textils abgeschätzt und potentielle Greifpositionen zur Weiterverarbeitung angegeben werden.

2.2 Aufbau der Arbeit

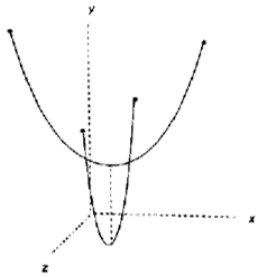
Im folgenden Verlauf des zweiten Kapitels werden verwandte Arbeiten vorgestellt und diskutiert. Anschließend wird in Kapitel 3 das in der Simulation der Textilien verwendete Modell vorgestellt. Dabei wird zuerst die Gesamtenergie des Systems definiert und daraus anhand der Lagrangeschen Betrachtungsweise die Bewegungsgleichung des Systems aufgestellt. Diese wird anschließend in eine Semi-Implizite Form überführt. Im darauffolgenden Kapitel 4 wird zuerst die Funktionsweise des fertigen Tools vom Aufbau der Datenbank von Simulationsergebnnisen bis zum Bildervergleich mit Kameradaten vorgestellt. Anschließend werden die Details der Implementierung der Simulation sowie der Bilderkennung erläutert. Danach werden in Kapitel 5 Ergebnisse der Arbeit aus Laufzeit und numerischen Experimenten präsentiert, bevor die gesamte Arbeit in Kapitel 6 zusammengefasst wird. Ebenfalls wird dort ein Ausblick auf mögliche zukünftige Arbeiten gegeben.

2.3 Verwandte Arbeiten

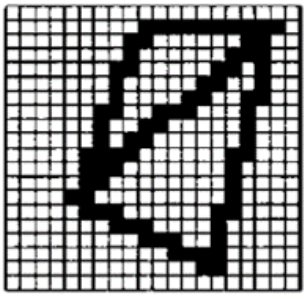
Schon seit vielen Jahrhunderten beschäftigt die Menschen die realitätsgetreue Darstellung von Textilien. In früher Zeit war dies hauptsächlich ein Problem, welches Maler und Bildhauer beschäftigte. In Zeiten moderner Computergraphik ist dieses Problem nun wieder sehr aktuell.
Im folgenden Abschnitt wird der Inhalt verwandter Arbeiten zu diesem Thema erläutert sowie die Differenzen zu dieser Arbeit aufgezeigt.

2.3.1 Geometrische Modellierung

Die ersten Ansätze zur Modellierung von Textilien basieren auf geometrischen Methoden. Hierbei wird die Form des Textils durch mathematische Funktionen angenähert. Meist werden dazu die Lagen einzelner Fäden des Textils durch eine mathemtische Funktion beschrieben. Aus der Gesamtheit dieser Beschreibungen ergibt sich dann die Form des deformierten Textils. Der von [Weil, 1986] vorgestellte Ansatz zielt darauf ab, die Endposition eines deformierten Textils zu bestimmen, welches an einigen definierten Punkten festgehalten wird. Dazu wird die Form des Textils als rechteckig vorausgesetzt und durch ein Gitter diskretisiert. In zwei Schritten wird zuerst eine grobe Approximation des hängenden Textils und anschließend die Endposition berechnet. Im ersten Schritt wird dabei iterativ die ungefähre Position eines jeden Gitterpunkts in der deformierten Lage geschätzt. Bei dieser Annäherung wird ausgenutzt, dass die exakten Endpositionen der festgehaltenen Punkte bereits bekannt sind, da sie sich während der Simulation nicht ändern. Es wird dabei für alle Greifpunkte, mit Hilfe der mechanischen Seiltheorie und des initialen Abstands der Punkte, die Lage eines Fadens berechnet, welcher an diesen beiden Punkten gehalten wird. Diese Lage wird dabei durch eine parabelförmige Funktion beschrieben. Für den Fall, dass zwei dieser Fäden direkt übereinander liegen, wird der untere der beiden entfernt. Die Lage dieser Fäden zwischen den Greifpunkten sowie des Problemfalls zweier übereinanderliegender Fäden ist in Abbildung 1 a) dargestellt.



(a) Lage der ersten Fäden zwischen den Greifpunkten

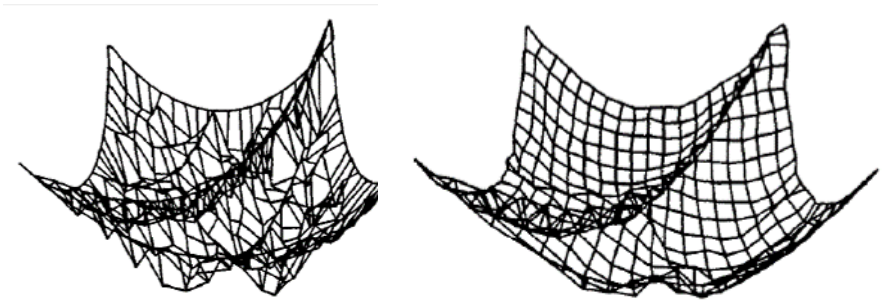


(b) Aufteilung des Textils in Dreiecke

Abbildung 1: Geometrische Modellierung, Grafiken aus [Weil, 1986]

Durch die so berechneten Fäden zwischen vier Greifpunkten wurde das zu simulierende Textil wie in Abbildung 1 b) dargestellt in Dreiecke aufgeteilt. Im weiteren Verlauf der ersten Approximation der Lage des Textils wird nun iterativ jedes entstandene Dreieck in zwei neue Dreiecke aufgeteilt und die Lage eines Fadens zwischen den neu entstandenen Eckpunkten berechnet. Dieser Vorgang wird so lange fortgesetzt bis alle Gitterpunkte abgebildet worden sind. In diesem ersten Schritt werden somit zwar die physikalischen Eigenschaften der einzelnen hängenden Fäden berücksichtigt, jedoch die Eigenschaften des Textils als Ganzes vernachlässigt. Die Autoren merkten deshalb an, dass die Genauigkeit dieser Methode begrenzt ist,

da zum Beispiel die für Textilien typische Faltenbildung nicht abgebildet werden kann.
Der zweite Schritt der Modellierung ist dann gegeben durch die Relaxationsphase, in welcher die physikalischen Eigenschaften des gesamten Textils berücksichtigt werden sollen. Dazu wird für jeden Punkt ein sogenannter Deformationsvektor bestimmt, in dessen Richtung der Punkt verschoben wird. Dieser Vektor ergibt sich aus den Positionen des Gitterpunkts und seiner vier Nachbarn. Ebenso wird aus diesen Positionen die Länge der Strecke berechnet, um die der Gitterpunkt später verschoben wird. Dieser Schritt wird für alle Gitterpunkte so lange durchgeführt, bis die größte Verschiebung eines Punktes unter einen gewissen Wert fällt. Eine so berechnete Position eines deformierten Textils nach der ersten Relaxationsphase sowie die vorangegangene Approximation aus dem ersten Schritt sind in Abbildung 2 dargestellt.



(a) Positionen der Punkte nach der ersten Schätzung

(b) Positionen der Punkte nach der ersten Relaxation

Abbildung 2: Visualisierung der Positionen der Gitterpunkte nach dem ersten und zweiten Schritt, Grafiken aus [Weil, 1986]

Als Nachteil dieses Schritts führen die Autoren auf, dass eine mögliche Selbstdurchdringung des Textils weder erkannt noch korrigiert wird.
Spätere Arbeiten wie [Zhang und Yuen, 2001] betonen zwar die hohe Geschwindigkeit der geometrischen Methoden, bestätigen aber auch ihre mangelnde Genauigkeit aufgrund der bereits erwähnten Nachteile. Insgesamt sind die geometrischen Ansätze für eine realitätsgetreue Darstellung von Textilien also ungeeignet, da die physikalischen Eigenschaften des gesamten Textils, wie etwa die Faltenbildung, nicht abgebildet werden.

2.3.2 Physikalische Modellierung

Aufgrund der mangelnden Genauigkeit der geometrischen Modelle wurden im Laufe der Zeit neue Modelle entwickelt, welche die physikalischen Eigenschaften der Textilien berücksichtigen können. Diese Modelle lassen sich in die drei folgenden Kategorien aufteilen:

- Kontinuumsmechanische Modelle

- Masse-Feder Modelle
- Partikelbasierte Modelle

Im folgenden werden die Vor- und Nachteile jedes Modellansatzes aufgezeigt.

Kontinuumsmechanische Modelle Bei diesen Modellansätzen wird das Textil mit Mitteln der Kontinuumsmechanik dargestellt. So werden Textilien beispielsweise als dünne Platten [Lindberg et al., 1961] dargestellt. In [Eischen et al., 1996] wurde ein Modell vorgestellt, welches die geometrisch exakte Schalen-Theorie verwendet. Dabei wurde der Ansatz gewählt, das zu modellierende Textil durch mehrere Schalenelemente zu diskretisieren. Für jedes Schalenelement erfolgt hierbei die Berücksichtigung der Dicke sowie der Steifigkeit. Somit können die physikalischen Eigenschaften des Textils in das Modell einfließen. Anschließend wird die Bewegungsgleichung des Systems anhand der Schalentheorie aufgestellt. Es ergibt sich dadurch ein lineares Gleichungssystem der Form

$$\mathbf{K}\Delta\mathbf{\Phi} = \mathbf{F} - \mathbf{P} \tag{1}$$

Hierbei steht $\mathbf{K}$ für die Steifigkeitsmatrix des Systems aus Schalenelementen und $\Delta\mathbf{\Phi}$ für die Positionsänderung der Schalenelemente in diesem Zeitschritt. Auf der rechten Seite der Gleichung steht $\mathbf{F}$ für die externen Kräfte auf die Elemente und $\mathbf{P}$ für die intern wirkenden Kräfte.
Der Vorteil dieser Modellierung besteht darin, dass in jedem Zeitschritt nur ein lineares Gleichungssystem gelöst werden muss.
Ein anderer großer Vorteil dieser Ansätze ist, dass dafür auch bereits vorhandene kommerzielle Software zur Simulation benutzt werden kann, da die meisten kommerziellen Finite Elemente Modelle über Schalenelemente verfügen. Da die kontinuumsmechanischen Modelle jedoch nicht speziell für diesen Anwendungsfall konstruiert wurden, ergeben sich auch einige Nachteile dieses Ansatzes. So muss nach [Eischen et al., 1996] der Zeitschritt in der ersten Phase der Berechnung sehr klein gewählt werden, um Divergenz zu vermeiden. So entsteht dabei ein 'exzessiver Anteil an Rechenzeit', welcher die Effizienz des Verfahrens deutlich einschränkt.
Ebenso führen die bei der Simulation von Textilien verhältnismäßig großen Deformationen zu Konvergenzproblemen, welche sich in einem dauerhaft sehr kleinen Zeitschritt der Simulation äußern. So sind die Berechnungen nach diesen Modellen sehr zeitaufwändig und die Genauigkeit der Abbildung ist aufgrund der großen Deformationen nach [Magnenat-Thalmann et al., 2007] sehr gering. Insgesamt sind diese Modelle also trotz ihrer schnellen Verfügbarkeit für diese Arbeit ungeeignet.

Masse-Feder Modelle Bei sogenannten Masse-Feder Modellen wird ein Textil als Netz von Massepunkten, welche durch Federn miteinander verbunden sind, diskretisiert. Im einfachsten Fall ergibt sich nach [Magnenat-Thalmann et al., 2007] ein in Abbildung 3 zu sehendes, rechteckiges Gitter aus $m \times n$ Massenpunkten, die durch jeweils 4 Federn mit ihren direkten Nachbarn verbunden sind.

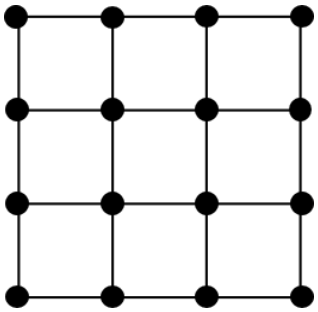


Abbildung 3: Einfachste Form einer Masse-Feder Diskretisierung

In jedem Berechnungsschritt werden nun für jeden Massenpunkt und jeden seiner Nachbarn die wirkenden Federkräfte anhand der Differenz des aktuellen und des ursprünglichen Abstands der Massenpunkte bestimmt. Zusätzlich wird für jeden Massenpunkt die Schwerkraft berücksichtigt. Durch die Superposition dieser Kräfte kann für jeden Massenpunkt die gesamte auf ihn wirkende Kraft berechnet werden. Es ergibt sich nach der Newtonschen Mechanik für jeden Gitterpunkt eine Bewegungsgleichung der Form:

$$\ddot{\mathbf{x}} = \frac{F_{ges}}{m} \tag{2}$$

Diese Einfachheit der Beschreibung senkt die Berechnungskosten enorm. Jedoch kann durch solch vereinfachte Modelle das Verhalten von Textilien nur sehr ungenau dargestellt werden. So ist es zum Beispiel nicht möglich die Biegung und Faltenbildung von Textilien abzubilden. Ebenso lässt sich das nichtlineare Materialverhalten von Textilien mit Federn nicht modellieren, da Federkräfte immer linear mit dem Abstand zunehmen. Um diesr Ungenauigkeit entgegenzuwirken, wird in [Provot et al., 1995] die Einführung weiterer Federn zu weiter entfernten Nachbarn vorgeschlagen. Eine Mögliche Variante ist in Abbildung 4 gezeigt.

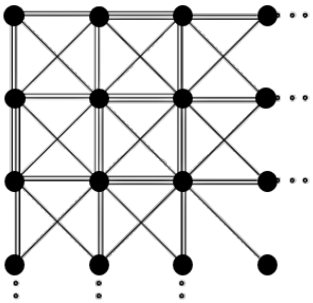


Abbildung 4: Variante einer Masse-Feder Diskretisierung, Grafik aus [Provot et al., 1995]

Mit diesem Ansatz ist es nun möglich auch das Biegeverhalten von Textilien darzustellen. Die zusätzlichen Federn verkomplizieren jedoch die Berechnungen, da zur

Berechnung der insgesamt wirkenden Kraft deutlich mehr Massepunkte berücksichtigt werden müssen. Um die durch die Erweiterung des Modells enstandenen höheren Berechnungskosten zu senken, wurde in [Zhang und Yuen, 2001] ein Modell mit schrittweise verfeinertem Gitter vorgestellt. Hierbei wird mit einem groben Gitter von Massepunkten zuerst eine erste Annäherung der deformierten Lage berechnet, bevor das Gitter verfeinert wird. Dieser Prozess lässt sich mehrere Male wiederholen, bis eine hinreichend hohe Genauigkeit vorliegt. Dieser Ansatz ermöglicht es, zu Beginn der Berechnung ein gröberes Gitter zu verwenden und so die Berechnungskosten zu senken. Durch das später iterativ verfeinerte Gitter bleibt die Qualität der Abbildung jedoch gegeben.
Mit diesen Erweiterungen büßen die Masse-Feder Modelle jedoch ihre anfängliche Einfachheit ein. Ebenfalls kann durch diese Modelle aufgrund der verwendeten Federn nur lineares Materialverhalten abgebildet werden. Da das Deformationsverhalten von Textilien jedoch hochgradig nichtlinear ist, kann dieses nach [Breen et al., 1994] durch Masse-Feder Modelle nicht realitätsgetreu abgebildet werden. Somit ist auch diese Art der Modellierung von Textilien für diese Arbeit ungeeignet.

Partikelbasierte Modelle Um die anfängliche Einfachheit der Masse-Feder Modelle zu erhalten, aber gleichzeitig das Deformationsverhalten von Textilien realitätsgetreu abbilden zu können, schlägt [Breen et al., 1994] einen Partikelbasierten Ansatz vor. In diesem Ansatz wird das sich deformierende Textil durch ein rechteckiges Gitter von $m \times n$ untereinander interagierenden Partikeln diskretisiert. Wie bei dem ersten Ansatz für Masse-Feder Modelle interagiert hier jedes Partikel nur mit seinen vier Nachbarn. Um das Verhalten der Partikel festzulegen, wird eine globale Energiefunktion über alle Partikel definiert. Diese Energiefunktion setzt sich dabei aus der Summe der Energieen der einzelnen Partikel zusammen. Die Endlage des deformierten Partikels entspricht dann dem Minimum dieser Energiefunktion. Der Unterschied zu den Masse-Feder Modellen besteht hierbei nicht nur darin, dass in den Energiefunktionen auch nichtlineare Terme aufgenommen werden können. Ferner wird in diesem Ansatz die Energie nicht nur aus den Entfernungen der jeweiligen Partikel, sondern unter anderem auch aus den in Abbildung 5 zu sehenden Winkeln, in denen diese zueinander stehen, berechnet.

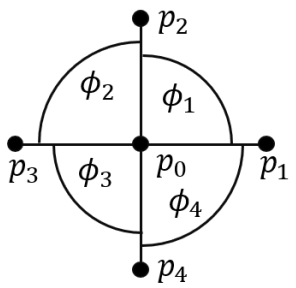


Abbildung 5: Berücksichtigung der Scherwinkel in der Energiegleichung

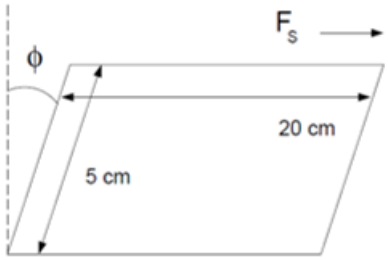
Dem zu Folge kann auch reines Scheren der Partikel, bei dem sich die Entfernun-

gen dieser relativ zueinander nicht ändern, einen Beitrag zur Gesamtenergie leisten. Durch die Betrachtung dieser Energie, kann ein System die Reaktion des Textils auf Scheren abbilden, da es diesem Vorgang zur Energieminimierung entgegenwirkt.
Die Simulation nach [Breen et al., 1994] besteht dabei aus drei Schritten. Im ersten Schritt erfolgt dabei für alle Partikel eine Schätzung der nächsten Position. Als Schätzung wird dabei die Bewegung der Partikel beim Fallen durch ein Fluid unter Gravitation verwendet. Im zweiten Schritt werden die so neu geschätzten Positionen anhand der Energiefunktion angepasst. Dazu wird mit stochastischen Mitteln der Energiegradient des Textils berechnet und jedes Partikel entlang des jeweiligen Gradienten verschoben. Durch die stochastische Komponente bei der Berechnung des Energiegradienten soll das Verbleiben in lokalen Minima vermieden werden. Nachdem ein Minimum der Energiefunktion gefunden wurde, werden im dritten Schritt die Geschwindigkeiten aller Partikel, je nach der in den vorherigen Schritten zurückgelegten Strecke angepasst. In [Eberhardt et al., 1996] wird dieser Ansatz fortgeführt und die Energiefunktion zur effizienteren Berechnung vereinfacht. Die Gesamtenergie des Textils setzt sich hier aus der Summe mehrerer Energiefunktionen der einzelnen Partikel zusammen. Es werden dabei die Gravitationsenergie, Spannungsenergie sowie die Energie aus Scheren und Biegung berücksichtigt. Analog zu [Breen et al., 1994] wird mit diesem Ansatz ebenfalls das Verhalten des Textils im Fallen durch ein Fluid modelliert. Die Energiefunktionen werden dabei so gewählt, dass nichtlineares Materialverhalten abgebildet werden kann. Um realistische Ergebnisse zu erreichen und das System zu stabilisieren, wird zusätzlich die beim Fallen durch ein viskoses Fluid auftretende Reibung berücksichtigt. Im Gegensatz zur in [Breen et al., 1994] vorgestellten Methode wird hier nicht die globale Energiefunktion anhand ihres Gradienten minimiert, sondern die exakte Bewegungsgleichung des Systems aufgestellt und gelöst. Der sich am Ende der Bewegung einstellende Fixpunkt der Bewegung entspricht gleichzeitig auch dem Minimum der globalen Energiefunktion. Die Herleitung der Bewegungsgleichung des Textils erfolgt anhand der Lagrangeschen Mechanik. Dazu wird die globale Energiefunktion in eine Lagrangefunktion überführt und anhand dieser die Lagrangesche Bewegungsgleichung bestimmt.
Des Weiteren wurde in [Breen et al., 1994] eine Methode vorgestellt, um mittels Versuchsdaten eines bestimmten Textils, dieses in der Simulation realitätsgetreu abbilden zu können. Diese basiert auf dem Kawabata Evaluation System welches im Folgenden vorgestellt wird. Damit lässt sich das nichtlineare Verhalten eines bestimmten Textils realitätsgetreu abbilden. Aufgrund der eben genannten Vorteile dieser Modellierung, wurde in dieser Arbeit ein partikelbasierter Ansatz nach [Eberhardt et al., 1996] gewählt.

