Bachelorarbeit

Modellierung der Elektrophysiologie des Herzens

Julian Lucas Hilbert
Studiengang: B.Sc. Informatik
Matrikelnummer: 5533406

Frankfurt am Main
xx.xx.2018

Examiner: ¡Examiner¿
Advisor: ¡Advisor¿

Professorship for ¡Working Group¿
Institute of Computer Science
Department of Computer Science and Mathematics
Goethe University Frankfurt

Erklärung zur Abschlussarbeit

gemäß § 25, Abs. 22 der Ordnung für den Bachelorstudiengang Informatik vom 06. Dezember 2010

Hiermit erkläre ich Herr

Julian Lucas Hilbert

Die vorliegende Arbeit habe ich selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst.

Ebenso bestätige ich, dass diese Arbeit nicht, auch nicht auszugsweise, für eine andere Prüfung oder Studienleistung verwendet wurde.

Zudem versichere ich, dass die von mir abgegebenen schriftlichen gebundenen Versionen meiner Bachelorarbeit mit der auf einem Datenträger abgegebenen elektronischen Version übereinstimmen.

Frankfurt am Main, den xx.xx.2018

Zusammenfassung

TODO

Contents

1 Einführung

Das Herz ist ein muskuläres Organ, welches durch rhythmische Kontraktionen Blut durch den Körper pumpt. Ein Herzschlag wird durch elektrische Anregung der Herzmuskelzellen ausgelöst, wobei dies durch spezialisierte Zellen geschieht.[1, S. 29] Hierbei handelt es sich um einen hochkomplexen Prozess, TODO: IN BUCH NACHSCHAUEN WIE GENAU DAS GEHT...

1.1 Anatomie des Herzens

1.2 Elektrische Signalübertragung in Muskelfasern

Jede Herzmuskelzelle ist von einer Zellmembran umgeben, welche den intrazellulären Raum vom extrazellulären Raum abtrennt. In diese Membran eingelassen sind Proteine, die als Verbindung zwischen intrazellulären Raum und extrazellulären Raum dienen. Oft sind diese Proteine nur durchlässig für eine bestimmte Art von Ionen, dies wird als selektive Permeabilität bezeichnet. [2] Im Folgenden werden diese Proteine als *Ionenkanäle* bezeichnet.
Wenn nun ein Konzentrationsgefälle einer bestimmten Ionensorte α an der Membran vorliegt während ein Ionenkanal für Ionen vom Typ α durchlässig ist, werden diese Ionen von dem Bereich höherer Konzentration in den Bereich niedriger Konzentration fließen. Da Ionen grundsätzlich elektrisch geladen sind führt dies zu einer Bewegung elektrischer Ladung und somit zum Stromfluss, wodurch sich eine Potentialdifferenz zwischen intra- und extrazellulärem Raum aufbaut, welche der Diffusion bezüglich dem Konzentrationsgefälle entgegenwirkt.
Die Potentialdifferenz E_α für einen Ionentyp α an der Zellmembran kann durch die Nernstgleichung (siehe Equation 1.1) berechnet werden.

$$E_\alpha = \frac{RT}{z_\alpha F} \cdot \ln\left(\frac{c_{\text{extra}}(\alpha)}{c_{\text{intra}}(\alpha)}\right) \tag{1.1}$$

Hierbei ist T die Temperatur in Kelvin, was im Körper ungefähr 309K entspricht.
$R = 8.314$ J mol^{-1} K^{-1} ist die universelle Gaskonstante, $F = 96.485$ C^3 mol^{-3} die Faraday'sche Konstante und z_α die absolute Ladung des Ions.[1, S. 20-23]
Die Ionentypen die das Membranpotential am meisten beeinflussen sind Natirum-, Kalium- und Calciumionen. Die jeweiligen Konzentrationen auf beiden Seiten der Zellmembran

Ion	$C_{\text{Intrazellulär}}$	$C_{\text{Extrazellulär}}$	E_α
$[Na^+]$	10	140	70
$[K^+]$	145	5.4	−88
$[Ca^{2+}]$	$1.2 \cdot 10^{-5}$	1.8	128

Figure 1.1: Ionenkonzentrationen inner- und außerhalb der Herzmuskelzelle in mmol $\cdot L^{-1}$, sowie das errechnete Nernstpotential in mV. [1, S. 22]

sowie die daraus resultierende Potentialdifferenz sind in Figure 1.1 aufgelistet.
Typischerweise liegt das Ruhemembranpotential einer Herzmuskelzelle bei −85 mV. Auffällig ist, dass dies relativ nah an dem Kaliumpotential E_{K^+} liegt, da die Permeabilität für Kaliumionen in Ruhe höher ist als die der anderen Ionen.