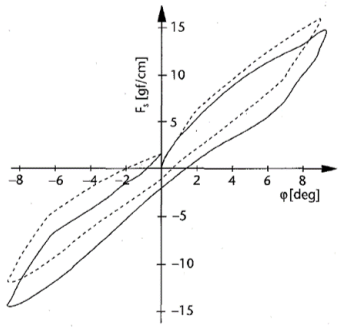
2.3.3 Parametergenerierung aus dem Kawabata Evaluation System

Ein großes Problem bei der Modellierung von Textilien ist es, die richtigen Materialparameter zu setzen, beispielsweise die Steifigkeiten der Federn im Masser-Feder Modell oder die Skalierung der Energiegleichung im Partikelbasierten Modell. Um die Materialparameter eines partikelbasierten Systems optimal an das zu modellierende Textil anzupassen, wurde in [Breen et al., 1994] eine Methode entwickelt,

um diese direkt aus den Testdaten des Kawabata Evaluation Systems (KES) zu generieren.
Das Kawabata Evaluation System besteht aus mehreren Tests, bei denen die Eigenschaften des Textils untersucht werden. Aus den Ergebnissen dieser Tests ergeben sich dann die für die Simulation benötigten Materialparameter. Unter anderem wird dabei der Widerstand des Textil gegen reines Scheren gemessen. Dabei wird der Verlauf der Widerstandskraft des Textils für einen zunehmenden Scherwinkel gemessen. Das Ergebnis dieses Test ist die Kurve des Verlaufs der Scherkraft des Textils in Abhängigkeit des Scherwinkels. Der dafür nötige Versuchsaufbau sowie das resultierende Diagramm der Scherkraft sind in Abbildung 6 dargestellt.



(a) Versuchsaufbau für den Kawabata Scher-Test



(b) Ergebnisse des Kawabata Scher-Tests

Abbildung 6: Aufbau und Auswertung des Kawabata Scher-Tests, Grafiken aus [Eberhardt et al., 1996]

Da sich die Eigenschaften des Textils herstellungsbedingt entlang zweier orthogonaler Richtungen, der sogenannten Kett- und der Schussrichtung stark unterscheiden, erfolgt die Messung der Widerstandskraft entlang beider Richtungen. Durch die Anpassung des Models an diese Ergebnisse kann dann auch das anisotrope Verhalten des Textils abgebildet werden.
Zusätzlich zur Scherkraft wird in einem weiteren Test das Moment bestimmt, welches nötig ist, um das Textil um einen bestimmten Winkel zu biegen. Diese Messung wird jeweils für Kett- und Schussrichtung des Textils durchgeführt. Mithilfe dieser Kurve und dem bereits gezeigten Verlauf der Scherkraft kann später im Modell anhand der auftretenden Winkel die resultierende Scherkraft sowie das Biegemoment berechnet werden. Somit ergibt sich eine Vorschrift zur expliziten Berechnung des nichtlinearen Materialverhaltens des getesteten Textils.

2.3.4 Integration der Bewegungsgleichung

In den meisten obigen Fällen zur Modellierung von Textilien, wurde die hergeleitete Bewegungsgleichung mithilfe eines expliziten Lösers wie zum Beispiel des Runge-Kutta Verfahrens vierter Stufe gelöst. Aufgrund der hohen Steifigkeit der Bewegungsgleichung muss hierbei der Zeitschritt sehr klein gewählt werden, was

den Berechnungsaufwand enorm erhöht. Um diesem Problem entgegenzutreten, wird in [Baraff und Witkin, 1998] eine Semi-Implizite Integrationsvorschrift vorgestellt, mit deren Hilfe der Zeitschritt deutlich höher gewählt werden kann. Die Voraussetzung für die Anwendung dieser Variante ist dabei eine Bewegungsgleichung der Form

$$\ddot{\mathbf{x}} = \frac{\mathbf{f}}{m}, \tag{3}$$

wobei $\mathbf{x}$ für den Positionsvektor des Textils und $\mathbf{f}$ für den Wert der Bewegungsgleichung steht. Diese Bewegungsgleichung kann wie folgt in ein System von Differentialgleichungen erster Ordnung umgeschrieben werden:

$$\dot{\mathbf{x}} = \mathbf{v} \tag{4}$$

$$\dot{\mathbf{v}} = \frac{\mathbf{f}}{m} \tag{5}$$

Hier steht $\mathbf{v}$ für den Geschwindigkeitsvektor des Textils.
Zur Lösung des Systems wird zuerst die normale Form des impliziten Euler Verfahrens aufgestellt:

$$\begin{pmatrix} \mathbf{x}(t_n) \\ \mathbf{v}(t_n) \end{pmatrix} = \begin{pmatrix} \mathbf{x}(t_{n-1}) \\ \mathbf{v}(t_{n-1}) \end{pmatrix} + dt \begin{pmatrix} \mathbf{v}(t_n) \\ \mathbf{f}(t_n)/m \end{pmatrix} \tag{6}$$

Da der auf der rechten Seite auftretende Term $\mathbf{f}(t_n)$ von den zu diesem Zeitpunkt unbekannten Werten $\mathbf{x}(t_n), \mathbf{v}(t_n)$ abhängt, muss in jedem Zeitschritt ein nichtlineares Gleichungssystem gelöst werden, um $\mathbf{x}(t_n), \mathbf{v}(t_n)$ zu berechnen. Da das Lösen dieses teils hochgradig nichtlinearen Gleichungssystems meist sehr zeitaufwändig oder gar unmöglich ist, schlagen die Autoren von [Baraff und Witkin, 1998] die Zuhilfenahme einer Taylorreihenentwicklung des Terms $\mathbf{f}(t_n)$ vor. Mithilfe der Taylorreihenentwicklung erster Ordnung ergibt sich dann eine explizite Bildungsvorschrift für die Werte

$$\Delta\mathbf{x}(t_n) = \mathbf{x}(t_n) - \mathbf{x}(t_{n-1}), \tag{7}$$

$$\Delta\mathbf{v}(t_n) = \mathbf{v}(t_n) - \mathbf{v}(t_{n-1}). \tag{8}$$

Diese Vorschrift hat die Form

$$\mathbf{A}\Delta\mathbf{v}(t_n) = \mathbf{b}, \tag{9}$$

$$\Delta\mathbf{x}(t_n) = c \cdot (\mathbf{v}(t_{n-1}) + \Delta\mathbf{v}(t_n)). \tag{10}$$

Zur Lösung des obigen Systems muss dann nur noch ein lineares Gleichungssystem mit der Systemmatrix $\mathbf{A}$ gelöst werden.
Da diese Integrationsvorschrift auf dem impliziten Eulerverfahren basiert, sinkt zusätzlich der nötige Rechenaufwand pro Zeitschritt, was sich ebenfalls positiv auf die Rechenzeit auswirkt.

3 Modellierung und Lösung

Um das Verhalten eines Textils zu beschreiben, werden aufgrund der genannten Vorteile im Folgenden die Energiegleichung eines patikelbasierten Systems nach [Eberhardt et al., 1996] betrachtet. Dieses Modell beschreibt die Bewegung eines Textils beim Fallen durch ein Fluid. Die Bewegung wird dabei anhand der Positionen und Geschwindigkeiten aller Gitterpunkte beschrieben. Anhand des gewählten Modells für die Energiegleichungen wird in diesem Kapitel in Eigenarbeit die Lagrangesche Bewegungsgleichung hergeleitet und in eine Semi-Implizite Form überführt.

3.1 Diskretisierung des Textils

Die grundlegende Idee der Modellierung ist es, den Vorgang des Fallens eines Textils durch ein Fluid abzubilden. Dazu wird das Textil mithilfe eines kartesischen Gitters diskretisiert, was ein unterschiedliches Materialverhalten entlang der beiden orthogonalen Richtungen ermöglicht. Im Folgenden wird nun die Gesamtenergie des Textils definiert, welche von den Positionen und Geschwindigkeiten der Partikel abhängt. Mithilfe dieser Energie lässt sich dann die Bewegungsgleichung des Systems herleiten. Zur Definition der Gesamtenergie werden die jeweiligen Energien der einzelnen Partikel aufaddiert. Zur Vereinfachung des Modells wird angenommen, dass die Energie eines Partikels nur durch sich selbst und seine direkten Nachbarn beeinflusst wird. Dazu bezeichnen wir im folgenden das Partikel, dessen Energiefunktion wir betrachten als p_0, und seine vier Nachbarn, wie in Abbildung 7 dargestellt, als p_1, p_2, p_3, p_4.

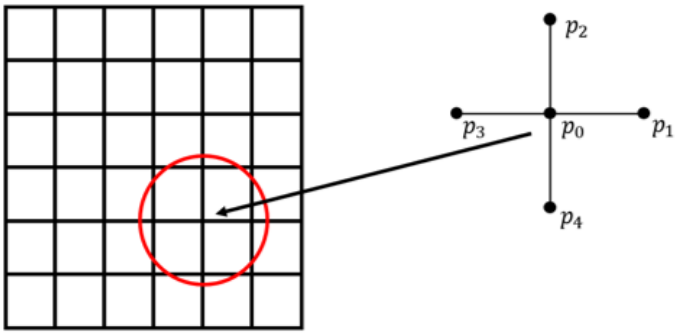


Abbildung 7: Positionen von p_0 und seiner Nachbarn im Gitter

3.1.1 Gesamtenergie des Systems

Die Gesamtenergie des Systems setzt sich nach [Eberhardt et al., 1996] aus den folgenden Energien zusammen:

- Kinetische Energie E^{kin}:

Die kinetische Energie ergibt sich dabei aus der Masse sowie der Geschwindigkeit des jeweiligen Gitterpunktes.

- Potentielle Energie E^{pot}:
  Die potentielle Energie ergibt sich dabei aus der vertikalen Position des Gitterpunktes

- Spannungsenergie E^t:
  Die Spannungsenergie ergibt sich dabei aus den Abständen des Gitterpunkts zu seinen Nachbarn.

- Scherenergie E^s:
  Die Scherenergie ergibt sich dabei aus den Änderungen der Winkel zwischen den jeweilgen 4 Nachbarn des Gitterpunktes.

- Biegungsenergie E^b:
  Die Biegungsenergie ergibt sich dabei aus den Änderungen der Winkel zwischen jeweils 2 gegenüberliegenden Nachbarn des Gitterpunktes.

Durch die hier berücksichtigten Energien leisten die vertikale Position sowie eine Änderung des Winkels zwischen benachbarten Gitterpunkten einen Beitrag zu einer höheren Gesamtenergie. Da ein System immer seine Energieminimierung anstrebt, wird in dem gewählten Modell eine möglichst niedrige vertikale Position sowie die Beibehaltung der ursprünglichen Winkel zwischen den Gitterpunkten angestrebt. Ersteres modelliert dabei ein Fallen des Textils in vertikale Richtung. Zweiteres modelliert die physikalischen Eigenschaften des Textils. Um später die Lagrangesche Bewegungsgleichung

$$\frac{d}{dt}\frac{\partial L}{\partial \mathbf{v}_r} - \frac{\partial L}{\partial \mathbf{x}_r} + R_r = 0 \quad \forall r = 0...n-1 \tag{11}$$

aufzustellen, wird die Lagrangefunktion L als

$$L = E^{kin} - \left(E^{pot} + E^t + E^s + E^b\right) \tag{12}$$

definiert. Dabei wird auch eine Reibungskraft R_r berücksichtigt, welche später die Fallgeschwindigkeit des Textils ausbremst. Damit wird ein Fallen durch ein Fluid mit entsprechend zu wählender Dichte modelliert.
Im Folgenden werden die Beiträge der einzelnen Energiefunktionen zur Bewegungsgleichung bestimmt. Da sich nach [Eberhardt et al., 1996] die gesamte Energie aus der Summe der Energien der einzelnen Teilchen zusammensetzt, wird im Folgenden allgemein die Energie des r-ten Teilchens untersucht. Über die Addition dieser einzelnen Energien ergibt sich dann die Gesamtenergie des Systems.
Es bezeichnet dabei p_0 die Position des r-ten Partikels und p_1, p_2, p_3, p_4 die Positionen der vier Nachbarpartikel.

3.1.2 Kinetische Energie

Die kinetische Energie der r-ten Partikels E_r^{kin} ist gegeben durch

$$E_r^{kin} = \frac{1}{2} m_r \mathbf{v}_r^2 \ , \tag{13}$$

mit m_r der Masse des r-ten Partikels.
Der Beitrag der kinetischen Energie des r-ten Partikels zur Bewegungsgleichung ist somit gegeben durch:

$$\frac{d}{dt}\frac{\partial E_r^{kin}}{\partial \mathbf{v}_r} - \frac{\partial E_r^{kin}}{\partial \mathbf{x}_r} = \frac{d}{dt}\frac{\partial}{\partial \mathbf{v}_r}\left(\frac{1}{2}m_r\mathbf{v}_r^2\right) \tag{14}$$

$$= \frac{d}{dt}\left(m_r\mathbf{v}_r\right) \tag{15}$$

$$= m_r\mathbf{a}_r \tag{16}$$

Hierbei ist $\mathbf{a}_r$ die Beschleunigung des r-ten Partikels.

3.1.3 Potentielle Energie

Die potentielle Energie des r-ten Partikels E_r^{pot} ist gegeben durch

$$E_r^{pot} = m_r g z_r \tag{17}$$

mit g der Erdbeschleunigung und z_r der vertikalen Komponente des Positionsvektors $\mathbf{x}_r$ des r-ten Partikels.
Somit ergibt sich der Beitrag zur Bewegungsgleichung durch:

$$\frac{d}{dt}\frac{\partial(-E_r^{pot})}{\partial \mathbf{v}_r} - \frac{\partial(-E_r^{pot})}{\partial \mathbf{x}_r} = -\frac{\partial}{\partial \mathbf{x}_r}\left(m_r g z_r\right) \tag{18}$$

$$= m_r g \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \tag{19}$$

3.1.4 Spannungsenergie

Nach [Breen et al., 1994] ergibt sich die Energie durch Spannung des r-ten Partikels durch

$$E_r^t = \sum_{j=1}^{4} E_{r,j}^t = \sum_{j=1}^{4} \begin{cases} C_0\left(\frac{(d-r_{0j})^5}{r_{0j}}\right) & r_{0j} \leq d \\ C_0\left(\frac{r_{0j}-d}{d}\right)^5 & r_{0j} > d \end{cases}, \tag{20}$$

wobei C_0 einen Skalierungsfaktor, r_{0j} den aktuellen Abstand zwischen p_0 und dem j-ten Nachbar p_j und d den ursprünglichen Abstand zweier benachbarter Gitterpunkte bezeichnet. Der aktuelle Abstand zweier benachbarter Gitterpunkte $r_{0,j}$ ergibt sich aus

$$r_{0j} = ||p_0 - p_j||, \quad j \in \{1,2,3,4\}. \tag{21}$$

Da die Spannungsenergie des r-ten Teilchens somit nicht nur von der eigenen Position abhängt, sondern auch von der der Nachbarpunkte, liefert diese Energie auch einen Beitrag zur Bewegungsgleichung der Nachbarpunkte.
Die Beiträge zu den Bewegungsgleichungen ergeben sich in Stencil-Notation zu:

$$\frac{-\partial E_r^t}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \frac{\partial(-E_r^t)}{\partial p_2} & 0 \\ \frac{\partial(-E_r^t)}{\partial p_3} & \frac{\partial(-E_r^t)}{\partial p_0} & \frac{\partial(-E_r^t)}{\partial p_1} \\ 0 & \frac{\partial(-E_r^t)}{\partial p_4} & 0 \end{pmatrix} \tag{22}$$

Der Beitrag zur Bewegungsgleichung für p_0 ist dann gegeben durch:

$$-\frac{\partial E_r^t}{\partial p_0} = \sum_{j=1}^{4} -\frac{\partial E_{r,j}^t}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_0} \tag{23}$$

$$= \sum_{j=1}^{4} \begin{cases} -C_0 \left(5\frac{(d-r_{0j})^4}{r_{0j}} + \frac{(d-r_{0j})^5}{r_{0j}^2} \right) \frac{\partial r_{0j}}{\partial p_0} & r_{0j} \leq d \\ \frac{C_0}{d} \left(\frac{(r_{0j}-d)}{d} \right)^4 \frac{\partial r_{0j}}{\partial p_0} & r_{0j} > d \end{cases} \tag{24}$$

mit

$$\frac{\partial r_{0j}}{\partial p_0} = \frac{p_0 - p_j}{||p_0 - p_j||} \tag{25}$$

Da die Anteile der Spannungsenergie $E_{r,j}^t$ wie bereits erwähnt auch von den Positionen der jeweiligen Nachbarn abhängen, ergibt sich für $p_1, ..., p_4$ je der folgende Beitrag zur Bewegungsgleichung:

$$-\frac{\partial E_{r,j}^t}{\partial p_j} = -\frac{\partial E_{r,j}^t}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_j} \tag{26}$$

$$= \begin{cases} -C_0 \left(5\frac{(d-r_{0j})^4}{r_{0j}} + \frac{(d-r_{0j})^5}{r_{0j}^2} \right) \frac{\partial r_{0j}}{\partial p_j} & r_{0j} \leq d \\ \frac{C_0}{d} \left(\frac{(r_{0j}-d)}{d} \right)^4 \frac{\partial r_{0j}}{\partial p_j} & r_{0j} > d \end{cases} \tag{27}$$

mit

$$\frac{\partial r_{0j}}{\partial p_j} = -\frac{p_0 - p_j}{||p_0 - p_j||} \tag{28}$$

3.1.5 Scherenergie

Die Energie durch Scheren des r-ten Partikels ergibt sich nach [Eberhardt et al., 1996] zu

$$E_r^s = \sum_{j=1}^{4} E_{r,j}^s = \sum_{j=1}^{4} \frac{1}{2} C_{s_j} (\phi_j - \frac{\pi}{2})^2 \tag{29}$$

wobei ϕ_j, wie in Abbildung 8 zu sehen, den Winkel zwischen den Vektoren $p_j - p_0$ und $p_{j+1} - p_0$ beschreibt.

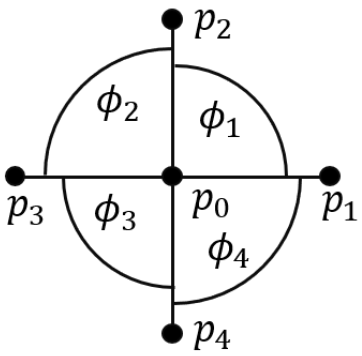


Abbildung 8: Lage der Winkel ϕ_j im Gitter

Dabei wird mit dem Faktor C_j^s die Materialeigenschaft des Textils berücksichtigt. C_j^s ergibt sich aus der Steigung des Verlaufs der Scherkraft im Kawabata Evaluation System (KES) und wird wie in Abbildung 9 gezeigt bestimmt. Dabei ergibt sich der zu verwendende Scherwinkel aus $\phi_j - \frac{\pi}{2}$.

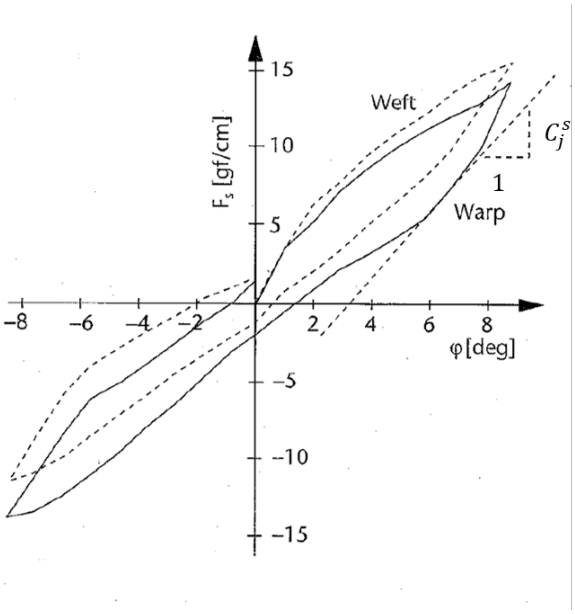


Abbildung 9: Bestimmung von C_{s_j} aus den Ergebnissen des KES, Grafik aus [Eberhardt et al., 1996]

Für den Winkel ϕ zwischen zwei Vektoren $\mathbf{k}_1, \mathbf{k}_2$ gilt:

$$\phi(\mathbf{k_1}, \mathbf{k_2}) = acos\left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right) \tag{30}$$

Somit ergibt sich ϕ_j aus

$$\phi_j = \phi(p_j - p_0, p_{j+1} - p_0). \tag{31}$$

Der Beitrag zur Bewegungsgleichung von p_0 ergibt sich dann zu:

$$-\frac{\partial E_r^s}{\partial p_0} = -\sum_{j=1}^{4} \frac{\partial E_{r,j}^s}{\partial \phi_j} \frac{\partial \phi_j}{\partial p_0} \tag{32}$$

$$= -\sum_{j=1}^{4} C_{s_j}(\phi_j - \frac{\pi}{2}) \frac{\partial \phi_j}{\partial p_0} \tag{33}$$

Dabei wird der Faktor C_j^s nach [Eberhardt et al., 1996] beim Ableiten als konstant angenommen.

Durch Ableiten der Winkelfunktion ϕ ergibt sich

$$\frac{\partial \phi}{\partial \mathbf{k_1}} = \frac{-1}{\sqrt{1-\left(\frac{\mathbf{k_1}\cdot\mathbf{k_2}}{||\mathbf{k_1}||\cdot||\mathbf{k_2}||}\right)^2}}\left(\frac{\mathbf{k_2}}{||\mathbf{k_1}||\cdot||\mathbf{k_2}||} - \frac{(\mathbf{k_1}\cdot\mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3\cdot||\mathbf{k_2}||}\right), \tag{34}$$

$$\frac{\partial \phi}{\partial \mathbf{k_2}} = \frac{-1}{\sqrt{1-\left(\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}\right)^2}}\left(\frac{\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||} - \frac{(\mathbf{k_2}\cdot\mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3\cdot||\mathbf{k_1}||}\right). \tag{35}$$

Es ergibt sich mit $\phi_j = \phi(\underbrace{p_j - p_0}_{\mathbf{k_1}}, \underbrace{p_{j+1} - p_0}_{\mathbf{k_2}})$:

$$\frac{\partial \phi_j}{\partial p_0} = \frac{\partial \phi}{\partial \mathbf{k_1}}\frac{\partial \mathbf{k_1}}{\partial p_0} + \frac{\partial \phi}{\partial \mathbf{k_2}}\frac{\partial \mathbf{k_2}}{\partial p_0} \tag{36}$$

$$= -\frac{\partial \phi}{\partial \mathbf{k_1}} - \frac{\partial \phi}{\partial \mathbf{k_2}} \tag{37}$$

Der Beitrag der Nachbarpunkte ergibt sich zu:

$$-\frac{\partial E^s_{r,j}}{\partial p_j} = C_{s_j}(\phi_j - \frac{\pi}{2})\frac{\partial \phi_j}{\partial p_j} \tag{38}$$

$$-\frac{\partial E^s_{r,j}}{\partial p_{j+1}} = C_{s_j}(\phi_j - \frac{\pi}{2})\frac{\partial \phi_j}{\partial p_{j+1}} \tag{39}$$

$$\tag{40}$$

Hier gilt die Äquivalenz:

$$\frac{\partial \phi_j}{\partial p_j} = \frac{\partial \phi}{\partial \mathbf{k_1}} \tag{41}$$

$$\frac{\partial \phi_j}{\partial p_{j+1}} = \frac{\partial \phi}{\partial \mathbf{k_2}} \tag{42}$$

$$\tag{43}$$

Der gesamte Beitrag der Scherenergie des r-ten Teilchens zur Bewegungsgleichung ergibt sich in Stencil-Notation zu:

$$\begin{pmatrix} 0 & \frac{\partial(-E^s_r)}{\partial p_2} & 0 \\ \frac{\partial(-E^s_r)}{\partial p_3} & \frac{\partial(-E^s_r)}{\partial p_0} & \frac{\partial(-E^s_r)}{\partial p_1} \\ 0 & \frac{\partial(-E^s_r)}{\partial p_4} & 0 \end{pmatrix} \tag{44}$$

3.1.6 Biegungsenergie

Die Energie durch Biegung des r-ten Partikels ergibt sich nach [Eberhardt et al., 1996] zu

$$E^b_r = \sum_{j=1}^{2} E^b_{r,j} = \sum_{j=1}^{2} \frac{1}{2} C^b_j (\Phi_j - \pi)^2 \tag{45}$$

wobei Φ_j,wie in Abbildung 10 zu sehen, den Winkel zwischen den Vektoren $p_j - p_0$ und $p_{j+2} - p_0$ beschreibt.

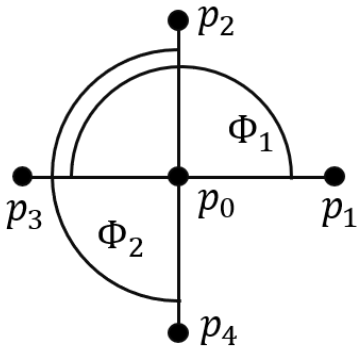


Abbildung 10: Lage der Winkel Φ_j im Gitter

Dabei wird mit dem Faktor C_j^b die Materialeigenschaft des Textils berücksichtigt. C_j^b ergibt sich analog zur Scherenergie aus der Steigung des Verlaufs des Biegemoments im Kawabata Evaluation System. Dabei ergibt sich der zu verwendende Biegewinkel aus $\phi_j - \pi$.
Der Beitrag zur Bewegungsgleichung von p_0 ergibt sich dann zu:

$$-\frac{\partial E_r^b}{\partial p_0} = -\sum_{j=1}^{2} \frac{\partial E_{r,j}^b}{\partial \Phi_j} \frac{\partial \Phi_j}{\partial p_0} \tag{46}$$

$$= -\sum_{j=1}^{2} C_j^b (\Phi_j - \pi) \frac{\partial \Phi_j}{\partial p_0} \tag{47}$$

Analog zur Scherenergie wird der Faktor C_j^b beim Ableiten als konstant angenommen.
Es ergibt sich mit $\Phi_j = \phi(\underbrace{p_j - p_0}_{\mathbf{k_1}}, \underbrace{p_{j+2} - p_0}_{\mathbf{k_2}})$:

$$\frac{\partial \Phi_j}{\partial p_0} = \frac{\partial \phi}{\partial \mathbf{k_1}} \frac{\partial \mathbf{k_1}}{\partial p_0} + \frac{\partial \phi}{\partial \mathbf{k_2}} \frac{\partial \mathbf{k_2}}{\partial p_0} \tag{48}$$

$$= -\frac{\partial \phi}{\partial \mathbf{k_1}} - \frac{\partial \phi}{\partial \mathbf{k_2}} \tag{49}$$

Der Beitrag der Nachbarpunkte ergibt sich zu:

$$-\frac{\partial E_{r,j}^b}{\partial p_j} = C_{b_j} (\Phi_j - \pi) \frac{\partial \Phi_j}{\partial p_j} \tag{50}$$

$$-\frac{\partial E_{r,j}^b}{\partial p_{j+2}} = C_{b_j} (\Phi_j - \pi) \frac{\partial \Phi_j}{\partial p_{j+2}} \tag{51}$$

Hier gilt:

$$\frac{\partial \Phi_j}{\partial p_j} = \frac{\partial \phi}{\partial \mathbf{k_1}} \tag{52}$$

$$\frac{\partial \Phi_j}{\partial p_{j+2}} = \frac{\partial \phi}{\partial \mathbf{k_2}} \tag{53}$$

$$\tag{54}$$

Der gesamte Beitrag der Biegeenergie des r-ten Teilchens zur Bewegungsgleichung ergibt sich in Stencil-Notation zu:

$$\begin{pmatrix} 0 & \frac{\partial(-E_r^b)}{\partial p_2} & 0 \\ \frac{\partial(-E_r^b)}{\partial p_3} & \frac{\partial(-E_r^b)}{\partial p_0} & \frac{\partial(-E_r^b)}{\partial p_1} \\ 0 & \frac{\partial(-E_r^b)}{\partial p_4} & 0 \end{pmatrix} \tag{55}$$

3.1.7 Reibung

Die auftretende Reibung an dem r-ten Gitterpunkt ergibt sich nach [Eberhardt et al., 1996] zu:

$$\mathbf{R}_r = -\frac{1}{2}\rho c_w A_r |\mathbf{v}_r|^2 \frac{\mathbf{v}_r}{|\mathbf{v}_r|} \tag{56}$$

$$= -\frac{1}{2}\rho c_w A_r |\mathbf{v}_r| \mathbf{v}_r \tag{57}$$

Hierbei steht ρ für die Dichte des Fluids, durch welches das Textil fällt, c_w für den Widerstandsbeiwert und A_r für die mit Reibung behaftete Fläche.
Wie bereits erwähnt, kann die Reibung direkt in die Bewegungsgleichung aufgenommen werden.

3.1.8 Bewegungsgleichung

Mithilfe der vorangegangenen Kapitel lässt sich nun die vollständige Bewegungsgleichung für jedes Partikel wie folgt aufstellen:

$$m_r \mathbf{a}_r = -\frac{\partial E^{pot}}{\partial \mathbf{x}_r} - \frac{\partial E^t}{\partial \mathbf{x}_r} - \frac{\partial E^s}{\partial \mathbf{x}_r} - \frac{\partial E^b}{\partial \mathbf{x}_r} + \mathbf{R}_r \tag{58}$$

Hier setzen sich die Gesamtenergien $E^{pot}, E^s, ...$ jeweils aus den Energien der einzelnen Partikel zusammen. Allgemein gilt für jede Energie:

$$E = \sum_{r=0}^{n} E_r \tag{59}$$

Wobei E_r die Energie des r-ten Gitterpunktes darstellt und n die Anzahl aller Gitterpunkte ist.

3.1.9 Umschreiben der Bewegungsgleichung

Um die Differentialgleichung (58) später lösen zu können, wird sie im Folgenden in ein System von Differentialgleichungen erster Ordnung überführt.
Mit

$$\mathbf{a}_r = \ddot{\mathbf{x}}_r = \dot{\mathbf{v}}_r$$

und

$$\mathbf{v}_r = \dot{\mathbf{x}}_r$$

folgt das System von Differentialgleichungen erster Ordnung zu:

$$\dot{\mathbf{x}}_r = \mathbf{v}_r \tag{60}$$

$$\dot{\mathbf{v}}_r = \frac{1}{m_r}\left(-\frac{\partial E^{pot}}{\partial \mathbf{x}_r} - \frac{\partial E^t}{\partial \mathbf{x}_r} - \frac{\partial E^s}{\partial \mathbf{x}_r} - \frac{\partial E^b}{\partial \mathbf{x}_r} + \mathbf{R}_r\right) := \frac{\mathbf{f}_r}{m_r} \quad \forall r = 1, ..., n \tag{61}$$

In Vektor-Schreibweise ergibt sich aus der Zusammenfassung der obigen Gleichungen für alle r:

$$\dot{\mathbf{x}} = \mathbf{v} \tag{62}$$

$$\dot{\mathbf{v}} = \frac{1}{m_0}\left(-\frac{\partial E^{pot}}{\partial \mathbf{x}} - \frac{\partial E^t}{\partial \mathbf{x}} - \frac{\partial E^s_r}{\partial \mathbf{x}} - \frac{\partial E^b}{\partial \mathbf{x}} + \mathbf{R}\right) := \frac{\mathbf{f}}{m_0} \tag{63}$$

3.1.10 Skalierung der Energiegleichungen

Um alle Energiegleichungen und die daraus resultierende Bewegunsgleichung in den selben Einheiten darzustellen, müssen diese entsprechend skaliert werden. Als einheitliches System wird hier das GCS-System gewählt. Für die Energiegleichungen ergeben sich dann die folgenden Skalierungen.

- Potentielle Energie: $[E^{pot}] = \frac{g \cdot m \cdot cm}{s^2}$
  Um diese Energie im CGS-System darzustellen, muss sie mit dem Faktor $\frac{100cm}{m}$ multipliziert werden.

- Spannungsenergie: Da der Parameter für die Spannungsenergie nicht aus Testdaten generiert, sondern an die Ergebnisse angepasst wird, erübrigt sich hier die Skalierung.

- Scherenergie: $[E^s] = \frac{gf}{cm}$ Um diese Energie im CGS-System darzustellen, muss sie mit dem Faktor $\frac{1000N \cdot d}{gf}$ multipliziert werden. Der Faktor d steht dabei für den ursprünglichen Abstand der Gitterpunkte und damit sinnbildlich für die Berücksichtigung der Strecke, auf der die Scherkraft wirkt.

- Biegungsenergie: $[E^b] = gf$ Um diese Energie im CGS-System darzustellen, muss sie mit dem Faktor $\frac{1000N}{gf}$ multipliziert werden.

3.2 Semi-Implizite Lösung des Modells

Aufgrund der sehr hohen Steifigkeit der Differentialgleichung (58) muss der Zeitschritt des Lösers sehr klein gewählt werden. Ebenso muss ein Verfahren mit hoher Konsistenzordnung gewählt werden, um die Konvergenz zu gewährleisten. Nach [Baraff und Witkin, 1998] kann mit einem impliziten Lösungsverfahren ein wesentlich größerer Zeitschritt gewählt werden. Dies motivierte die Herleitung eines Semi-impliziten Lösers für das Modell.

3.2.1 Semi-Implizites Euler-Verfahren

Das explizite Euler-Verfahren zur Lösung des obigen Systems (61) ist wie folgt definiert:

$$\begin{pmatrix} \mathbf{x}(t_n) \\ \mathbf{v}(t_n) \end{pmatrix} = \begin{pmatrix} \mathbf{x}(t_{n-1}) \\ \mathbf{v}(t_{n-1}) \end{pmatrix} + dt \begin{pmatrix} \mathbf{v}(t_{n-1}) \\ \mathbf{f}(t_{n-1})/m \end{pmatrix} \tag{64}$$

Die Vektoren $\mathbf{x}, \mathbf{v}$ enthalten dabei die Positionen beziehungsweise Geschwindigkeiten aller Partikel. Die Reihenfolge der Numerierung der einzelnen Partikel ergibt sich durch das zeilenweise Durchlaufen des diskretisierten Gitters. Die Einträge der Vektoren $\mathbf{v}(t_{n-1})$ und $\mathbf{f}(t_{n-1})$ ergeben sich aus Gleichung (61).
Im Gegensatz dazu ist das implizite Euler-Verfahren definiert als:

$$\begin{pmatrix} \mathbf{x}(t_n) \\ \mathbf{v}(t_n) \end{pmatrix} = \begin{pmatrix} \mathbf{x}(t_{n-1}) \\ \mathbf{v}(t_{n-1}) \end{pmatrix} + dt \begin{pmatrix} \mathbf{v}(t_n) \\ \mathbf{f}(t_n)/m \end{pmatrix} \tag{65}$$

Da der auf der rechten Seite vorkommende Ausdruck $\mathbf{f}(t_n)$ von $\mathbf{x}(t_n)$ und $\mathbf{v}(t_n)$ abhängt, muss in jedem Zeitschritt ein nichtlineares Gleichungssystem gelöst werden. Der Aufwand hierfür ist bei n = Anzahl Gitterpunkte allerdings viel zu groß. Abhilfe schafft hier eine Taylorreihenentwicklung erster Ordnung des Ausdrucks $\mathbf{f}(t_n)(\mathbf{x}(t_n), \mathbf{v}(t_n))$ um $(\mathbf{x}(t_{n-1}), \mathbf{v}(t_{n-1}))$. Es folgt:

$$\mathbf{f}(t_n) = \mathbf{f}(\mathbf{x}(t_n), \mathbf{v}(t_n)) \approx \underbrace{\mathbf{f}(\mathbf{x}(t_{n-1}), \mathbf{v}(t_{n-1}))}_{:=\mathbf{f}(t_{n-1})} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \Delta \mathbf{v} \tag{66}$$

mit

$$\Delta \mathbf{x} = \mathbf{x}(t_n) - \mathbf{x}(t_{n-1}) \tag{67}$$

$$\Delta \mathbf{v} = \mathbf{v}(t_n) - \mathbf{v}(t_{n-1}) \tag{68}$$

Dabei sind $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ Matrizen, die sich aus 3×3 Blöcken zusammensetzen. Die Enträge des i,k-ten Blocks sind dabei durch

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(i,k) = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_k} \tag{69}$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{v}}(i,k) = \frac{\partial \mathbf{f}_i}{\partial \mathbf{v}_k} \tag{70}$$

gegeben. Mithilfe der Definition des impliziten Euler-Verfahrens in (65) folgt für $\Delta\mathbf{x}$ und $\Delta\mathbf{v}$:

$$\Delta\mathbf{x} = \mathbf{x}(t_n) - \mathbf{x}(t_{n-1}) = dt(\mathbf{v}(t_{n-1}) + \Delta\mathbf{v}) \tag{71}$$

$$\Delta\mathbf{v} = \mathbf{v}(t_n) - \mathbf{v}(t_{n-1}) = \frac{dt}{m}\mathbf{f}(t_n) \tag{72}$$

Es ergibt sich also insgesamt für $\Delta\mathbf{v}$ folgender Ausdruck:

$$\Delta\mathbf{v} = \frac{dt}{m}\mathbf{f}(t_n) \approx \frac{dt}{m}\left(\mathbf{f}(t_{n-1}) + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{\partial\mathbf{f}}{\partial\mathbf{v}}\Delta\mathbf{v}\right) \tag{73}$$

$$= \frac{dt}{m}\left(\mathbf{f}(t_{n-1}) + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\left(dt(\mathbf{v}(t_{n-1}) + \Delta\mathbf{v})\right) + \frac{\partial\mathbf{f}}{\partial\mathbf{v}}\Delta\mathbf{v}\right) \tag{74}$$

$$\Leftrightarrow \Delta\mathbf{v} - \frac{dt^2}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Delta\mathbf{v} - \frac{dt}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{v}}\Delta\mathbf{v} = \frac{dt}{m}\left(\mathbf{f}(t_{n-1}) + dt\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathbf{v}(t_{n-1})\right) \tag{75}$$

$$\Leftrightarrow \left(\mathbf{I} - \frac{dt^2}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{x}} - \frac{dt}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{v}}\right)\Delta\mathbf{v} = \frac{dt}{m}\left(\mathbf{f}(t_{n-1}) + dt\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathbf{v}(t_{n-1})\right) \tag{76}$$

$$\Leftrightarrow \Delta\mathbf{v} = \left(\mathbf{I} - \frac{dt^2}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{x}} - \frac{dt}{m}\frac{\partial\mathbf{f}}{\partial\mathbf{v}}\right)^{-1}\frac{dt}{m}\left(\mathbf{f}(t_{n-1}) + dt\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathbf{v}(t_{n-1})\right) \tag{77}$$

Somit existiert nun eine explizite Berechnungsvorschrift für $\Delta\mathbf{v}$. Damit lässt sich $\Delta\mathbf{x}$ leicht zu

$$\Delta\mathbf{x} = dt(\mathbf{v}(t_{n-1}) + \Delta\mathbf{v}) \tag{78}$$

berechnen.
Nach Berechnung der $\Delta\mathbf{x}, \Delta\mathbf{v}$ ergibt sich dann die Bildungsvorschrift für $\mathbf{x}(t_n), \mathbf{v}(t_n)$ zu:

$$\mathbf{x}(t_n) = \mathbf{x}(t_{n-1}) + \Delta\mathbf{x} \tag{79}$$

$$\mathbf{v}(t_n) = \mathbf{v}(t_{n-1}) + \Delta\mathbf{v} \tag{80}$$

Um mit dieser Vorschrift implizit zu rechnen, werden im Folgenden die Terme $\frac{\partial\mathbf{f}}{\partial\mathbf{x}}, \frac{\partial\mathbf{f}}{\partial\mathbf{v}}$ berechnet. Diese Matrizen setzen sich dabei aus 3×3 Blöcken zusammen. Somit müssen die Einträge der Blöcke

$$\frac{\partial\mathbf{f}_i}{\partial\mathbf{x}_k}, \frac{\partial\mathbf{f}_i}{\partial\mathbf{v}_k}$$

für alle Kombinationen aus $0 \leq i, k < n$ berechnet werden. Da die bereits definierten Energien und somit auch die in $\mathbf{f}$ vorkommenden Terme aber immer nur von einem Gitterpunkt und seinen vier Nachbarn abhängen, ergeben sich alle Einträge $\frac{\partial\mathbf{f}_i}{\partial\mathbf{x}_k}, \frac{\partial\mathbf{f}_i}{\partial\mathbf{v}_k}$ zu null, für die gilt:

- $k \neq i$
- Gitterpunkt k ist kein Nachbar von Gitterpunkt i

Somit ergibt sich für eine große Anzahl an Gitterpunkten eine dünn besetzte Matrix.