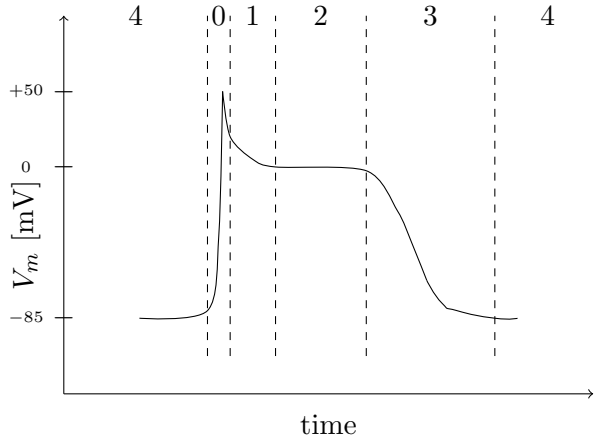

Figure 1.2: Zeitlicher Verlauf des Aktionspotentials einer Herzmuskelzelle. Durch die gestrichelten Linien werden die vier Phasen aufgezeigt, die Ruhephase ist durch die *4* gekennzeichnet. Die Abbildung ist jener in [1, S. 24] nachempfunden.

Wenn eine äußere Anregung in Form eines Impulses das Membranpotential über einen kritischen Punkt verändert, erfolgt eine aktive Reaktion der Zelle. Diese Reaktion wird dabei als *Aktionspotential* bezeichnet. Dabei ist das Aktionspotential normalerweise unabhängig von der Amplitude des Impulses, solange eine bestimmte Grenzwertamplitude überschritten wird. In der ersten Phase des Aktionspotentials (siehe Figure 1.2 Phase 0) steigt das Membranpotential innerhalb kurzer Zeit stark bis in den positiven Bereich an. Dies ist darauf zurückzuführen, dass sich aufgrund des anfänglichen Impulses spannungsaktivierende Natriumkanäle öffnen und somit die extrazellulär höher konzentrierten Na^+ - Ionen in die Zelle einströmen, was dazu führt, dass sich das Membranpotential V_m dem Gleichgewichtspotential von Natrium E_{Na^+} annähert (Depolarisation).
Nach Phase 0 werden die Natriumkanäle wieder geschlossen und V_m fällt rasch gegen 0 mV

ab (Phase 1).
Anschließend kommt es zu Phase 2, auch Plateuphase genannt, wo V_m über einen Zeitraum von bis zu mehreren hundert Milisekunden relativ konstant gehalten wird. Hier wird ein Ausstrom von positiven Ionen durch einen Einstrom von Ca^{2+}- Ionen ausgeglichen. In der Repolarisation (Phase 3) nähert sich V_m wieder gegen das Ruhepotential von -85 mV, da die Membrandurchlässigkeit für K^+- Ionen ansteigt. Dabei kann es vorkommen, dass das Membranpotential zunächst auf etwas weniger als -85mV abfällt. Auffällig ist, dass die Repolarisation nicht durch den Ausstrom der in Phase 0 eingeströmten Na^+- Ionen geschieht, sondern durch ausströmen von bereits in der Zelle vorhandenen K^+- Ionen. Sobald das Ruhepotential wieder erreicht wurde, befindet sich das System wieder Phase 4. Um die nun gestörten Ausgangskonzentrationen an K^+- Ionen und Na^+- Ionen wieder herzustellen arbeitet unter Anderem die Na^+-K^+- Pumpe. Diese tauscht jeweils $3Na^+$- Ionen von innerhalb der Zelle mit $2K^+$- Ionen außerhalb der Zelle unter Verbrauch von Adenosintriphosphat. [3][1, S. 23-25]