3.2.2 Potentielle Energie

Für die potentielle Energie gilt

$$\mathbf{f}_{pot} = m_r g \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = const. \tag{81}$$

Somit ergeben sich die Terme $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}, \frac{\partial \mathbf{f}}{\partial \mathbf{v}}$ zu Null.

3.2.3 Spannungsenergie

Die im vorherigen Kapitel 3.1.4 definierte Spannungsenergie des r-ten Partikels E_r^t hängt von der Position des r-ten Partikels, sowie der seiner vier Nachbarn ab. Offensichtlich gilt in diesem Fall $\frac{\partial \mathbf{f}}{\partial \mathbf{v}} = 0$.
Durch die einzelne Betrachtung der Beiträge für alle vier Nachbarn ergeben sich je die zwei folgenden Beiträge zur Bewegungsgleichung

$$\frac{\partial E_{r,j}^t}{\partial p_0}, \frac{\partial E_{r,j}^t}{\partial p_j}. \tag{82}$$

Aus diesen zwei Termen, welche einen Beitrag zum Vektor $\mathbf{f}$ leisten, ergeben sich nach erneutem Ableiten für alle $j = 1, ..., 4$ die folgenden zu berücksichtigenden Kombinationen:

$$\frac{\partial^2 E_{r,j}^t}{\partial p_0^2}, \frac{\partial^2 E_{r,j}^t}{\partial p_0 \partial p_j}, \frac{\partial^2 E_{r,j}^t}{\partial p_j \partial p_0}, \frac{\partial^2 E_{r,j}^t}{\partial p_j^2} \tag{83}$$

Es ergibt sich für $\frac{\partial^2 E_{r,j}^t}{\partial p_0^2}$:

$$\frac{\partial^2 E_{r,j}^t}{\partial p_0^2} = \frac{\partial}{\partial p_0} \frac{\partial E_{r,j}^t}{\partial p_0} \tag{84}$$

$$= \frac{\partial}{\partial p_0} \left(\frac{E_{r,j}^t}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_0} \right) \tag{85}$$

$$= \frac{\partial^2 E_{r,j}^t}{\partial r_{0j}^2} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E_{r,j}^t}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_0^2} \tag{86}$$

Durch Ableiten ergeben sich die noch nicht berechneten Terme zu:

$$\frac{\partial^2 E_{r,j}^t}{\partial r_{0j}^2} = \begin{cases} C_0 \frac{10}{d^2} \left(\frac{r_{0j} - d}{d} \right)^3 & r_{0j} > d \\ C_0 \left(20 \frac{(d - r_{0j})^3}{r_{0j}} + 10 \frac{(d - r_{0j})^4}{r_{0j} 2} + 2 \frac{(d - r_{0j})^4}{r_{0j}^3} \right) & r_{0j} \leq d \end{cases} \tag{87}$$

$$\frac{\partial^2 r_{0j}}{\partial p_0^2} = \frac{1}{r_{0j}} \left(\mathbf{I} - (p_0 - p_j) \otimes (p_0 - p_j) \right) \tag{88}$$

Analog lassen sich sich auch die Terme $\frac{\partial^2 E^t_{r,j}}{\partial p_0 \partial p_j}, \frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_0}, \frac{\partial^2 E^t_{r,j}}{\partial p_j^2}$ berechnen: Wie in Kapitel 7 gezeigt wird, gelten die folgenden Identitäten:

$$\frac{\partial^2 E^t_{r,j}}{\partial p_0 \partial p_j} = -\frac{\partial^2 E^t_{r,j}}{\partial p_0^2} \tag{89}$$

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_0} = -\frac{\partial^2 E^t_{r,j}}{\partial p_0^2} \tag{90}$$

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j^2} = \frac{\partial^2 E^t_{r,j}}{\partial p_0^2} \tag{91}$$

3.2.4 Scherenergie

Die im vorherigen Kapitel 3.1.5 definierte Scherenergie des r-ten Partikels E^s_r hängt von der Position des r-ten Partikels sowie der seiner vier Nachbarn ab. Offensichtlich gilt in diesem Fall $\frac{\partial \mathbf{f}}{\partial \mathbf{v}} = 0$.
Durch die einzelne Betrachtung der Beiträge für alle vier Nachbarn ergeben sich je die drei folgenden Beiträge zur Bewegungsgleichung

$$\frac{\partial E^s_{r,j}}{\partial p_0}, \frac{\partial E^s_{r,j}}{\partial p_j}, \frac{\partial E^s_{r,j}}{\partial p_{j+1}}. \tag{92}$$

Aus diesen drei Termen, welche einen Beitrag zum Vektor $\mathbf{f}$ leisten, ergeben sich nach erneutem Ableiten für alle $j = 1, ..., 4$ die folgenden zu berücksichtigenden Kombinationen:

$$\frac{\partial^2 E^s_{r,j}}{\partial p_0^2}, \frac{\partial^2 E^s_{r,j}}{\partial p_0 \partial p_j}, \frac{\partial^2 E^s_{r,j}}{\partial p_j \partial p_0}, \frac{\partial^2 E^s_{r,j}}{\partial p_j^2}, \frac{\partial^2 E^s_{r,j}}{\partial p_0 \partial p_{j+1}}, \frac{\partial^2 E^s_{r,j}}{\partial p_{j+1} \partial p_0}, \frac{\partial^2 E^s_{r,j}}{\partial p_{j+1}^2}, \frac{\partial^2 E^s_{r,j}}{\partial p_{j+1} \partial p_j}, \frac{\partial^2 E^s_{r,j}}{\partial p_j \partial p_{j+1}} \tag{93}$$

Es ergibt sich für $\frac{\partial^2 E^s_{r,j}}{\partial p_0^2}$:

$$\frac{\partial^2 E^s_{r,j}}{\partial p_0^2} = \frac{\partial}{\partial p_0} \frac{\partial E^s_{r,j}}{\partial p_0} \tag{94}$$

$$= \frac{\partial}{\partial p_0} \left(\frac{\partial E^s_{r,j}}{\partial \phi_j} \frac{\partial \phi_j}{\partial p_0} \right) \tag{95}$$

$$= \frac{\partial^2 E^s_{r,j}}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_0} \otimes \frac{\partial \phi_j}{\partial p_0} + \frac{\partial E^s_{r,j}}{\partial \phi_j} \left(\frac{\partial}{\partial p_0} \left(\frac{\partial \phi_j}{\partial \mathbf{k_1}} \frac{\partial \mathbf{k_1}}{\partial p_0} + \frac{\partial \phi_j}{\partial \mathbf{k_2}} \frac{\partial \mathbf{k_2}}{\partial p_0} \right) \right) \tag{96}$$

$$= \frac{\partial^2 E^s_{r,j}}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_0} \otimes \frac{\partial \phi_j}{\partial p_0} \tag{97}$$

$$+ \frac{\partial E^s_{r,j}}{\partial \phi_j} \left(-\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} \frac{\partial \mathbf{k_1}}{\partial p_0} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \partial \mathbf{k_1}} \frac{\partial \mathbf{k_2}}{\partial p_0} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_1} \partial \mathbf{k_2}} \frac{\partial \mathbf{k_1}}{\partial p_0} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \frac{\partial \mathbf{k_2}}{\partial p_0} \right) \tag{98}$$

$$= \frac{\partial^2 E^s_{r,j}}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_0} \otimes \frac{\partial \phi_j}{\partial p_0} + \frac{\partial E^s_{r,j}}{\partial \phi_j} \left(\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} + \frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \partial \mathbf{k_1}} + \frac{\partial^2 \phi_j}{\partial \mathbf{k_1} \partial \mathbf{k_2}} + \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \right) \tag{99}$$

Somit fehlen zur Berechnung die Komponenten

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_1 k_2}}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_2 k_1}}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2}, \frac{\partial^2 E_r^s}{\partial \phi_j^2}, \tag{100}$$

die in Kapitel 7 bestimmt werden.

Für die restlichen zu berücksichtigenden Terme gilt dann analog:

$$\frac{\partial^2 E_{r,j}^s}{\partial p_0 \partial p_j} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_j} \otimes \frac{\partial \phi_j}{\partial p_0} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(-\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \partial \mathbf{k_1}} \right) \tag{101}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_j \partial p_0} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_0} \otimes \frac{\partial \phi_j}{\partial p_j} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(-\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_1} \partial \mathbf{k_2}} \right) \tag{102}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_0 \partial p_{j+1}} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_{j+1}} \otimes \frac{\partial \phi_j}{\partial p_0} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(-\frac{\partial^2 \phi_j}{\partial \mathbf{k_1} \partial \mathbf{k_2}} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \right) \tag{103}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_{j+1} \partial p_0} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_0} \otimes \frac{\partial \phi_j}{\partial p_{j+1}} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(-\frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \partial \mathbf{k_1}} - \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \right) \tag{104}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_j^2} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_j} \otimes \frac{\partial \phi_j}{\partial p_j} + \frac{\partial E_r^s}{\partial \phi_j} \left(\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} \right) \tag{105}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_{j+1}^2} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_{j+1}} \otimes \frac{\partial \phi_j}{\partial p_{j+1}} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(\frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \right) \tag{106}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_{j+1} \partial p_j} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_j} \otimes \frac{\partial \phi_j}{\partial p_{j+1}} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(\frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \partial \mathbf{k_1}} \right) \tag{107}$$

$$\frac{\partial^2 E_{r,j}^s}{\partial p_j \partial p_{j+1}} = \frac{\partial^2 E_{r,j}^s}{\partial \phi_j^2} \frac{\partial \phi_j}{\partial p_{j+1}} \otimes \frac{\partial \phi_j}{\partial p_j} + \frac{\partial E_{r,j}^s}{\partial \phi_j} \left(\frac{\partial^2 \phi_j}{\partial \mathbf{k_1} \partial \mathbf{k_2}} \right) \tag{108}$$

3.2.5 Biegungsenergie

Die im vorherigen Kapitel definierte Biegungsenergie des r-ten Partikels E_r^b hängt von der Position des r-ten Partikels, sowie der seiner vier Nachbarn ab. Offensichtlich gilt in diesem Fall $\frac{\partial \mathbf{f}}{\partial \mathbf{v}} = 0$.
Durch die einzelne Betrachtung der Beiträge für alle vier Nachbarn ergeben sich je die drei folgenden Beiträge zur Bewegungsgleichung

$$\frac{\partial E_{r,j}^b}{\partial p_0}, \frac{\partial E_{r,j}^b}{\partial p_j}, \frac{\partial E_{r,j}^b}{\partial p_{j+2}}. \tag{109}$$

Aus diesen drei Termen, welche einen Beitrag zum Vektor $\mathbf{f}$ leisten, ergeben sich nach erneutem Ableiten für alle $j = 1, ..., 4$ die folgenden zu berücksichtigenden Kombinationen:

$$\frac{\partial^2 E_{r,j}^b}{\partial p_0^2}, \frac{\partial^2 E_{r,j}^b}{\partial p_0 \partial p_j}, \frac{\partial^2 E_{r,j}^b}{\partial p_j \partial p_0}, \frac{\partial^2 E_{r,j}^b}{\partial p_j^2}, \frac{\partial^2 E_{r,j}^b}{\partial p_0 \partial p_{j+2}}, \frac{\partial^2 E_{r,j}^b}{\partial p_{j+2} \partial p_0}, \frac{\partial^2 E_{r,j}^b}{\partial p_{j+2}^2}, \frac{\partial^2 E_{r,j}^b}{\partial p_{j+2} \partial p_j}, \frac{\partial^2 E_{r,j}^b}{\partial p_j \partial p_{j+2}} \tag{110}$$

Es ergibt sich für $\frac{\partial^2 E^b_{r,j}}{\partial p_0^2}$:

$$\frac{\partial^2 E^b_{r,j}}{\partial p_0^2} = \frac{\partial}{\partial p_0}\frac{\partial E^b_{r,j}}{\partial p_0} \tag{111}$$

$$= \frac{\partial}{\partial p_0}\left(\frac{\partial E^b_{r,j}}{\partial \Phi_j}\frac{\partial \Phi_j}{\partial p_0}\right) \tag{112}$$

$$= \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_0} \otimes \frac{\partial \Phi_j}{\partial p_0} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(\frac{\partial}{\partial p_0}\left(\frac{\partial \Phi_j}{\partial \mathbf{k_1}}\frac{\partial \mathbf{k_1}}{\partial p_0} + \frac{\partial \Phi_j}{\partial \mathbf{k_2}}\frac{\partial \mathbf{k_2}}{\partial p_0}\right)\right) \tag{113}$$

$$= \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_0} \otimes \frac{\partial \Phi_j}{\partial p_0} \tag{114}$$

$$+ \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(-\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2}\frac{\partial \mathbf{k_1}}{\partial p_0} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\partial \mathbf{k_1}}\frac{\partial \mathbf{k_2}}{\partial p_0} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\partial \mathbf{k_2}}\frac{\partial \mathbf{k_1}}{\partial p_0} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}\frac{\partial \mathbf{k_2}}{\partial p_0}\right) \tag{115}$$

$$= \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_0} \otimes \frac{\partial \Phi_j}{\partial p_0} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2} + \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\partial \mathbf{k_1}} + \frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\partial \mathbf{k_2}} + \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}\right) \tag{116}$$

Somit fehlen zur Berechnung die Komponenten

$$\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\mathbf{k_2}}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\mathbf{k_1}}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}, \frac{\partial^2 E^b_r}{\partial \Phi_j^2}, \tag{117}$$

die in Kapitel 7 bestimmt werden.

Für die restlichen zu berücksichtigenden Terme gilt dann analog:

$$\frac{\partial^2 E^b_{r,j}}{\partial p_0 \partial p_j} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_j} \otimes \frac{\partial \Phi_j}{\partial p_0} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(-\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\partial \mathbf{k_1}}\right) \tag{118}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_j \partial p_0} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_0} \otimes \frac{\partial \Phi_j}{\partial p_j} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(-\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\partial \mathbf{k_2}}\right) \tag{119}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_0 \partial p_{j+2}} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_{j+2}} \otimes \frac{\partial \Phi_j}{\partial p_0} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(-\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\partial \mathbf{k_2}} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}\right) \tag{120}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_{j+2} \partial p_0} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_0} \otimes \frac{\partial \Phi_j}{\partial p_{j+2}} + \frac{\partial E^b_{r,j}}{\partial \Phi_j}\left(-\frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\partial \mathbf{k_1}} - \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}\right) \tag{121}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_j^2} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_j} \otimes \frac{\partial \Phi_j}{\partial p_j} + \frac{\partial E^b_r}{\partial \Phi_j}\left(\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2}\right) \tag{122}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_{j+2}^2} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_{j+2}} \otimes \frac{\partial \Phi_j}{\partial p_{j+2}} + \frac{\partial E^b_r}{\partial \Phi_j}\left(\frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}\right) \tag{123}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_{j+2} \partial p_j} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_j} \otimes \frac{\partial \Phi_j}{\partial p_{j+2}} + \frac{\partial E^b_r}{\partial \Phi_j}\left(\frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\partial \mathbf{k_1}}\right) \tag{124}$$

$$\frac{\partial^2 E^b_{r,j}}{\partial p_j \partial p_{j+2}} = \frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2}\frac{\partial \Phi_j}{\partial p_{j+2}} \otimes \frac{\partial \Phi_j}{\partial p_j} + \frac{\partial E^b_r}{\partial \Phi_j}\left(\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\partial \mathbf{k_2}}\right) \tag{125}$$

3.2.6 Reibung

Nach den Annahmen im vorigen Kapitel ergibt sich der Beitrag der Reibung des r-ten Partikels als Reibung beim Fallen durch ein Fluid zu:

$$\mathbf{R}_r = -\frac{1}{2}\rho c_w A_r ||\mathbf{v}_r|| \mathbf{v}_r \tag{126}$$

Es gilt damit offensichtlich $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = 0$.
Da $\mathbf{R}_r$ nur von der Geschwindigkeit $\mathbf{v}_r$ der r-ten Partikels abhängt, liefert auch nur die Ableitung nach $\mathbf{v}_r$ einen Beitrag zu $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$. Dieser ergibt sich zu:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{v}_r} = -\frac{1}{2}\rho c_w A_r \frac{\partial}{\partial \mathbf{v}_r}(||\mathbf{v}_r||) \cdot \mathbf{v}_r - \frac{1}{2}\rho c_w A_r ||\mathbf{v}_r|| \frac{\partial}{\partial \mathbf{v}_r}(\mathbf{v}_r) \tag{127}$$

$$= -\frac{1}{2}\rho c_w A_r \frac{\mathbf{v}_r}{||\mathbf{v}_r||} \otimes \mathbf{v}_r - \frac{1}{2}\rho c_w A_r ||\mathbf{v}_r|| \mathbf{I} \tag{128}$$

3.3 Kollisionsfreiheit

Mit dem bisher hergeleiteten Modell kann das Verhalten eines Textils unter Berücksichtigung seiner physikalischen Eigenschaften modelliert werden. Jedoch ist es dem Textil noch möglich, sich während der Simulation selbst zu durchdringen. Da reale Textilien dies nicht tun, müssen auftretende Kollisionen des Textils mit sich selbst abgefangen werden. Dazu wird im folgenden Kapitel drei verschiedene Algorithmen zur Kollisionsfreiheit des Textils vorgestellt, welche jeweils nach jedem Zeitschritt angewandt werden müssen. In allen drei Versionen wird jeweils die gesamte Fläche des zu simulierenden Textils, wie in Abbildung 11 gezeigt, durch Dreiecksflächen diskretisiert.

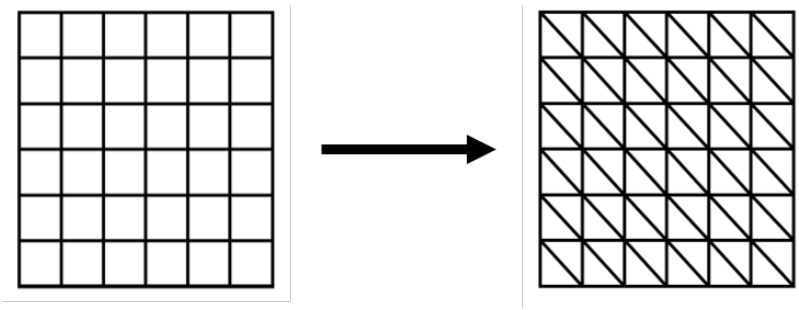


Abbildung 11: Aufteilung des Gitters in Dreiecksflächen

Die einzelnen Dreiecke werden dann auf eine mögliche Selbstdurchdringung des Textils überprüft. Um dabei alle Flächen zu berücksichtigen, werden für jeden Gitterpunkt die in Abbildung 12 gezeigten zwei Dreiecksflächen betrachtet. Aufgespannt werden diese je durch die drei Punkte $(p_0, p_1, p_2), (p_0, p_3, p_4)$. Mit der Iteration über alle Gitterpunkte werden so auch alle Dreiecksflächen und so das gesamte Textil berücksichtigt.

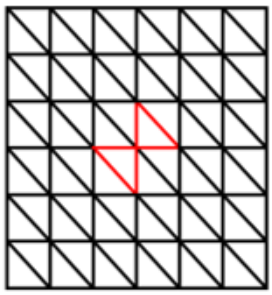


Abbildung 12: Berücksichtigte Flächen für jeden Gitterpunkt

3.3.1 Version 1: Überprüfung auf Durchdringung eines Gitterpunkts

Die erste Version des Algorithmus zur Kollisionsfreiheit entspricht derjenigen aus [Eberhardt et al., 1996]. Es wird hier nach jedem Zeitschritt und für jeden Punkt überprüft, ob dieser auf seinem zurückgelegten Weg eine Dreiecksfläche durchdrungen hat. Dafür wird für jeden Gitterpunkt die Strecke zwischen alter und neuer Position auf einen Schnittpunkt mit einer Dreiecksfläche überprüft.
Im Falle einer Kollision an der Stelle P wird der das Textil durchdringende Gitterpunkt an die Stelle $P' = P + d_{min} \cdot \mathbf{n}$ gesetzt, mit $\mathbf{n}$ der Flächennormalen der geschnittenen Fläche und d_{min} einem gewählten Wert für den minimalem Abstand. Ebenfalls wird die Geschwindigkeit $\mathbf{v}$ des Gitterpunkts angepasst zu $\mathbf{v}' = v_{parallel} + \|v_{normal}\| \cdot \mathbf{n}$, um eine Abstoßung zu modellieren und weitere Kollisionen zu verhindern. Dabei bezeichnet $v_{parallel}$ den zur durchdrungenen Fläche parallelen Anteil der Geschwindigkeit und v_{normal} den zur Fläche orthogonalen Anteil.
Eine durch diese Version abgefangene Kollision ist dabei in Abbildung 13 zu sehen. Dabei ist der die Fläche durchdringende Gitterpunkt je an der alten und neuen Position in rot sowie die durchdrungene Fläche in blau dargestellt. Da die rot gestrichelt eingezeichnete Strecke zwischen der aktuellen und neuen Position des Gitterpunkts eine Dreiecksfläche schneidet, kann diese Kollision abgefangen werden.