Herzmuskelzellen sind im Myokard durch so genannte *Glanzstreifen* (englisch *intercalated disks*) miteinander verknüpft. Die elektrische Verbindung zwischen den einzelnen, benachbarten Zellen erfolgt über *Gap Junctions*, welche in den Glanzstreifen enthalten sind. Dabei handelt es sich um eine Ansammlung von Kanälen zwischen beiden Zellen über die ein Austausch von Signalen stattfinden kann. Die für den Herzschlag entscheidende Kontraktion der Herzmuskelzellen beginnt mit der oben beschriebenen elektrischen Anregung.
[4]

1.3 Mathematische Grundlagen

1.3.1 ¡prolate spheroidal coordinates¿

Eine Darstellung der Herzgeometrie in kartesischen Koordinaten wird im Grunde nur für die Darstellung in ProMesh, sowie die anschließende Rechnung verwendet. Ansonsten erfolgt die Darstellung in ¡prolate spheroidal coordinates¿. Es handelt sich dabei um ein orthogonales Koordinatensystem, welches durch Rotation eines zweidimensionalen elliptischen Koordinatensystem um die Hauptachse der Ellipse entsteht.
Ein einzelner Punkt wird somit bei gegebenem Brennpunkt a eindeutig durch den Schnittpunkt von drei Oberflächen gegeben: λ entspricht einem Rotationsellipsoid, μ einem Hyperboloid und θ beschreibt die oben erwähnte Rotation einer Halbebene um die Hauptachse der Ellipse (siehe Figure 1.4). Die Umwandlung von ¡prolate spheroidal coordinates¿(λ, μ, θ) in kartesische Koordinaten(x, y, z) erfolgt über folgende Transformation [5]:

$$\begin{aligned} x &= a \cosh\lambda \cos\mu \\ y &= a \sinh\lambda \sin\mu \cos\theta \\ z &= a \sinh\lambda \sin\mu \sin\theta \end{aligned} \tag{1.2}$$

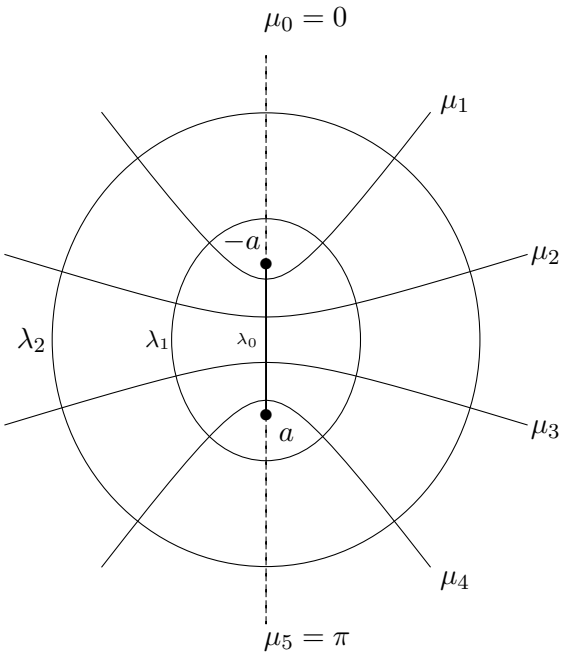

Figure 1.3: Elliptisches Koordinatensystem. Punkt wird beschrieben durch die Schnittstelle einer Hyperbel μ_i mit einer Ellipse λ_i, wobei diese konfokal sind, also die gleichen Brennpunkte (a und $-a$) besitzen.