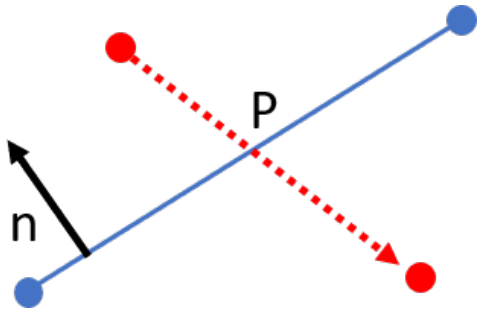


Abbildung 13: Abgefangene Kollision mit Version 1

Jedoch ist dieser Algorithmus nicht in der Lage, alle auftretenden Kollisionen abzu-

fangen. Sollte sich ein Punkt in einem Zeitschritt parallel zu einer Ebene bewegen, ohne diese dabei auf der zurückgelegten Strecke zu durchdringen und sollten die Punkte dieser Ebene ihre Position so ändern, dass sowohl der Punkt zum alten Zeitschritt als auch der zum neuen Zeitschritt auf der anderen Seite der Ebene liegen, so tritt eine Kollision auf, ohne das die Strecke zwischen den Punkten eine Ebene schneidet. Dieses Szenario tritt sehr häufig auf und macht diesen Algorithmus dadurch unbrauchbar. Beispielhaft ist es in Abbildung 14 in den gleichen Farben wie der abgefangene Fall skizziert. Wie zu sehen, bewegt sich der Gitterpunkt nicht normal zur Ebene und die Strecke zwischen der alten und neuen Position schneidet keine der Dreiecksflächen. Da der Gitterpunkt aber nach dem Zeitschritt offensichtlich trotzdem auf der anderen Seite der Ebene liegt, tritt eine Kollision, auf welche mit der bisherigen Version nicht abgefangen werden kann.

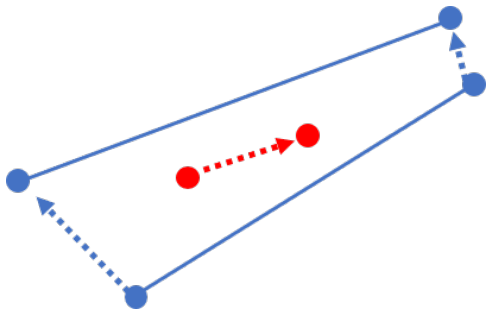


Abbildung 14: Versagensfall der ersten Version der Kollisionsfreiheit

3.3.2 Version 2: Überprüfung auf Durchdringung einer Gitterkante

Aufgrund des Versagens der ersten Version wird im Folgenden ein neu erarbeiteter Algorithmus vorgestellt.
Dieser macht sich zu Nutze, dass keine Kollision auftreten kann, ohne dass eine von dem durchdrigenden Gitterpunkt ausgehende Gitterkante eine Fläche schneidet. Dieser Umstand gilt, da sich die Nachbarn des durchdringenden Gitterpunktes nach der Durchdringung auf der anderen Seite des Textils befinden. So muss die Verbindungskante dieser Gitterpunkte eine Fläche schneiden. Um Kollisionen dieser Art zu finden, werden nach jedem Zeitschritt alle Kanten zwischen benachbarten Gitterpunkten auf Durchdringung einer Fläche überprüft. Im Falle einer Durchdringung wird für beide Gitterpunkte der Kante überprüft, ob sie ihre Orientierung bezüglich der Ebene geändert haben. Sollte dies für einen der Punkte der Fall sein, wird dieser analog zur Version 1 verschoben. Es wird hier der Schnittpunkt P_{neu} der Kante mit der Ebene bestimmt und der das Textil durchdringende Gitterpunkt an die Stelle $P' = P_{neu} + d_{min} \cdot \mathbf{n}$ gesetzt. Ebenfalls wird die Geschwindigkeit $\mathbf{v}$ des Gitterpunkts angepasst zu $\mathbf{v'} = v_{parallel} + ||v_{normal}|| \cdot \mathbf{n}$.
In Abbildung 15 ist dabei eine beispielhafte Selbstdurchdringung des Textils gezeigt, welche durch die Kollisionsfreiheit behoben werden kann. Die Farben wurden dabei analog zur vorherigen Version gewählt. Zusätzlich wurden dabei in schwarz die Kanten zwischen dem aktuell das Textil durchdringenden Gitterpunkt und zwei

seiner Nachbarn eingezeichnet. Da diese Kanten eine Dreiecksfläche schneiden und der in rot dargestellte Gitterpunkt die Orientierung bezüglich der Ebene ändert, kann diese Kollision abgefangen werden.

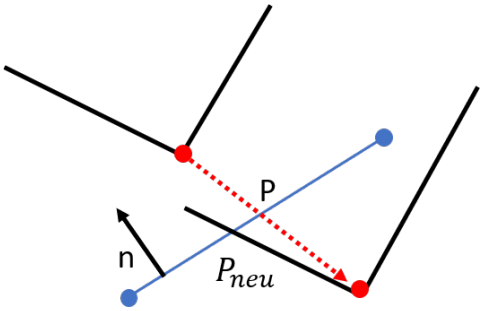


Abbildung 15: Abgefangene Kollision mit Version 2

Auch der Problemfall der vorherigen Version der Kollisionsfreiheit wird mit dieser Methode behoben, da die abgehenden Kanten zu den Nachbarn des durchdringenden Gitterpunktes eine Dreiecksfläche schneiden. Dies ist in Abbildung 16 bildlich dargestellt.

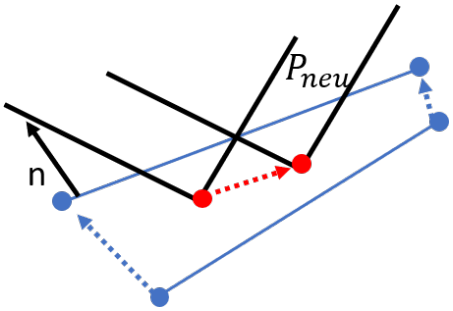


Abbildung 16: Abgefangene Kollision mit Version 2

Jedoch existieren auch für diesen Algorithmus Fälle, in denen eine Kollision nicht abgefangen werden kann. Schneidet nämlich alle Elementkanten eine andere Fläche als jene, die der Gitterpunkt im letzten Zeitschritt durchdrungen hat, so wird diese Kollision nicht behoben. Beispielhaft ist dies in Abbildung 17 skizziert.

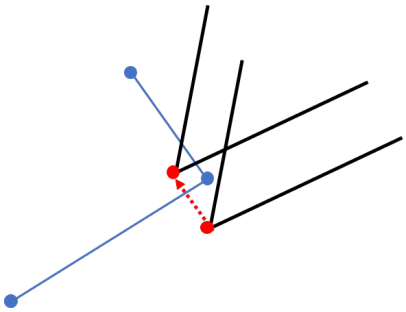


Abbildung 17: Versagensfall der ersten Version

Ein weiteres Problem tritt durch das Wegsetzen eines Punktes um einen festen Abstand auf. Dies kann dazu führen, dass ein Gitterpunkt zur Beseitigung einer bestehenden Kollision durch eine andere Dreiecksfläche hindurch gesetzt wird und so eine neue Kollision verursacht. Ein solcher Fall führt entweder zu nicht behobenen Kollisionen oder bei erneuter Anwendung der Kollisionsfreiheit zur Divergenz der Simulation, da der Punkt unendlich oft durch beide Ebenen hindurch gesetzt wird.
Ebenso kann die Richtung der Normalen für das Wegsetzen des Punktes ungeschickt gewählt sein, da in dieser Richtung ein benachbarter Punkt liegen kann, der den weggesetzten Gitterpunkt im nächsten Schritt stark abstoßen wird. Bei einem hinreichend großen Zeitschritt führt diese große Abstoßung ebenfalls zur Divergenz der Simulation, da der Punkt zu weit weggesetzt wird und das System instabil wird.

3.3.3 Version 3: Erweiterte Überprüfung auf Durchdringung einer Gitterkante

Um auch die Fälle, die die vorherigen Versionen nicht behandeln konnten abfangen zu können, wurde die finale Version des Algorithmus nach zwei Kriterien aufgestellt:

1. Im Falle eines Schnittpunktes einer Kante mit einer Ebene müssen auch benachbarte Flächen auf Durchdringung eines Gitterpunkts der Kante überprüft werden.

2. Im Falle einer Kollision muss für den die Fläche durchdringenden Gitterpunkt eine unproblematische neue Position gefunden werden.

Um diese Kriterien zu erfüllen, wird analog zu Version 2 zuerst wieder nach Schnittpunkten von Kanten zwischen zwei Nachbarn mit einer Dreiecksfläche gesucht. Im Falle eines vorhandenen Schnittpunktes mit einer Fläche, die durch einen Punkt p_0 aufgespannt wird, werden auch für die in Abbildung 18 gezeigten benachbarten Flächen überprüft, ob einer der an der Kante liegenden Gitterpunkte diese Fläche im letzten Zeitschritt durchdrungen hat. Dazu wird für beide Gitterpunkte und jedes berücksichtigte Dreieck überprüft, ob sich für einen der beiden im letzten Zeitschritt die Orientierung zu einer der Ebenen geändert hat. Es sind dabei die von p_0 aufgespannten Flächen, von denen eine von einer Kante geschnitten wird, in rot und die verbleibenden für eine eventuelle Kollision berücksichtigten Flächen in blau dargestellt. So können immer alle Kollisionen identifiziert werden.

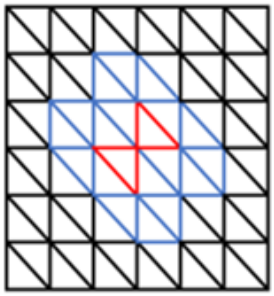


Abbildung 18: Alle berücksichtigten Flächen bei Schnittpunkt im roten Bereich

Um nun nach einer Kollision eine neue zulässige Position für den Gitterpunkt zu finden, wird der Umstand ausgenutzt, dass sich die Punkte im vorherigen Zeitschritt in einer gültigen Konfiguration befanden. Dies ist der Fall, da nach jedem Zeitschritt die Kollisionsfreiheit sichergestellt wird. So wird im Falle einer Kollision der das Textil durchdringende Gitterpunkt auf die Position des letzten Zeitschritts zurückgesetzt. Ebenso werden die Punkte, welche die durchdrungene Ebene aufspannen, jeweils auf die Position des letzten Zeitschritts zurückgesetzt. So ist garantiert, dass der Punkt wieder auf der richtigen Seite der Ebene liegt. Ebenfalls wird die Geschwindigkeit des durchdringenden Gitterpunktes sowie die der Eckpunkte der durchdrungenen Ebene analog zu den vorherigen Versionen zu $\mathbf{v'} = v_{parallel} + \|v_{normal}\| \cdot \mathbf{n}$ angepasst, um eine Abstoßung zu modellieren und weitere Kollisionen zu verhindern. Hierbei wird durch ein unterschiedliches Vorzeichen der Normalen berücksichtigt, dass sich der durchdringende Gitterpunkt und die Ebene in entgegengesetzte Richtung fortbewegen müssen.
Mit diesem Algorithmus werden nun alle auftretenden Kollisionen identifiziert und behoben. Der Vorteil dieses Algorithmus im Vergleich zu den vorhergegangenen Versionen liegt dabei, dass nach einer Kollision zwischen einem Punkt und einer Ebene kein fester minimaler Abstand besteht, sondern sich das Textil beliebig nahe kommen kann. In Abbildung 19 ist hier noch einmal der Versagensfall der vorherigen Version der Kollisionsfreiheit dargestellt. Dieser kann nun ebenfalls behoben werden, da beide in der Abbildung dargestellten Flächen auf Durchdringung überprüft werden und nicht wie in der vorherigen Version nur diejenige Fläche, welche von Kanten geschnitten wird.

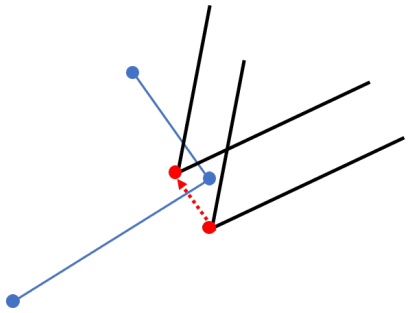


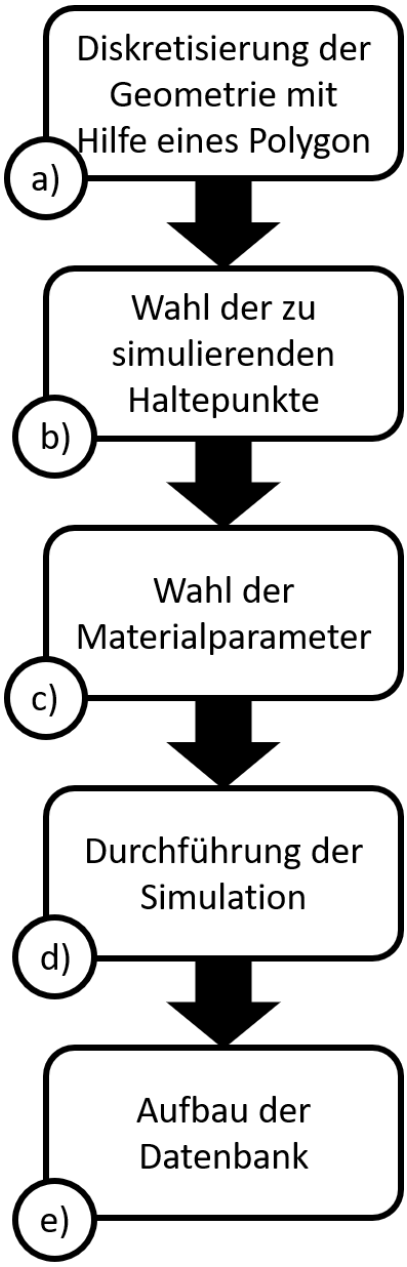
Abbildung 19: Versagensfall der zweiten Version

4 Implementierung

Im folgenden Kapitel wird die Implementierung des Tools für die automatische Bilderkennung von hängenden Textilen vorgestellt. Zu Beginn wird dazu der Ablauf des fertigen Tools erklärt, um anschließend auf Details der Implementierung einzugehen.

4.1 Ablauf des gesamten Tools

Das entwickelte Tool zur automatischen Bilderkennung von hängenden Textilien besteht aus zwei Teilen. Zum einen kann durch die Generierung neuer Simulationsergebnisse von hängenden Textilien eine neue Datenbank aufgebaut oder eine bereits bestehende Datenbank erweitert werden.
Zum anderen kann für ein aufgenommenes Foto eines deformierten Textils das Simulationsergebnis in der Datenbank mit der höchsten Übereinstimmung im Bildervergleich bestimmt werden. Im Folgenden wird der Ablauf beider Teile des Tools erläutert.



Aufnahme eines neuen Textils

a) Als erster Schritt zur Aufnahme eines neuen Textils in das Tool muss dessen Geometrie definiert werden werden. Dazu wird der Umriss des Textils über ein Polygon angenähert und daraus wie in Kapitel 4.7 erklärt die Eingabematrizen der Simulation generiert.

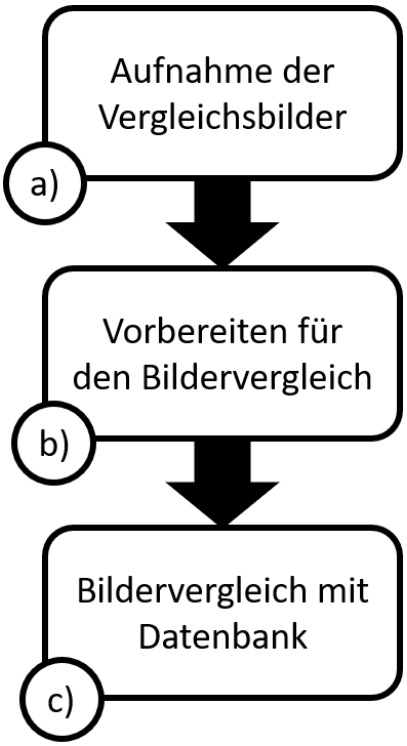
b) Anhand der erzeugten Geometrie werden dann durch den Benutzer die Punkte bestimmt, an welchen das Textil während der Simulation festgehalten werden soll. Es wird dabei für jeden gewählten Greifpunkt eine Simulation durchgeführt und die Endposition der Partikel gespeichert.

c) Anschließend müssen die Materialparameter für das Textil bestimmt werden. Diese können entweder direkt mithilfe des Kawabata Evaluation System (KES) bestimmt werden, oder aus einer Datenbank entnommen werden. Zusätzlich werden in diesem Schritt die Simulationsparameter, wie zum Beispiel die initiale Zeitschrittweite, gesetzt.

d) Mithilfe dieser Input-Daten kann die Simulation für jeden einzelnen Haltepunkt durchgeführt werden.

e) Mit den Simulationsergebnissen für alle Haltepunkte kann dann die Datenbank aufgebaut oder eine bereits bestehende Datenbank erweitert werden.

Mit diesem letzten Schritt ist die Aufnahme eines neuen Textils in die Datenbank abgeschlossen. Die entstandene oder erweiterte Datenbank besteht dann aus den für jede enthaltene Simulation generierten Matrizen mit den gesammelten Positionsdaten der einzelnen Gitterpunkte, sowie den dazugehörigen Geschwindigkeiten und Parametern der Simulation. Mithilfe dieser Daten kann die Lage des Textils beispielsweise über die Matlab Funktion 'surf' visualisiert werden. Der Vorteil bei der Speicherung aller Gitterpositionen und nicht nur einzelner Visualisierungen liegt darin, dass das Textil anschließend aus jedem beliebigen Blickwinkel visualisiert werden kann.



Auswertung für ein fotografiertes Objekt

a) Der erste Schritt um hängende Textilien anhand des Abgleichs mit einer Simulationsdatenbank zu erkennen, ist die Aufnahme von Bildern für den späteren Abgleich.

b) Diese Bilder müssen für den Vergleich wie in Kapitel 4.9 erklärt vorbereitet werden, indem zum Beispiel der Bildhintergrund entfernt wird.

c) Danach kann für ein aufgenommenes Bild eines deformierten Textils die Visualisierung des Simulationsergebnis mit der größten Übereinstimmung im Bildervergleich gefunden werden. Dafür werden von jedem in der Simulationsdatenbank befindlichen Ergebnis mehrere Aufnahmen für unterschiedliche Betrachtungswinkel gemacht und mit dem aufgenommenen Bild des Textils verglichen.

Die Ausgabe ist dann die Simulation mit der höchsten Übereinstimmung im Vergleich. Aus den Parametern dieser Simulation lässt sich für das aufgenommene Bild unter anderem der Punkt bestimmen, an dem es aufgehängt wurde. Außerdem können damit die exakten Positionen interessanter Punkte wie zum Beispiel den Ecken des Textils oder von im Vorhinein festgelegten Greifpunkten zur Weiterverarbeitung des Textil ausgegeben werden.

4.2 Wahl einer geeigneten Programmiersprache

Aufgrund des vielfältigen Funktionsumfangs wurde für erste Testzwecke Matlab als Programmierumgebung gewählt. Um deutliche Einsparungen bei der Laufzeit zu erzielen, wurde im weiteren Verlauf der Implementierung C++ verwendet. Zur Validierung der implementierten Methoden zur Lösung der Differentialgleichung wurden die Ergebnisse des C++ Codes mit denen des Matlab Codes verglichen.

4.3 Zeitschrittverfahren zur Lösung der Differentialgleichung

Das explizite Runge-Kutta Verfahren vierter Ordnung wurde als erster Ansatz zur Lösung des Systems eindimensionaler Differentialgleichungen verwendet. Aufgrund der besseren Stabilitäts- und Konvergenzeigenschaften und der geringeren Anzahl an Funktionsauswertungen wurde im weiteren Verlauf der Arbeit das in Kapitel

3.2 vorgestellte Semi-Implizite Euler Verfahren verwendet.