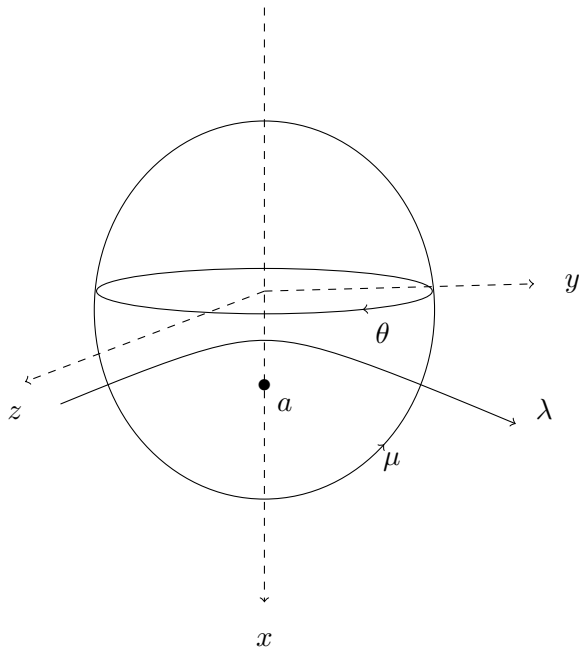

Figure 1.4: Darstellung eines ¡prolate spheroidal coordinate system¿ welches durch Rotation eines elliptischen Koordinatensystems um die x - Achse entsteht. Anschaulich betrachtet: Bei fixiertem μ und θ führt eine Änderung von λ zu einer Bewegung auf einer Hyperbel, bei fixiertem λ und θ führt eine Änderung von μ zu einer Bewegung auf einer Ellipse und bei fixiertem λ und μ führt eine Änderung von θ zu einer Kreisbewegung.

Verwenden eines solchen Koordinatensystem hat von Vorteil, dass es - vorausgesetzt günstige Brennpunkte wurden gewählt - bereits annähernd die Form eines Herzens beschreit. Im Datensatz wurde hierfür der Brennpunkt auf 35.25 gesetzt.
Zu beachten ist außerdem, dass die x - Achse von oben nach unten läuft.

1.3.2 Lokales Koordinatensystem

Elemente des Gitters werden durch eine Teilmenge der Knotenmenge definiert. Knoten, die einem Element angehören werden als lokale Knoten des Elements bezeichnet. Auf diesen lokalen Knoten wird nun ein lokales, orthonormales Koordinatensystem erstellt, wobei jeder lokale Knoten an einem Eckpunkt eines Würfels mit Seitenlänge eins liegt. Die Raumrichtungen des lokalen Koordinatensystems werden mit ξ_i bezeichnet. Zur vereinfachten Darstellung wird dies zunächst in zwei Raumdimensionen veranschaulicht.
In Figure 1.5 bilden jeweils vier Knoten ein Element, wobei jeder Knoten durch einen Kreis symbolisiert wird.
Das Beispielelement besteht in diesem Fall aus den Knoten mit globalen Inidizes aus der Indexmenge $\{a, b, c, d\}$. Unabhängig davon wird jedem Knoten in der lokalen Knotenmenge ein Index aus der Menge $\{1, 2, 3, 4\}$ zugewiesen.

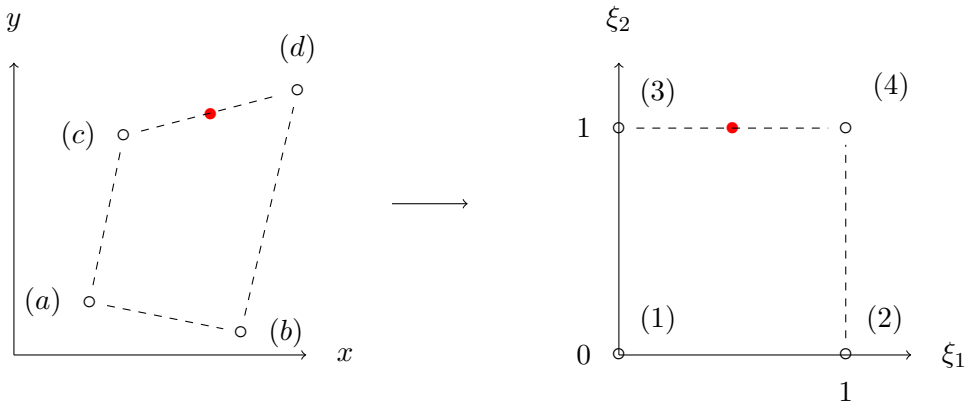

Figure 1.5: Darstellung eines viereckigen Elements in globalen (x, y)-Koordinaten (links) und die Abbildung auf die lokalen Elementkoordinaten (rechts).

Benachbarte Elemente im Gitter teilen sich die Knoten, über die sie in Kontakt stehen (siehe Figure 1.6). Somit teilen sich die Elemente an diesen Stellen auch die gleichen in den Knoten definierten Parameter, falls vorhanden. Um ein vorhandenes Gitter zu verfeinern müssen neue Knoten generiert werden. Dies kann Elementweise durch Interpolation der lokalen Knoten erreicht werden. Dazu bietet es sich das lokale Koordinatensystem in Verbindung mit geeigneten Basisfunktionen an.
In drei Raumdimensionen funktioniert dies analog.

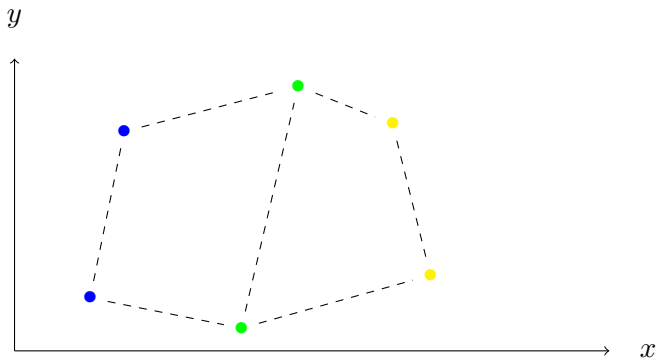

Figure 1.6: Zwei benachbarte Elemente. Die blauen Knoten gehören der lokalen Knotenmenge von Element 1 an, die gelben Knoten der lokalen Knotenmenge von Element 2 und die grünen Knoten befinden sich in den lokalen Knotenmengen beider Elemente.

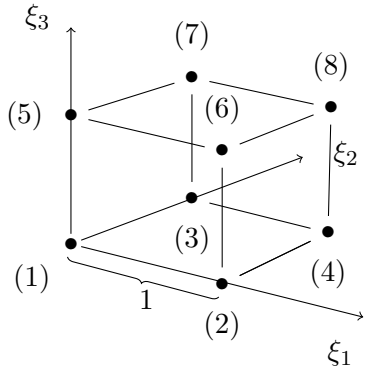

Figure 1.7: lokales Koordinatensystem in drei Raumrichtungen. Angegeben sind die lokalen Knotenindizes.

1.3.3 Interpolation

Das generieren neuer Knoten in einem Element kann durch Interpolation mit verschiedenen Basisfunktionen erreicht werden. Zunächst wird wieder der zweidimensionale Fall betrachtet. Relativ einfach geht dies mit den linearen Lagrange Basisfunktionen Ψ_i. Diese sind in Abhängigkeit von den lokalen Koordinaten $0 \leq \xi_i \leq 1$ wie folgt definiert [1, S. 52]:

$$\begin{aligned} \Psi_1(\xi_1, \xi_2) &= (1 - \xi_1)(1 - \xi_2) \\ \Psi_2(\xi_1, \xi_2) &= \xi_1(1 - \xi_2) \\ \Psi_3(\xi_1, \xi_2) &= (1 - \xi_1)\xi_2 \\ \Psi_4(\xi_1, \xi_2) &= \xi_1\xi_2 \end{aligned} \tag{1.3}$$

Sei nun v ein neuer Knoten, der an irgendeinem Punkt (ξ_1, ξ_2) im lokalen liegt. Für die in v definierten Parameter $u(\xi_1, \xi_2)$, gilt anschließend:

$$u(\xi_1, \xi_2) = \Psi_1(\xi_1, \xi_2)u^{(1)} + \Psi_2(\xi_1, \xi_2)u^{(2)} + \Psi_3(\xi_1, \xi_2)u^{(3)} + \Psi_4(\xi_1, \xi_2)u^{(4)} \tag{1.4}$$