4.4 Löser für das Lineare Gleichungssystem

Um das in (77) auftretende lineare Gleichungssystem zu lösen, wurde die Funktion 'linlsqrsolvesparse' des C++ Pakets alglib verwendet. Diese Funktion macht sich die dünnbesetzte Struktur der Matrix zunutze. Um die Rechenzeit zur Lösung des linearen Gleichungssystems weiter zu verkürzen, wurde in einem weiteren Schritt die Matrix in (77) reduziert. Dazu wurde der Aufbau der Matrix aus 3×3 Blöcken ausgenutzt. Es werden bei dieser Reduzierung nur die Blöcke auf der Diagonale der Matrix berücksichtigt. Die entstandene reduzierte Matrix kann durch diese Form entkoppelt und auf n einzelne 3×3 Gleichungssystem reduziert werden. Diese werden jeweils mit der expliziten Bildungsvorschrift für die Inverse einer 3×3 Matrix gelöst. Dieses Ansatzes zur Reduzierung des Gleichungssystems wird in Kapitel 5.4.1 validiert.

4.5 Dynamische Zeitschrittweite des Zeitschrittverfahrens

Um sich dem aktuellen Verhalten der Bewegungsgleichung anzupassen, wird die Zeitschrittweite des Verfahrens mit jedem Zeitschritt überwacht und gegebenenfalls angepasst. Hierzu wird die in jedem Zeitschritt zurückgelegte Entfernung aller Gitterpunkte aufsummiert und gespeichert. Wächst diese Norm in einem Zeitschritt um mehr als 20% an, oder wächst sie im Vergleich zu den vorherigen Zeitschritten mehr als sieben mal so schnell an, so wird die Zeitschrittweite verkleinert. Dazu wird die Schrittweite mindestens halbiert. Sollte der Anstieg der Norm im Vergleich zu den Änderungen bei den vorherigen Zeitschritten mindestens um den Faktor sieben steigen, so wird der Zeitschritt zusätzlich durch eben diesen Faktor geteilt.
Sollte die Größe der Norm über 15 Zeitschritte hinweg abnehmen, wird das System als stabil betrachtet, und die Zeitschrittweite um den Faktor 1.5 erhöht.
Die hier beschrieben Faktoren für zum Beispiel den maximalen Anstieg der Norm wurden dabei so gewählt, dass sich in Tests ein möglichst stabiler Verlauf der Zeitschrittweite ergab.

4.6 Parallelisierung der Berechnungen

Der Beitrag der einzelnen Gitterpunkte zur Bewegungsgleichung des Systems hängt nur von den Positionen der jeweiligen Nachbarpunkte ab, welche sich während eines Zeitschritts nicht ändern. Ebenso werden bei der Kollisionsfreiheit auch nur die aktuellen Positionen aller Gitterpunkte berücksichtigt, welche außer im Fall einer Kollision unverändert bleiben. Somit ist die Berechnung des Beitrags zur Bewegungsgleichung für jeden Punkt unabhängig von den Beiträgen der anderen Punkte. Die Reihenfolge der Berechnung spielt also keine Rolle. Insbesondere können auch mehrere Beiträge verschiedener Punkte parallel berechnet werden, ohne dabei das Ergebnis zu beeinflussen. Als Resultat dessen wird die Berechnung der Beiträge der einzelnen Gitterpunkte zur Bewegungsgleichung parallel ausgeführt. Ebenso wird die Kollisionsfreiheit für alle Gitterpunkte parallel sichergestellt. Da

hier allerdings im Falle einer Kollision die Positionen der beteiligten Gitterpunkte verändert werden, muss der Algorithmus zur Kollisionsfreiheit nach jeder Kollision erneut durchgeführt werden, bis keine Kollisionen mehr auftreten.
Zur Parallelisierung der Berechnungen wurde in C++ die Bibliothek Open MP verwendet.

4.7 Speicherung der Gitterpositionen in einer Matrix

Für ein kartesisches Gitter wie in diesem Fall bietet sich eine Matrix an, um die Positionen der Gitterpunkte zu speichern. Diese wird dreidimensional aufgebaut und enthält so an der Stelle i, j den Positionsvektor des i, j-en Gitterpunkts. Die in den vorherigen Kapiteln gewählte Nummerierung der Gitterpunkte durch nur einen Index r ergibt sich dabei durch zweilenweises Durchlaufen der Matrix. Damit ergibt sich der Index r aus den Indizes i, j zu

$$r = i * \sqrt{n} + j, \tag{129}$$

mit n der Gesamtanzahl der Gitterpunkte im quadratischen Gitter.
Diese Matrix wird dabei immer quadratisch gewählt, sodass $n = m^2,\ m \in \mathbb{N}$ gilt. Um nun aber nicht nur quadratische Textilien abbilden zu können, muss zusätzlich die Form eines Textil individuell gespeichert werden. Dazu wird zuerst ein quadratisches Gitter erzeugt, aus dem in einem weiteren Schritt Gitterpunkte entfernt werden können. Dazu wird in der bereits erwähnten Matrix zusätzlich zur Position ein weiterer Wert $k \in \{0, 1\}$ gespeichert. Der Wert $k = 1$ steht für einen berücksichtigten Gitterpunkt an dieser Stelle der Geometrie. Der Wert $k = 0$ hingegen sagt aus, dass der Gitterpunkt an dieser Stelle entfernt beziehungsweise in der Simulation nicht berücksichtigt wird. Das finale Gitter ergibt sich dann aus allen Punkten, für die $k = 1$ gilt.
Um diesen Wert k für alle Punkte der Geometrie zu bestimmen, wird der Umriss des Textils über ein Polygon angenähert. Für das Polygon können beliebig viele Punkte in der zweidimensionalen Ebene gewählt werden. Da Textilien normalerweise eine Achsensymmetrie aufweisen, genügt es mit dem Polygon lediglich eine Hälfte des Umrisses des Textils zu beschreiben. Alle Punkte im Inneren des Polygons werden dann in der Simulation berücksichtigt, die Punkte die außerhalb liegen nicht. Die berücksichtigten Punkte in der anderen Hälfte des Textils ergeben sich dann durch die Aschensymmetrie.
Anhand der Einträge der Matrix kann ein solches Gitter, wie in Abbildung 20 gezeigt, visualisiert werden. Dabei ist links das zur Erzeugung des finalen Gitters verwendete Polygon und rechts das finale Gitter dargestellt.

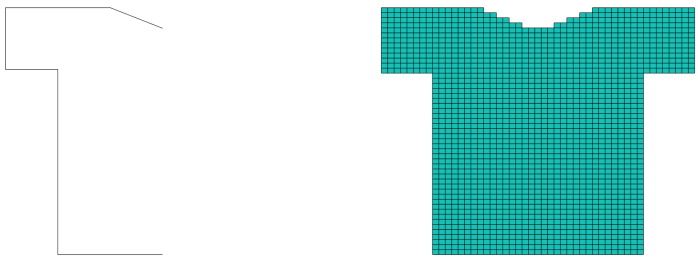


Abbildung 20: Generierung des Gitters aus einem Polygon

Die Geschwindigkeit aller während der Simulation berücksichtigten Gitterpunkte wird analog zu deren Positionen in einer quadratischen Matrix gespeichert.

4.8 Fortsetzen einer begonnenen Simulation

Um eine bereits begonnene Simulation zu deren Endzeitpunkt fortsetzen zu können, werden nach Beendigung der Simulation alle Werte die zur Weiterführung dieser notwendig sind, in csv-Dateien gespeichert. Dazu gehören die Positionen sowie die Geschwindigkeiten aller Partikel, die wie im vorangegangenen Absatz beschrieben in Matrizen gespeichert werden. Zusätzlich werden Simulationsparameter wie der aktuelle Zeitschritt und der Greifpunkt sowie Materialparameter wie die Masse des Textils in einer Parameterdatei gespeichert. Zur Fortsetzung einer Simulation werden diese Daten vom Programm eingelesen und die Simulation kann weitergeführt werden. Um eine Simulation zu starten, werden in Matlab Matrizen mit der initialen Geometrie des Textils sowie der initialen Geschwindigkeit erzeugt und in csv-Dateien gespeichert. Ebenfalls werden die bereits erwähnten Material- und Simulationsparameter gespeichert. Mithilfe dieser Werte kann dann die Simulation gestartet beziehungsweise vom initialen Zeitpunkt aus fortgesetzt werden.

4.9 Bilderkennung mit Matlab

Um eine automatische Bilderkennung von deformierten Textilien zu realisieren, müssen diese Bilder mit den Ergebnissen aus der Simulationsdatenbank abgeglichen werden. Als vereinfachte Version eines Bildervergleichs wird in dieser Arbeit nur die Silhouette des Bildes beziehungsweise der visualisierten Simulationsergebnisse betrachtet. Für den Vergleich werden die Bilder im Voraus so bearbeitet, dass sich der Aufhängepunkt des Textils in der Mitte des oberen Rands des Bildes befindet. Das Bild wird dann quadratisch zugeschnitten, wobei hier als Seitenlänge die Tiefe des hängenden Textils gewählt wird. Anschließend wird der Hintergrund des Bildes komplett weiß eingefärbt. Dieses vorbearbeitete Bild kann in Matlab eingelesen und mit Grauwerten für alle Pixel in einer quadratischen Matrix gespeichert werden. Zwei solcher Matrizen lassen sich dann auf Ähnlichkeit vergleichen. In den Matrizen wird für den Vergleich anhand der Grauwerte für alle Punkte eingetragen ob sie zum Hintergrund gehören oder nicht. Dabei wird für alle Punkte,

die nicht zum Hintergrund gehören der Wert 1 und ansonsten der Wert 0 eingetragen. Visualisieren lässt sich dieses Vorgehen durch ein Einfärben aller nicht zum Hintergrund gehörigen Pixel in schwarz und des Rests in weiß.
Zu sehen ist diese Vorbereitung zur Bilderkennung für Kameradaten in Abbildung 21.

Abbildung 21: Vorbereiten der Bilder auf die Bilderkennung

Analog dazu werden auch die visualisierten Simulationsergebnisse für den Bildvergleich vorbereitet. Dabei werden von jedem visualisierten Ergebnis insgesamt 36 Aufnahmen gemacht. Diese ergeben sich aus den Ansichten des Textils in der vertikalen Ebene jeweils in 10 Grad Schritten.
Für den Vergleich der Bilder mit den Visualisierungen durch die jeweiligen Matrizen wurden zwei Varianten implementiert. Zuerst wurde die Übereinstimmung der beiden Matrizen durch den komponentenweisen Vergleich jedes Eintrags bestimmt und durch die Anzahl an Einträgen der Matrix geteilt. Dieser Wert befindet sich so immer zwischen 0 und 1 und kann als Ähnlichkeitswert der Bilder benutzt werden. Dieser Ähnlichkeitswert enthält allerdings auch die Informationen über die Ähnlichkeit des Bildhintergrunds, da die Übereinstimmung aller Komponenten in den Ähnlichkeitswert mit einfließt.
Das Ziel bei der Entwicklung der zweiten Version war es deshalb, den Hintergrund von dem Vergleich auszuschließen. So sollte der Ähnlichkeitswert zweier Silhouetten bei 0 liegen, wenn sie an keinem Punkt miteinander übereinstimmen. Analog dazu soll dieser Wert bei 1 liegen, wenn die Silhouetten identisch sind. Deshalb wurden in der zweiten Version nur noch Einträge der Matrizen betrachtet, an denen mindestens einer der Punkte schwarz ist, dieser also zu der Silhouette mindestens eines Textils gehört. Für den Ähnlichkeitswert wird dann die Anzahl derjenigen Einträge an denen beide Matrizen übereinstimmen durch die Gesamtanzahl aller berücksichtigten Einträge geteilt. So ergibt sich ein von der Größe des Hintergrunds unverfälschtes Maß dafür, wie ähnlich sich die Bilder sind.

Da die Bilder pixelweise verglichen werden, kann selbst bei Bildern identischer Textilien, von denen eines falsch bearbeitet wurde, nur eine relativ kleine Übereinstimmung erkannt werden. Um dies zu vermeiden und somit Bedienfehler auszuschließen, werden die Bilder mehrmals abgeglichen und dabei die Silhouette eines der Bilder um einige Pixel verschoben. Die besten Werte ergaben sich hier für das

Verschieben der Silhouette in 1-Pixel Schritten um insgesamt je 30 Pixel nach links und rechts und für jeden dieser Fälle ebenfalls um 5 Pixel nach oben und unten. Die damit einhergehende Erhöhung der Laufzeit des Vergleichs um den Faktor 600 wird für die bessere Qualität des Vergleichs in Kauf genommen, da sie mit insgesamt 1 sec. pro Abgleich für eine kleine Datenbank immer noch tragbar ist.

5 Ergebnisse

5.1 Entwicklung der Laufzeit

In den nachfolgenden Abschnitten ist die Entwicklung der Laufzeit für einen Zeitschritt der Simulation beschrieben. Die Zusammenfassung der Entwicklung sowie die jeweiligen Einsparungen zur vorherigen Implementierung sind danach in Tabelle 5.1 dargestellt.

Erste Implementierung in Matlab Die erste Version der Implementierung in Matlab und ohne jegliche Optimierung lag bei circa 47 Sekunden pro Zeitschritt. Das verwendete Zeitschrittverfahren war hierbei das Runge-Kutta-Verfahren vierter Ordnung. Die Zusammensetzung der 47 Sekunden ergibt sich dabei durch die viermalige Berechnung der Lagrange Anteile (je circa 8 Sekunden) und der einmaligen Sicherstellung der Kollisionsfreiheit (circa 15 Sekunden)

Semi-Implizite Implementierung in Matlab Durch die Verwendung des Semi-Impliziten Euler Verfahrens ist nur noch eine einmalige Berechnung der Lagrange Anteile pro Zeitschritt nötig (circa 8 Sekunden). Zusätzlich müssen nun aber die Semi-Impliziten Anteile berechnet werden (circa 8 Sekunden) und ein lineares Gleichungssystem gelöst werden (circa 10 Sekunden). Somit sinkt die Laufzeit pro Zeitschritt auf insgesamt circa 40 Sekunden.

Semi-Implizite Implementierung in C++ Nach der Implementierung des Modells in C++ sinkt die Laufzeit weiter von 40 auf circa 13 Sekunden. Durch die Parallelisierung der Berechnungen sinkt die Laufzeit deutlich auf circa eine Sekunde. Der Großteil der verbleibenden Laufzeit entfällt dabei mit 0.9 Sekunden auf das Lösen des linearen Gleichungssystems in Gleichung 77.

Semi-Implizite Implementierung in C++ mit Reduzierung der Systemmatrix Dadurch, dass der Großteil der verbleibenden Laufzeit des verbleibenden Codes auf die Lösung des LGS entfällt, wurde eine effizientere Lösung dessen motiviert. Ein valider Ansatz dafür ist die in Kapitel 4.4 vorgestellte Reduzierung und Entkoppelung der Systemmatrix. Damit vereinfacht sich die Lösung des Gleichungssystem deutlich und die Laufzeit für einen Zeitschritt sinkt weiter auf 0.06 Sekunden.

Zusammenfassung der Laufzeiteinsparungen In der nachfolgenden Tabelle sind alle im vorherigen Teil des Kapitels erklärten Maßnahmen zur Laufzeiteinsparung noch einmal aufgeführt. Zu jeder Maßnahme ist die Rechenzeit eines Simulationsschrittes sowie die Einsparung in Prozent im Vergleich zur vorherigen Implementierung gegeben.

Stand der Implementierung	Laufzeit in Sek.	Einsparung in %
Matlab, Runge Kutta 4	47	-
Matlab, Semi-Implizit	40	15
C++, Semi-Implizit	13	68
C++, Semi-Implizit, parallel	1	92
C++, Semi-Implizit, parallel, Reduzierung	0.06	94

Insgesamt konnte mit Hilfe aller Maßnahmen circa 99.87% der anfänglichen Laufzeit eingespart werden.

5.2 Effizienz der adaptiven Zeitschrittweitenregelung

Im Folgenden Abschnitt wird die Effizienz der adaptiven Schrittweitenregelung untersucht. Dazu wurde während einer Simulation die jeweilig gewählte Zeitschrittweite für jeden Zeitschritt aufgezeichnet. In Abbildung 22 ist der zeitliche Verlauf der Zeitschrittweite über der Anzahl der Zeitschritte aufgetragen.

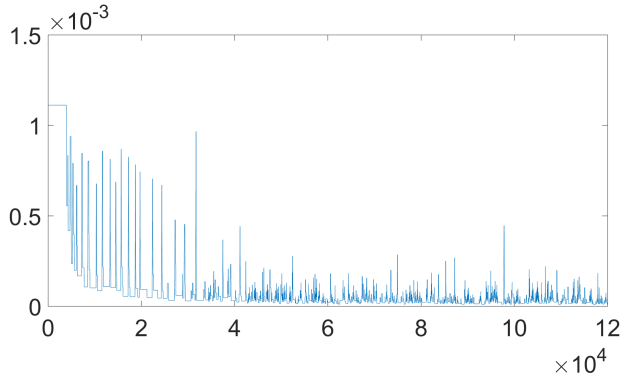


Abbildung 22: Verlauf der Zeitschrittweite in einer Simulation

Wie zu sehen ist, nimmt die Zeitschrittweite im Schnitt bis zum Ende der Simulation immer weiter ab. Auffällig sind die starken Schwankungen in der Zeitschrittweite. Dies lässt sich durch die Einfachheit der adaptiven Regelung erklären. Mit einem Durchschnittswert von $1.0e-4sec.$ liegt der Wert knapp unterhalb des fest gewählten Zeitschitts des Runge-Kutta Verfahrens vierter Ordnung, welcher bei $3.333e-04sec.$ lag. Da jedoch für einen Zeitschritt des semi-impliziten Euler Verfahrens nur eine Funktionsauswertung nötig ist und nicht vier wie bei dem Runge-Kutta Verfahren, ist die Laufzeit des Semi-Impliziten Euler Verfahrens trotzdem geringer.

5.3 Auswertung des Bildervergleichs in Matlab

Im folgenden Abschnitt werden die beiden Ansätze aus Kapitel 4.9 zur Bestimmung der Ähnlichkeit zweier Bilder verglichen, sowie der Effekt der verbesserten Bilderkennung durch die Auswertung mehrerer Bilder mit verschobenen Silhouetten aufgezeigt.
Dazu wird zuerst die Ähnlichkeit zweier Bilder mit den beiden Ansätzen bestimmt. Die dazu verwendeten Silhouetten sind in Abbildung 23 dargestellt.

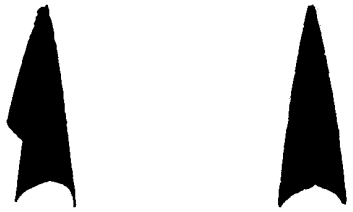


Abbildung 23: Silhouetten der beiden verglichenen Bilder

Für die erste Version des Bildervergleichs, in dem auch der Bildhintergrund berücksichtigt wird, ergibt sich dabei eine Ähnlichkeit von 97% . Obwohl sich die Bilder grundsätzlich ähnlich sind, ist eine Ähnlichkeit von 97 % offensichtlich zu hoch, da im linken Bild eine Falte, die rechts nicht vorhanden ist, sehr markant absteht.
Für die zweite Version der Ähnlichkeitsbestimmung ergibt sich dagegen eine Ähnlichkeit von nur noch 80 %, was der Realität eher entspricht.
Noch deutlicher wird die Unbrauchbarkeit der ersten Version bei einem Bildervergleich der in Abbildung 23 auf der linken Seite dargestellten Silhouette mit einem leeren, also weißem Bild. Für die erste Version des Bildervergleichs ergibt sich hier eine Übereinstimmung von über 80%, die allein auf den Hintergrund zurückzuführen und daher bei der Suche nach einem ähnlichen Bild irreführend ist. Für die zweite Version des Bildervergleichs hingegen ergibt sich eine Ähnlichkeit von 0%, welche auch der Realität entspricht. Somit ist die zweite Version für den Bildervergleich deutlich besser geeignet.