Wobei $u^{(i)}$ dem Knoten mit Index i des lokalen Koordinatensystems als Parameter zugewiesen wurde.
Aus Equation 1.3 ist ersichtlich, dass für $(\xi_1, \xi_2) \in (\{0, 1\} \times \{0, 1\})$ jeweils die Werte $u^{(1)}, u^{(2)}, u^{(3)}$ und $u^{(4)}$ berechnet werden. Stetigkeit der Interpolation über die Elementgrenzen ist durch das Teilen der Kontaktknoten zweier Elemente gegeben.
Eine Einschränkung der linearen Lagrange Interpolation ist die Differenzierbarkeit. Bei dem Übergang von einem Element zum nächsten ist zwar der Wert der Parameter im geteilten Knoten identisch und somit stetig, dies muss jedoch nicht für die Ableitung gelten.
Dies ist anschaulich im eindimensionalen Fall zu erkennen. Figure 1.8

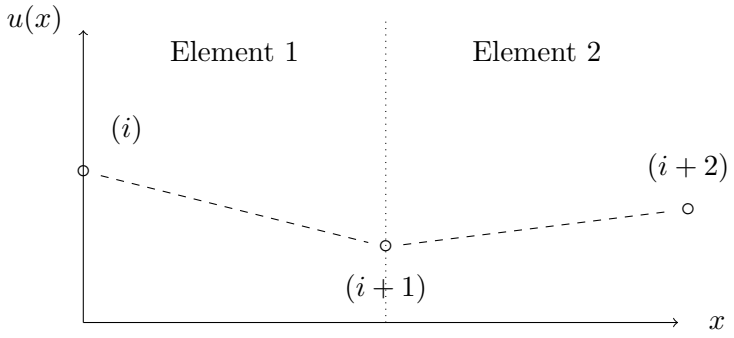

Figure 1.8: Eindimensionale lineare Interpolation. Die gestrichelte Linie beschreibt eine lineare Lagrange Interpolation zwischen jeweils zwei Punkten.

An der Grenze zwischen Element 1 und Element 2 in Knoten $i + 1$ hat die Funktion $u(x)$ einen Knick und ist somit nicht differenzierbar.
Die Lagrange Basisfunktionen gehören somit zur Differentiationsklasse C^0 [1, S. 55]. Es bietet sich an, Basisfunktionen zu wählen welche in C^1 liegen. Dies wird ermöglicht durch kubisch hermitesche Basisfunktionen. Zusätzlich zu den Parametern werden in jedem Knoten außerdem Ableitungen entsprechend der Raumrichtungen gesetzt. Da die Parameter in den Knoten benachbarter Elemente an den Kontaktstellen übereinstimmen gilt dies natürlich auch für die Ableitungen. Im eindimensionalen Fall sind also zwei abhängige Variablen pro Knoten benötigt: $u, \frac{\partial u}{\partial \xi}$. In zwei Raumdimensionen schon vier: $u, \frac{\partial u}{\partial \xi_1}, \frac{\partial u}{\partial \xi_2}, \frac{\partial^2 u}{\partial \xi_1 \xi_2}$. [1, S. 58]

Die eindimensionalen kubisch hermitesche Basisfunktionen sind wie folgt definiert:

$$\begin{aligned} \mathrm{H}_1^0(\xi) &= 1 - 3\xi^2 + 2\xi^3 \\ \mathrm{H}_1^1(\xi) &= \xi(\xi - 1)^2 \\ \mathrm{H}_2^0(\xi) &= \xi^2(3 - 2\xi) \\ \mathrm{H}_2^1(\xi) &= \xi^2(\xi - 1) \end{aligned} \tag{1.5}$$