Im zweiten Schritt der Bewertung des Bildervergleichs wird nun der Effekt des Vergleichens zweier Bilder, die zueinander leicht verschoben sind, veranschaulicht. Wie in Kapitel 4.9 erklärt, wird dazu eine der beiden Silhouette schrittweise um mehrere Pixel verschoben, um einen genaueren Bildabgleich zu erreichen. Zum Test der Methode wurden zwei identische Bilder benutzt, wobei bei einem der Bilder der Greifpunkt nicht exakt in der oberen Bildmitte platziert wurde. Beide Silhouetten sind in Abbildung 24 dargestellt. Zur besseren Sichtbarkeit wurde dabei die Bildmitte durch einen vertikalen Balken eingezeichnet.

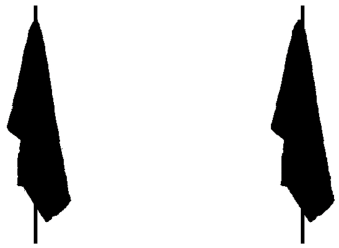
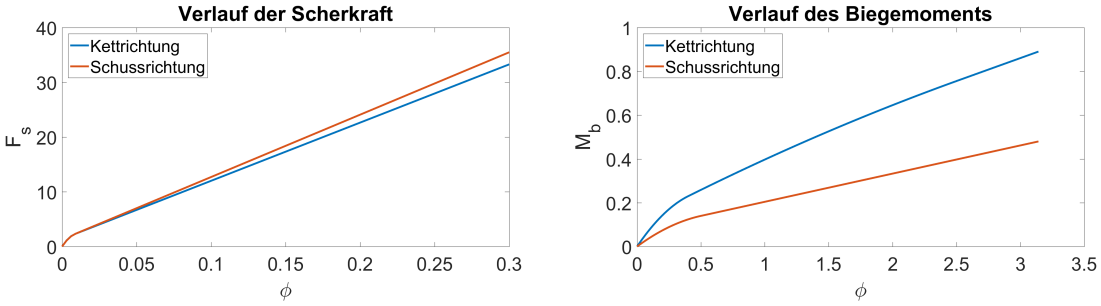


Abbildung 24: Vergleich zweier leicht versetzter aber ansonsten identischer Bilder

Optisch ist zwischen den beiden Bildern kein Unterschied zu erkennen, da eines der beiden nur minimal zur Seite bewegt wurde. Der Vergleich der beiden Silhouetten für diese Lage ergibt aber eine Übereinstimmung von nur 80% . Durch den Vergleich mehrerer Lagen der beiden Bilder ergibt sich eine Übereinstimmung von 100% . Dieses Beispiel verdeutlicht die Wichtigkeit dieses Schrittes beim Bildervergleich mit Matlab.

5.4 Numerische Experimente

Bei den folgenden Experimenten wurden immer die in Abbildung 25 gezeigten Daten aus dem Kawabata Evaluation System Tests von [Breen et al., 1994] für Textilien aus reine Baumwolle benutzt. Die benutzten Interpolationspolynome befinden sich in Kapitel 7.



(a) Verläufe der Scherkraft

(b) Verläufe des Biegemoments

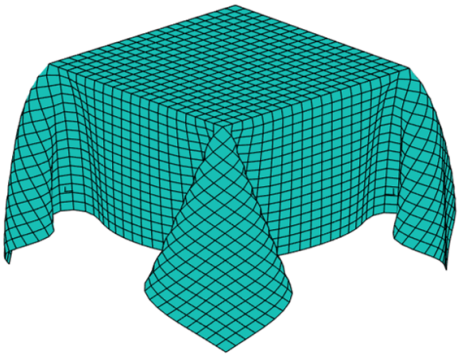
Abbildung 25: Benutzte Daten aus dem KES

5.4.1 Validierung der Simulationsergebnisse

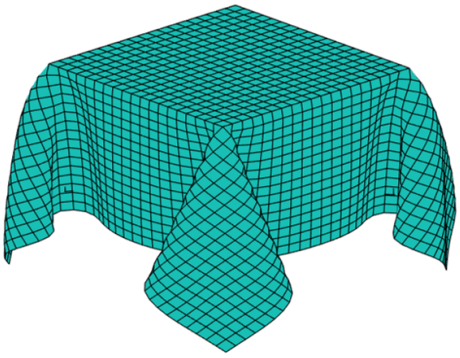
Um zu zeigen, dass mit dem hergeleiteten Modell das Verhalten von Textilien realitätsgetreu abgebildet werden kann, wird im Folgenden ein Validierungsexperiment durchgeführt. Dabei wird die deformierte Lage eines realen Textil sowie die

Visualisierung der Simulationsergebnisse für eben diesen Fall verglichen. Zusätzlich wird das Ergebnis der Simulation für das unveränderte Semi-Imlizite Euler Verfahren mit dem einer Simulation verglichen, in welcher die Matrix des linearen Gleichungssystems reduziert und entkoppelt wird.
Zum Abgleich zwischen Bild- und Simulationsdaten wurde als zu simulierendes Textil ein Baumwolltuch mit den Maßen 100cm×100cm gewählt. Dieses wurde dabei auf einer Fläche von 50cm×50cm festgehalten. Die Positionen der Gitterpunkte innerhalb dieser Fläche wurden fest vorgegeben und während der Simulation nicht verändert. Bildlich entspricht das einem Tischtuch, welches über einen Tisch geworfen wird. In Abbildung 26 sind die Bild- sowie die visualisierten Simulationsdaten für den beschriebenen Testfall dargestellt.

(a) Form des real deformierten Textils



(b) Visualisierung der Simulation ohne Entkoppelung



(c) Visualisierung der Simulation mit Entkoppelung

Abbildung 26: Vergleich der Test und Simulationsergebnisse

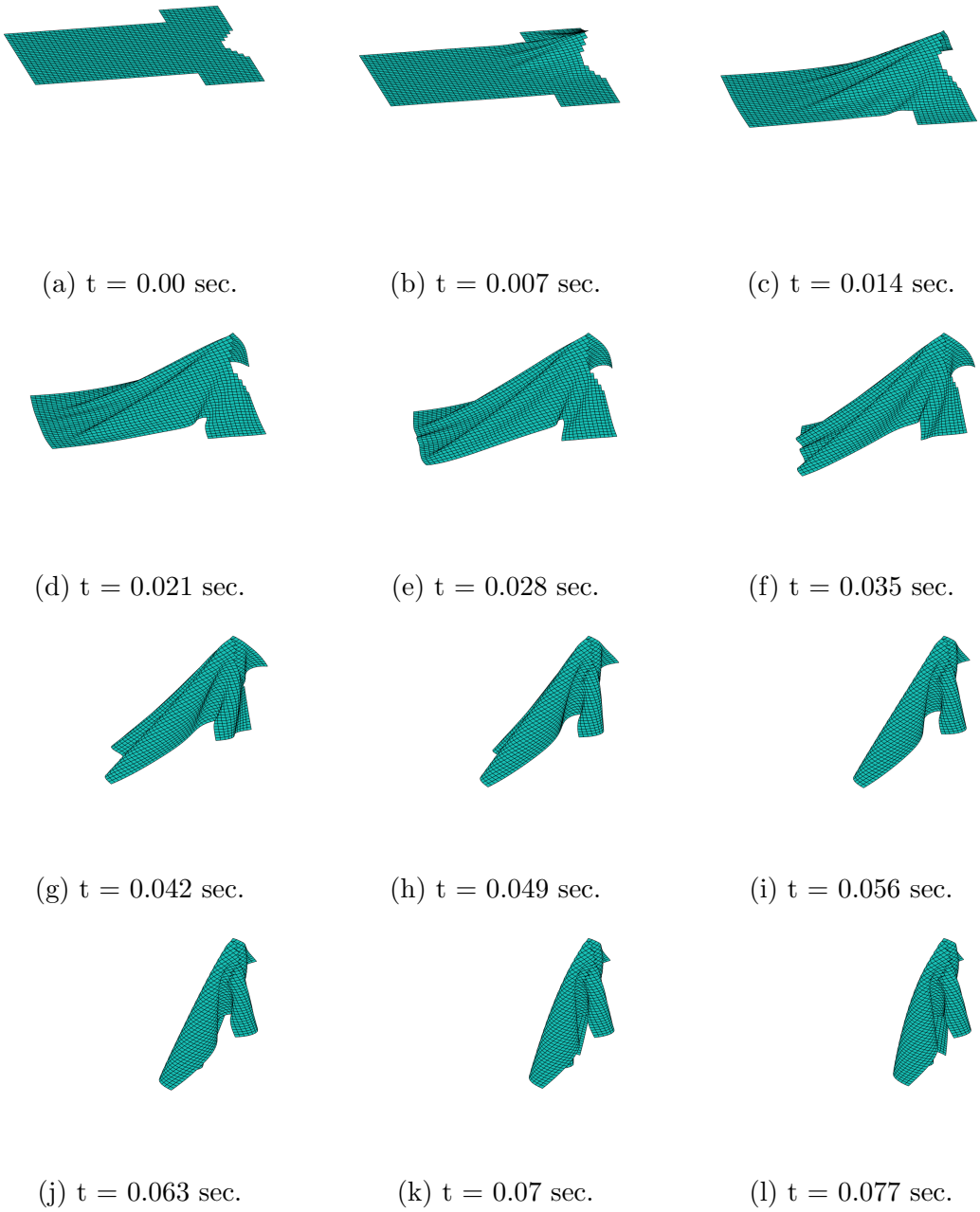
Beim direkten Vergleich des Bildes des realen Textils mit den Visualisierungen der Simulationsergebnisse ist kein offensichtlicher Unterschied festzustellen. Beide Implementierungen können das allgemeine Verhalten des Tuchs gut abbilden. Auch die Form der Falten an den Ecken des Tuchs kann durch das Modell vorhergesagt werden. Ebenfalls ist in beiden Visualisierungen eine große Ausbeulung des Textils entlang der langen unteren Kanten zu sehen, welche auch im Bild des realen Textils zu sehen ist. Für diesen Testfall kann die Realität somit von beiden

Implementierungen gut abgebildet werden. Der durchschnittliche Unterschied der Position zweier Gitterpunkte der beiden Implementierungen liegt bei 2.6 Millimetern. Es ist also kein Qualitätsunterschied zwischen den beiden Implementierungen zu erkennen. So kann die Version mit Entkoppelung der Diagonale ohne Bedenken benutzt werden.

Insgesamt lässt sich somit sagen, dass sowohl die Implementierung des Modells mit entkoppelter Lösung des linearen Gleichungssystems als auch die Implementierung ohne entkoppelte Lösung die Realität sehr gut abbilden. Beide Implementierungen eignen sich somit für eine automatische Bilderkennung. Um dabei Rechenzeit einzusparen, wurde bei den folgenden Ergebnissen immer die Implementierung mit entkoppelter Lösung des linearen Gleichungssystems gewählt.

5.4.2 Speichern der gesamten Bewegung

Da anhand des Modells eine Bewegungsgleichung für das Textil aufgestellt wird, liefert die Simulation nicht nur die Endposition des Textils, sondern den gesamten Bewegungsablauf des fallenden Textils. Dieser ist in Abbildung 27 durch die Visualisierung von 12 Zeitschritten beispielhaft dargestellt. Für dieses Experiment wurde eine T-Shirt ähnliche Geometrie beim Fallen durch ein Textil simuliert, die dabei am Kragen festgehalten wurde.



(a) t = 0.00 sec. (b) t = 0.007 sec. (c) t = 0.014 sec.

(d) t = 0.021 sec. (e) t = 0.028 sec. (f) t = 0.035 sec.

(g) t = 0.042 sec. (h) t = 0.049 sec. (i) t = 0.056 sec.

(j) t = 0.063 sec. (k) t = 0.07 sec. (l) t = 0.077 sec.

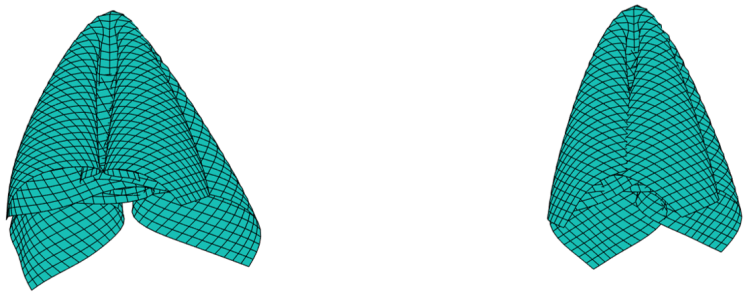
Abbildung 27: Komplette Bewegung eines fallenden Textils

Wie in der obigen Abbildung zu sehen ist, fallen die Punkte am unteren Rand des

T-Shirts bis zum Zeitpunkt 27(c) senkrecht nach unten. Dies lässt sich dadurch erklären, dass die Beschleunigung der Gitterpunkte, bis auf jene durch die Gravitation, nur von den direkten Nachbarn abhängt. So wird die Bewegung der äußeren Punkte erst ab dem Zeitpunkt 27(c) durch das Festhalten der Punkte in der Mitte beeinflusst. Ab diesem Zeitpunkt ist auch die für Textilien typischen Faltenbildung zu sehen. Zunächst bilden sich wie zum Zeitpunkt 27(c) zu sehen viele kleine Falten von der Mitte des Textils aus. Diese erreichen dann zum Zeitpunkt 27(e) auch die äußeren Punkte des Textils. Aus diesen Falten entwickelt sich im Laufe der Simulation eine große Falten vom unteren Ende des T-Shirts bis zum Haltepunkt am Kragen.

5.4.3 Effekt der Kollisionsfreiheit

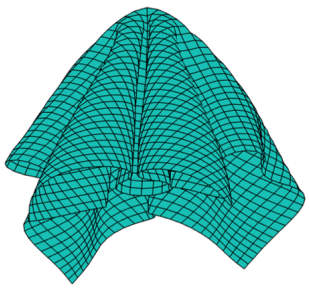
Um die Wichtigkeit der in Kapitel 3.3 hergeleiteten Kollisionsfreiheit zu zeigen, werden in diesem Experiment die Ergebnisse zweier Simulationen verglichen, bei denen nur eine vor Kollisionen geschützt wurde. Bei diesem Experiment wurde das Verhalten eines Baumwolltuchs mit den Abmessungen $30cm \times 30cm$ simuliert, welches an einem Punkt in der Mitte gehalten wurde. Für den ersten Fall wurde das Textil dabei nicht vor Kollisionen geschützt. In Abbildung 28 sind die Lagen des Textils kurz vor sowie nach einer Kollision zu sehen. Dabei ist deutlich zu erkennen, wie sich das Textil in der Mitte selbst durchdringt.



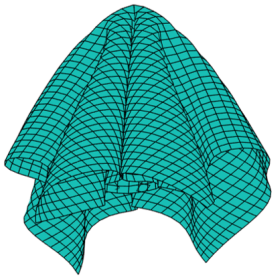
(a) Lage des Textils vor Kollision (b) Lage des Textils nach Kollision

Abbildung 28: Selbstdurchdringung des Textils

Da eine Durchdringung bei realen Textilien nicht auftritt, muss dieses Verhalten, wie in Kapitel 3.3 gezeigt, verhindert werden. In Abbildung 29 sind wieder die deformierten Lagen des Textils kurz vor und nach einer Kollision dargestellt, dieses mal wurde diese aber während der Simulation behoben.



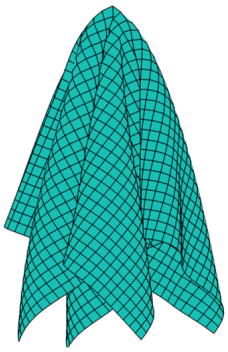
(a) Lage des Textils vor Kollision



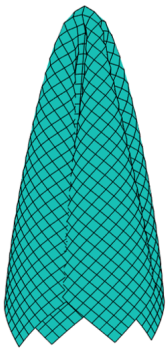
(b) Lage des Textils nach Kollision

Abbildung 29: Behobene Kollision des Textils

In der obigen Abbildung ist gut zu erkennen, wie sich zwei Teile des Textils nach der abgefangenen Kollision berühren, aber nicht durchdringen. Ebenfalls ist ein Unterschied in der Form des gesamten Textils erkennbar, obwohl die Ausgangsdaten sowie die gezeigten Zeitschritte vor und nach der Kollision die selben sind. In Abbildung 30 sind deshalb noch einmal die Endpositionen der beiden Simulationen mit und ohne Kollisionsfreiheit dargestellt.



(a) Endposition des Textils mit Kollisionsfreiheit



(b) Endposition des Textils ohne Kollisionsfreiheit

Abbildung 30: Endpositionen mit und ohne Kollisionsfreiheit

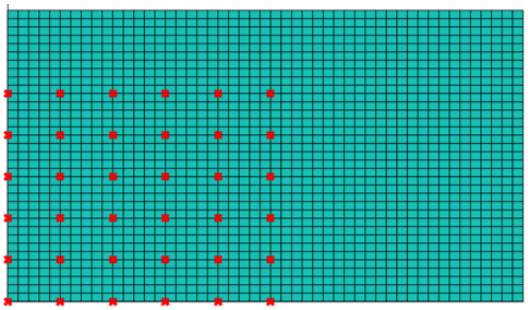
Es ist deutlich zu erkennen, dass die Verhinderung von Selbstdurchdringung des Textils einen großen Unterschied in der Endposition des Textils macht. So kann eine Simulation, welche eine Selbstdurchdringung des Textils zulässt, dieses nicht realitätsgetreu abbilden.

5.4.4 Simulation von verschiedenen Deformationsfällen mit Bilderkennung durch Matlab

Bei diesem Experiment wurden alle Schritte des Tools von der Simulation verschiedener Deformationsfälle, bis zur Bilderkennung über Matlab durchgeführt. Simuliert wurde dabei des Verhalten eines Tuchs mit den Maßen $64cm \times 46cm$ aus

Baumwolle für die Aufhängung an insgesamt 36 Positionen. Danach wurden die entstandenen Ergebnisse mit Bildaufnahmen eines entsprechenden Tuchs verglichen, welches in Abbildung 31 links zu sehen ist.
Aufgrund der Symmetrie des rechteckigen Tuches und der damit einhergehenden Äquivalenz von Simulationsergebnissen mit verschiedenen Haltepunkten wurden in diesem Experiment nur Punkte innerhalb eines Quadranten des Tuchs gewählt. Dabei wurde für jeden in Abbildung 31 rechts rot gekennzeichneten Haltepunkt eine Simulation durchgeführt.

(a) Deformierte Lage des zu erkennenden Textils



(b) Übersicht aller simulierten Haltepunkte

Abbildung 31: Test des Tools

Um anschließend die Übereinstimmung von Simulations- und Testdaten zu überprüfen, wurden die Silhouette der Bildaufnahme wie in Kapitel 4.9 beschrieben mit der gesamten Simulationsdatenbank verglichen. Für jede durchgeführte Simulation, die über den gewählten Haltepunkt identifiziert werden kann, ergibt sich dann ein Wert für die Ähnlichkeit zur Bildaufnahme. Der bei der Bildaufnahme gewählte Haltepunkt liegt in der Mitte der kürzeren Seite des Textils. In Abbildung 32 ist er in rot markiert.

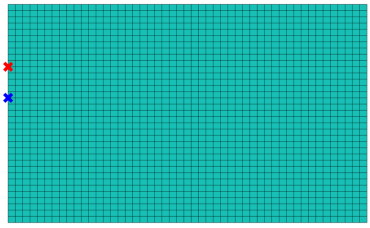
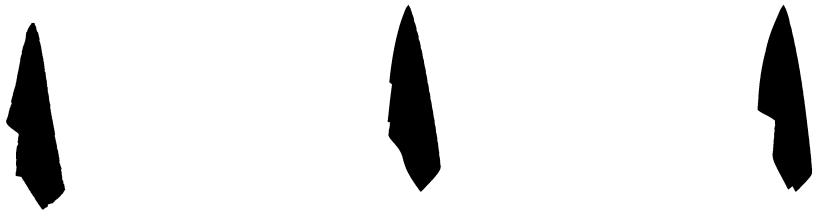


Abbildung 32: Übersicht über den eigentlichen Haltepunkt des Tests (rot) und den Haltepunkt mit der größten Übereinstimmung (blau)

Der Abgleich der Bildaufnahme mit der Simulationsdatenbank liefert eine 87,5 % Übereinstimmung für die Visualisierung des Simulationsergebnis mit dem korrekten Haltepunkte. Eine höhere Übereinstimmung von 88,7 % konnte nur für den in Abbildung 32 in blau dargestellten Haltepunkt erreicht werden. Dies kann dadurch erklärt werden, dass sich beide Deformationsfälle sehr ähnlich sind. Bei den beiden Haltepunkten handelt es sich um direkt benachbarte Haltepunkte am Rand des

Textils. Aufgrund der Einfachheit des Bildvergleichs, bei welchem nur die Silhouette der Bilder verglichen wird, können solch kleine Abweichungen nicht richtig erkannt werden. In der nachfolgenden Abbildung 33 sind die Silhouetten der Bildaufnahme sowie der beiden Visualisierungen mit den höchsten Übereinstimmungen im Bildervergleich dargestellt.



(a) Silhouette des Bildes der deformierten Lage

(b) Silhouette des Simulationsergebnisses für den blauen Haltepunkt

(c) Silhouette des Simulationsergebnisses für den roten Haltepunkt

Abbildung 33: Übersicht über die im Experiment verglichenen Silhouetten

Eine Möglichkeit zur Erhöhung der Genauigkeit der Vorhersage des richtigen Haltepunkts ist der Abgleich mit mehreren Bildern. Dazu muss von dem gesuchten deformierten Textil aus einem anderen Winkel ein zweites Bild aufgenommen und mit der Simulationsdatenbank abgeglichen werden. Werden die Simulationsexperimente zusätzlich mit einem zweiten Bild verglichen und die Ergebnisse für die Ähnlichkeiten beider Fälle kombiniert, so lässt sich in diesem Fall der korrekte Haltepunkt ermitteln. Dazu wurde das in Abbildung 34 gezeigte Bild gewählt.

(a) Deformierte Lage des zweiten zu erkennenden Textils

(b) Silhouette des zweiten zu erkennenden Textils

Abbildung 34: Zweites zu erkennendes Textil

Nach dem Abgleich der zweiten Bildaufnahme mit der Simulationsdatenbank, ergibt sich für die gesuchte Lage ebenfalls nur die zweithöchste Übereinstimmung im Bildvergleich. Für einen anderen Haltepunkt in der Mitte des Textils ergibt sich eine Übereinstimmung von 85,2% während die Visualisierung des roten Haltepunkts nur auf eine Übereinstimmung von 83,7% kommt. Die Silhouetten der beiden Visualisierungen sind in der folgenden Abbildung 35 dargestellt.

(a) Silhouette des Simulationsergebnisses mit der höchsten Übereinstimmung

(b) Silhouette des Simulationsergebnisses für den roten Haltepunkt mit der zweithöchsten Übereinstimmung

Abbildung 35: Übersicht über die im Experiment verglichenen Silhouetten

Durch die Kombination der Ergebnisse aus beiden Bildervergleichen, ergibt sich für die Visualisierung der Simulation mit dem gesuchten roten Haltepunkt die höchste Übereinstimmung von 85%. Die Übereinstimmung der anderen Haltepunkte fällt durch die Kombination auf 84% zurück.
Somit lässt sich mit diesem Tool die Lage eines Textil anhand der Kombination mehrerer Bilddaten erkennen.

6 Zusammenfassung und Ausblick

In den vorausgegangenen Kapiteln wurde zuerst ein partikelbasierter Modellansatz nach [Eberhardt et al., 1996] vorgestellt. Anhand der dort definierten Energiegleichungen wurde mithilfe der Lagrangeschen Betrachtungsweise die Bewegungsgleichung des Systems aufgestellt. Um diese Bewegungsgleichung effizienter zu lösen, wurde anschließend ein Semi-Impliziter Ansatz nach [Baraff und Witkin, 1998] gewählt. Die so berechnete Bewegungsgleichung wurde zur Lösung in ein System von Differentialgleichungen erster Ordnung umgeschrieben. Als nächstes wurde der Ablauf des fertigen Tools zur automatischen Bilderkennung im Gesamten beschrieben. Danach wurde die Implementierung der Lösung der Bewegungsgleichung sowie das zum Abgleich von Kameradaten und Simulationsergebnissen nötige Programm vorgestellt. Abschießend wurden Ergebnisse aus Laufzeit sowie aus numerischen Experimenten vorgestellt. Die Ergebnisse der Experimente bestätigten die realitätsgetreue Abbildung von Textilien durch das Modell, sowie die Fähigkeit der Bilderkennung, anhand von Bilddaten die entsprechenden Simulationsdaten zu finden. Anhand der so erkannten Simulationsdaten kann die genaue Lage des Textils sowie ausgewählter Greifpunkte ausgegeben werden. Somit ergibt sich ein funktionsfähiges und realitätsgetreues Tool zur automatischen Bilderkennung von hängenden Textilien.

In zukünftigen Arbeiten könnte die Funktionsweise und Geschwindigkeit noch weiter verbessert werden. Für eine höhere Genauigkeit könnte dazu der Bildervergleichs angepasst werden, sodass die Bilder während des Vergleichs nicht nur auf ihre Silhouette reduziert werden. Ebenfalls ließe sich der Bildervergleich durch die Implementierung in einer maschinennahen Sprache wie C++ deutlich beschleunigen. Zusätzlich könnte die Effizienz der Suche in den Simulationsdaten gesteigert werden, indem nicht alle Daten abgeglichen werden, sondern mithilfe einer Heuristik gezielt gesucht wird, bis ein hinreichend gutes Ergebnis gefunden wird. Zur weiteren Beschleunigung der Laufzeit des Simulationsprogramms bietet sich die Implementierung eines Semi-Implizten Verfahrens höherer Ordnung an, um die Zeitschritte weiter zu erhöhen. Ebenfalls könnte eine verbesserte Variante der adaptiven Schrittweitenregulierung zu einer weiteren Verminderung der Laufzeit führen.

7 Anhang

Verwendete Daten des KES

Für die durchgeführten Experimente wurden die in [Breen et al., 1994] gemessenen Daten für die Verläufe der Scherkraft und des Biegemoments von Textilien aus Baumwolle verwendet. Zur Berechnung der Größe der Kraft oder des Moments wurde dabei der Betrag des Winkels verwendet, da die Größe der Kraft oder des Moments unabhängig von der Orientierung des Winkels sind. Dafür wurden die folgenden Interpolationspolynome verwendet:

Scherkraft in Kettrichtung

$$F_s(\phi) = \begin{cases} -17993.714\phi^2 + 420.54446\phi & 0 \leq \phi \leq 0.00873 \\ 106.37421\phi + 1.3713532 & \phi > 0.00873 \end{cases} \tag{130}$$

Scherkraft in Schussrichtung

$$F_s(\phi) = \begin{cases} -17125.983\phi^2 + 412.96917\phi & 0 \leq \phi \leq 0.00873 \\ 113.94950\phi + 1.3052208 & \phi > 0.00873 \end{cases} \tag{131}$$

Biegemoment in Kettrichtung

$$M_b(\phi) = \begin{cases} -0.712212123\phi^2 + 0.85984849\phi & 0 \leq \phi \leq 0.4 \\ -0.018939394^2 + 0.30530303\phi + 0.11090909 & 0.4 < \phi \leq 2.52 \\ 0.20984848\phi + 0.23118183 & \phi > 2.52 \end{cases} \tag{132}$$

Biegemoment in Schussrichtung

$$M_b(\phi) = \begin{cases} -0.30257426\phi^2 + 0.43128713\phi & 0 \leq \phi \leq 0.5 \\ 0.12871287\phi + 0.075643564 & \phi > 0.5 \end{cases} \tag{133}$$

Implizite Version Spannungsenergie

Identität der Gleichung 91 a) Es ergibt sich für $\frac{\partial^2 E^t_{r,j}}{\partial p_0 \partial p_j}$

$$\frac{\partial^2 E^t_{r,j}}{\partial p_0 \partial p_j} = \frac{\partial}{\partial p_0} \frac{\partial E^t_{r,j}}{\partial p_j} \tag{134}$$

$$= \frac{\partial}{\partial p_0} \left(\frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_j} \right) \tag{135}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_j} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_0 \partial p_j} \tag{136}$$

$$\tag{137}$$

Durch Ableiten ergeben sich die noch nicht berechneten Terme zu:

$$\frac{\partial^2 r_{0j}}{\partial p_0 \partial p_j} = \frac{1}{r_{0j}} \left(-\mathbf{I} + (p_0 - p_j) \otimes (p_0 - p_j)\right) \tag{138}$$

$$= -\frac{\partial^2 r_{0j}}{\partial p_0^2} \tag{139}$$

Aufgrund der Identität

$$\frac{\partial r_{0j}}{\partial p_0} = \frac{p_0 - p_j}{||p_0 - p_j||} = -\left(-\frac{p_0 - p_j}{||p_0 - p_j||}\right) = -\frac{\partial r_{0j}}{\partial p_j} \tag{140}$$

gilt dann

$$\frac{\partial^2 E^t_{r,j}}{\partial p_0 \partial p_j} = \frac{\partial^2 E^t_{r,j}}{\partial r_{0j}^2} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_j} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_0 \partial p_j} \tag{141}$$

$$= -\frac{\partial^2 E^t_{r,j}}{\partial r_{0j}^2} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} - \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_0^2} \tag{142}$$

$$= -\left(\frac{\partial^2 E^t_{r,j}}{\partial r_{0j}^2} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_0^2}\right) \tag{143}$$

$$= -\frac{\partial^2 E^t_{r,j}}{\partial p_0^2}. \tag{144}$$

Identität der Gleichung 91 b) : Analog ergibt sich für $\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_0}$

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_0} = \frac{\partial}{\partial p_j} \frac{\partial E^t_{r,j}}{\partial p_0} \tag{145}$$

$$= \frac{\partial}{\partial p_j} \left(\frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_0}\right) \tag{146}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial r_{0j}^2} \frac{\partial r_{0j}}{\partial p_j} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_j \partial p_0} \tag{147}$$

$$\tag{148}$$

Durch Ableiten ergeben sich die noch nicht berechneten Terme zu:

$$\frac{\partial^2 r_{0j}}{\partial p_j \partial p_0} = \frac{1}{r_{0j}} \left(-\mathbf{I} + (p_0 - p_j) \otimes (p_0 - p_j)\right) \tag{149}$$

$$= -\frac{\partial^2 r_{0j}}{\partial p_0^2} \tag{150}$$

Aufgrund der Identität bereits gezeigten Identität $\frac{\partial r_{0j}}{\partial p_0} = -\frac{\partial r_{0j}}{\partial p_j}$ gilt:

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_0} = \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_j} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_j \partial p_0} \tag{151}$$

$$= -\frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} - \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p^2_0} \tag{152}$$

$$= -\left(\frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p^2_0} \right) \tag{153}$$

$$= -\frac{\partial^2 E^t_{r,j}}{\partial p^2_0} \tag{154}$$

Identität der Gleichung 91 c) :

Es ergibt sich für $\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_j}$:

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_j} = \frac{\partial}{\partial p_j} \frac{\partial E^t_{r,j}}{\partial p_j} \tag{155}$$

$$= \frac{\partial}{\partial p_j} \left(\frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial r_{0j}}{\partial p_j} \right) \tag{156}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_j} \otimes \frac{\partial r_{0j}}{\partial p_j} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_j \partial p_j} \tag{157}$$

$$\tag{158}$$

Durch Ableiten ergeben sich die noch nicht berechneten Terme zu:

$$\frac{\partial^2 r_{0j}}{\partial p_j \partial p_j} = \frac{1}{r_{0j}} \left(\mathbf{I} - (p_0 - p_j) \otimes (p_0 - p_j) \right) \tag{159}$$

$$= \frac{\partial^2 r_{0j}}{\partial p^2_0} \tag{160}$$

Aufgrund der Identität bereits gezeigten Identität $\frac{\partial r_{0j}}{\partial p_0} = -\frac{\partial r_{0j}}{\partial p_j}$ gilt:

$$\frac{\partial^2 E^t_{r,j}}{\partial p_j \partial p_j} = \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_j} \otimes \frac{\partial r_{0j}}{\partial p_j} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p_j \partial p_j} \tag{161}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \left(-\frac{\partial r_{0j}}{\partial p_0} \right) \otimes \left(-\frac{\partial r_{0j}}{\partial p_0} \right) + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p^2_0} \tag{162}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial r^2_{0j}} \frac{\partial r_{0j}}{\partial p_0} \otimes \frac{\partial r_{0j}}{\partial p_0} + \frac{E^t_{r,j}}{\partial r_{0j}} \frac{\partial^2 r_{0j}}{\partial p^2_0} \tag{163}$$

$$= \frac{\partial^2 E^t_{r,j}}{\partial p^2_0} \tag{164}$$

Implizite Version Scherenergie

Mithilfe von

$$\phi_j = \phi(\underbrace{p_j - p_0}_{\mathbf{k_1}}, \underbrace{p_{j+1} - p_0}_{\mathbf{k_2}}) \tag{165}$$

$$\frac{\partial \phi_j}{\partial \mathbf{k_1}} = \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right)^2}} \left(\frac{\mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} - \frac{(\mathbf{k_1} \cdot \mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \right) \tag{166}$$

$$\frac{\partial \phi_j}{\partial \mathbf{k_2}} = \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_2} \cdot \mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||}\right)^2}} \left(\frac{\mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||} - \frac{(\mathbf{k_2} \cdot \mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} \right) \tag{167}$$

$$\frac{\partial E^s_{r,j}}{\partial \phi_j} = C^s_j(\phi_j - \frac{\pi}{2}) \tag{168}$$

ergibt sich:

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2} = \frac{\partial}{\partial \mathbf{k_1}} \frac{\partial \phi_j}{\partial \mathbf{k_1}} \tag{169}$$

$$= \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right)^2}} \tag{170}$$

$$\cdot \left(-\frac{\mathbf{k_2} \otimes \mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} - \frac{\mathbf{k_1} \otimes \mathbf{k_2}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} - \frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \mathbf{I} + 3 \frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||^3} \mathbf{k_1} \otimes \mathbf{k_2} \right) \tag{171}$$

$$- \frac{\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}}{\left(1 - \left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right)^2\right)^{3/2}} \tag{172}$$

$$\cdot \left(\frac{\mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} - \frac{(\mathbf{k_1} \cdot \mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \right) \otimes \left(\frac{\mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} - \frac{(\mathbf{k_1} \cdot \mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \right) \tag{173}$$

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_2} \mathbf{k_1}} = \frac{\partial}{\partial \mathbf{k_2}} \frac{\partial \phi_j}{\partial \mathbf{k_1}} \tag{174}$$

$$= \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right)^2}} \tag{175}$$

$$\cdot \left(\frac{1}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} \mathbf{I} - \frac{\mathbf{k_2} \otimes \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||^3} - \frac{\mathbf{k_1} \otimes \mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} + \frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||^3} \mathbf{k_1} \otimes \mathbf{k_2} \right) \tag{176}$$

$$- \frac{\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}}{\left(1 - \left(\frac{\mathbf{k_1} \cdot \mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||}\right)^2\right)^{3/2}} \tag{177}$$

$$\cdot \left(\frac{\mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} - \frac{(\mathbf{k_1} \cdot \mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \right) \otimes \left(\frac{\mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||} - \frac{(\mathbf{k_2} \cdot \mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} \right) \tag{178}$$

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}\mathbf{k_2}} = \frac{\partial}{\partial \mathbf{k_1}} \frac{\partial \phi_j}{\partial \mathbf{k_2}} \tag{179}$$

$$= \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}\right)^2}} \tag{180}$$

$$\cdot \left(\frac{1}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||}\mathbf{I} - \frac{\mathbf{k_1} \otimes \mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||^3} - \frac{\mathbf{k_2} \otimes \mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} + \frac{\mathbf{k_2} \cdot \mathbf{k_1}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||^3}\mathbf{k_2} \otimes \mathbf{k_1} \right) \tag{181}$$

$$- \frac{\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}}{\left(1 - \left(\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}\right)^2\right)^{3/2}} \tag{182}$$

$$\cdot \left(\frac{\mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||} - \frac{(\mathbf{k_2} \cdot \mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} \right) \otimes \left(\frac{\mathbf{k_2}}{||\mathbf{k_1}|| \cdot ||\mathbf{k_2}||} - \frac{(\mathbf{k_1} \cdot \mathbf{k_2})\mathbf{k_1}}{||\mathbf{k_1}||^3 \cdot ||\mathbf{k_2}||} \right) \tag{183}$$

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} = \frac{\partial}{\partial \mathbf{k_2}} \frac{\partial \phi_j}{\partial \mathbf{k_2}} \tag{184}$$

$$= \frac{-1}{\sqrt{1 - \left(\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}\right)^2}} \tag{185}$$

$$\cdot \left(-\frac{\mathbf{k_1} \otimes \mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} - \frac{\mathbf{k_2} \otimes \mathbf{k_1}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} - \frac{\mathbf{k_2} \cdot \mathbf{k_1}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||}\mathbf{I} + 3\frac{\mathbf{k_2} \cdot \mathbf{k_1}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||^3}\mathbf{k_2} \otimes \mathbf{k_1} \right) \tag{186}$$

$$- \frac{\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}}{\left(1 - \left(\frac{\mathbf{k_2}\cdot\mathbf{k_1}}{||\mathbf{k_2}||\cdot||\mathbf{k_1}||}\right)^2\right)^{3/2}} \tag{187}$$

$$\cdot \left(\frac{\mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||} - \frac{(\mathbf{k_2} \cdot \mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} \right) \otimes \left(\frac{\mathbf{k_1}}{||\mathbf{k_2}|| \cdot ||\mathbf{k_1}||} - \frac{(\mathbf{k_2} \cdot \mathbf{k_1})\mathbf{k_2}}{||\mathbf{k_2}||^3 \cdot ||\mathbf{k_1}||} \right) \tag{188}$$

$$\frac{\partial^2 E^s_{r,j}}{\partial \phi_j^2} = \frac{\partial}{\partial \phi_j} \frac{\partial E^s_{r,j}}{\partial \phi_j} \tag{189}$$

$$= C^s_j + \frac{\partial}{\partial \phi_j}(C^s_j)(\phi_j - \frac{\pi}{2}) \tag{190}$$

Implizite Version Biegeenergie

Die Beiträge

$$\frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}^2}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_1}\mathbf{k_2}}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}\mathbf{k_1}}, \frac{\partial^2 \Phi_j}{\partial \mathbf{k_2}^2}$$

ergeben sich aus den im vorherigen Abschnitt hergeleiteten Beiträgen

$$\frac{\partial^2 \phi_j}{\partial \mathbf{k_1}^2}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_1}\mathbf{k_2}}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}\mathbf{k_1}}, \frac{\partial^2 \phi_j}{\partial \mathbf{k_2}^2} \tag{191}$$

durch Einsetzen von $\mathbf{k}_1 = p_j - p_0, \mathbf{k}_2 = p_{j+2} - p_0$, da

$$\Phi_j = \phi(\underbrace{p_j - p_0}_{\mathbf{k_1}}, \underbrace{p_{j+2} - p_0}_{\mathbf{k_2}}) \tag{192}$$

gilt und somit ϕ_j und Φ_j die selbe Winkelfunkion zugrunde liegt.
Der noch fehlende Term ergibt sich zu:

$$\frac{\partial^2 E^b_{r,j}}{\partial \Phi_j^2} = \frac{\partial}{\partial \Phi_j} \frac{\partial E^b_{r,j}}{\partial \Phi_j} \tag{193}$$

$$= C^b_j + \frac{\partial}{\partial \Phi_j}(C^b_j)(\Phi_j - \pi) \tag{194}$$

8 Literatur

[Baraff und Witkin, 1998] Baraff, D. und Witkin, A. (1998). Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA. ACM.

[Breen et al., 1994] Breen, D. E., House, D. H., und Wozny, M. J. (1994). Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH*.

[Eberhardt et al., 1996] Eberhardt, B., Weber, A., und Strasser, W. (1996). A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications*, 16(5):52–59.

[Eischen et al., 1996] Eischen, J. W., Deng, S., und Clapp, T. G. (1996). Finite-element modeling and control of flexible fabric parts. *IEEE Computer Graphics and Applications*, 16(5):71–80.

[Lindberg et al., 1961] Lindberg, J., Behre, B., und Dahlberg, B. (1961). Part iii: Shearing and buckling of various commercial fabrics. *Textile Research Journal*, 31(2):99–122.

[Magnenat-Thalmann et al., 2007] Magnenat-Thalmann, N., Luible, C., Volino, P., und Lyard, E. (2007). From measured fabric to the simulation of cloth. In *2007 10th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pages 7–18. IEEE.

[Provot et al., 1995] Provot, X. et al. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society.

[Weil, 1986] Weil, J. (1986). The synthesis of cloth objects. *ACM Siggraph Computer Graphics*, 20(4):49–54.

[Zhang und Yuen, 2001] Zhang, D. und Yuen, M. M. (2001). Cloth simulation using multilevel meshes. *Computers & Graphics*, 25(3):383–389.