Aus den in Equation 1.5 gegebenen kubischen Polynomen lässt sich folgende Abhängigkeit des Parameters u in zwei Raumdimensionen definieren[6]:

$$\begin{aligned} u(\xi_1, \xi_2) = {} & \mathrm{H}_1^0(\xi_1)\mathrm{H}_1^0(\xi_2)u^{(1)} + \mathrm{H}_2^0(\xi_1)\mathrm{H}_1^0(\xi_2)u^{(2)} \\ & + \mathrm{H}_1^0(\xi_1)\mathrm{H}_2^0(\xi_2)u^{(3)} + \mathrm{H}_2^0(\xi_1)\mathrm{H}_2^0(\xi_2)u^{(4)} \\ & + \mathrm{H}_1^1(\xi_1)\mathrm{H}_1^0(\xi_2)\left(\frac{\partial u}{\partial \xi_1}\right)^{(1)} + \mathrm{H}_2^1(\xi_1)\mathrm{H}_1^0(\xi_2)\left(\frac{\partial u}{\partial \xi_1}\right)^{(2)} \\ & + \mathrm{H}_1^1(\xi_1)\mathrm{H}_2^0(\xi_2)\left(\frac{\partial u}{\partial \xi_1}\right)^{(3)} + \mathrm{H}_2^1(\xi_1)\mathrm{H}_2^0(\xi_2)\left(\frac{\partial u}{\partial \xi_1}\right)^{(4)} \\ & + \mathrm{H}_1^0(\xi_1)\mathrm{H}_1^1(\xi_2)\left(\frac{\partial u}{\partial \xi_2}\right)^{(1)} + \mathrm{H}_2^0(\xi_1)\mathrm{H}_1^1(\xi_2)\left(\frac{\partial u}{\partial \xi_2}\right)^{(2)} \\ & + \mathrm{H}_1^0(\xi_1)\mathrm{H}_2^1(\xi_2)\left(\frac{\partial u}{\partial \xi_2}\right)^{(3)} + \mathrm{H}_2^0(\xi_1)\mathrm{H}_2^1(\xi_2)\left(\frac{\partial u}{\partial \xi_2}\right)^{(4)} \\ & + \mathrm{H}_1^1(\xi_1)\mathrm{H}_1^1(\xi_2)\left(\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}\right)^{(1)} + \mathrm{H}_2^1(\xi_1)\mathrm{H}_1^1(\xi_2)\left(\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}\right)^{(2)} \\ & + \mathrm{H}_1^1(\xi_1)\mathrm{H}_2^1(\xi_2)\left(\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}\right)^{(3)} + \mathrm{H}_2^1(\xi_1)\mathrm{H}_2^1(\xi_2)\left(\frac{\partial^2 u}{\partial \xi_1 \partial \xi_2}\right)^{(4)} \end{aligned} \tag{1.6}$$

Für die Interpolation im Herzmodell wird eine Kombination der obigen Verfahren benutzt um eine dreidimensionale Interpolation in den einzelnen Elementen zu ermöglichen.
Die Winkel θ und μ werden jeweils in alle drei Raumrichtungen mit linearen Lagrange Basisfunktionen interpoliert. λ hingegen wird in ξ_1- und ξ_2- Richtung mit kubisch hermiteschen Basisfunktionen und in ξ_3-Richtung linear interpoliert.
Da angenommen wird (TODO:REFERENZ), dass die Muskelfasern in der (ξ_1, ξ_2) Ebene liegen, wird beim Faserwinkel η in ξ_1- und ξ_2-Richtung linear- und in ξ_3 -Richtung kubisch hermitesch interpoliert[6].

2 Das Herzmodell

2.1 Geometrie

Um auf der Geometrie des Herzens bekannte Gleichungen zu lösen empfiehlt es sich die Geometrie in diskrete Teilgebiete (Elemente) aufzuteilen. In diesem Fall wurden die einzelnen Elemente durch einen Fitting-Prozess auf Basis von experimentell ermittelten dreidimensionalen Punkten erstellt.[5]
Im Datensatz enthalten sind die Koordinaten von 99 Knoten, welche 60 Elemente in Form von Hexaedern bilden. Bei den "Hexaedern" am Fuß des Herzens handelt es sich im Datensatz zwar um Hexaeder, jedoch bilden dort in Form eines *Edge-Collapse* mehrere lokale Knoten auf den selben globalen Knoten ab, sodass die Form eines Prismas entseht (siehe Figure 2.1). ProMesh unterstützt dieses Feature jedoch nicht, also wurden diese Elemente direkt als Prisma implementiert.

In jedem Knoten eines Elements sind Ableitungen entsprechend den lokalen Koordinaten (siehe Subsection 1.3.3) gegeben, sodass eine neuer Knoten gegebenenfalls auch durch kubische hermitesche Interpolation erzeugt werden kann. Dies wird im Folgenden auch verwendet, um feinere Gitter aus dem Datensatz zu generieren.
Da ein Knoten jeweils zu mehreren aneinandergrenzenden Elementen gehören kann, ist eine stetige Differenzierbarkeit über die einzelnen Elemente gegeben.
Wichtig für die Modellierung des Herzens sind außerdem die Ausrichtungen der Muskelfasern, da diese letztendlich die elektrische Leitfähigkeit der Herzmuskulatur mitbestimmen. In jedem Knoten ist hierfür der Winkel der in diesem Punkte verlaufenden Muskelfasern zwischen den lokalen Koordinatenachsen ξ_1 und ξ_2 angegeben (siehe Figure 2.3). Die Faserwinkel für neu erstellte Knoten werden ebenfalls durch Interpolation aus den Faserwinkeln der benachbarten Knoten berechnet.

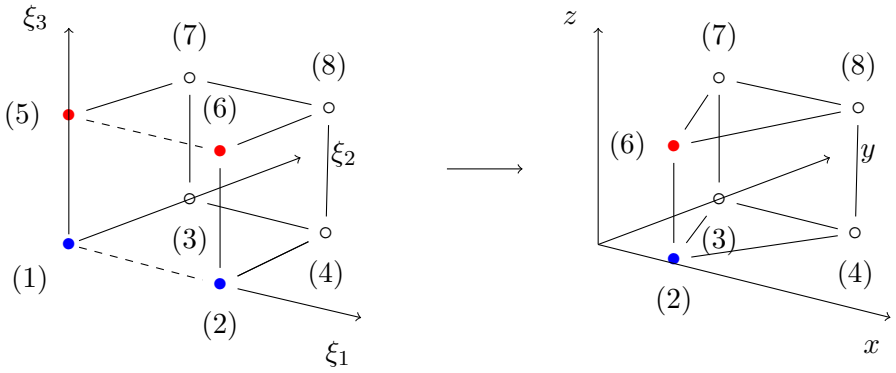

Figure 2.1: Beispiel für einen *Edge Collapse*: jeweils zwei farblich markierte Knoten im lokalen Koordinatensystem bilden auf denselben globalen Knoten ab. Es ergibt sich dir Form eines Prismas.

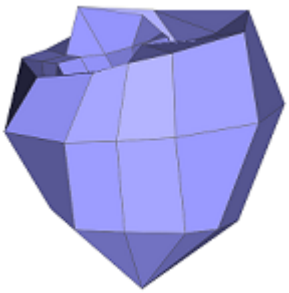

Figure 2.2: Darstellung der Herzgeometrie mit 99 Knoten in ProMesh

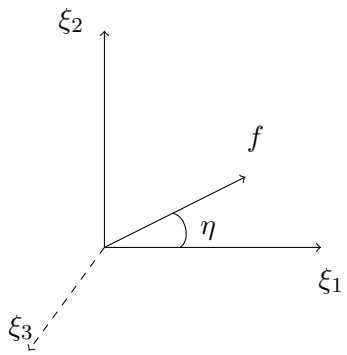

Figure 2.3: Faserwinkel η, angegeben in lokalen Koordinaten, wobei angenommen wird dass dieser in der (ξ_1, ξ_2)- Ebene liegt.

2.2 EX File Format und OpenCMISS Zinc

Der Datensatz besteht aus jeweils einer `heart.exnode` und `heart.exelem` Datei.
In `heart.exnode` sind 99 Knoten mit den räumlichen Koordinaten (in ¡prolate spheroidal coordinates¿) sowie den Faserwinkeln gegeben. Falls ein Wert durch Polynome höheren Grades interpoliert wird, sind außerdem die jeweils benötigten Ableitungen des Werts bezüglich der lokalen Koordinaten gespeichert.
`heart.exelem` beschreibt die einzelnen Elemente des Modells, also welche Knoten zu welchen Elementen gehören und welche Basisfunktionen zur Interpolation neuer Knoten verwendet werden. Genaueres zum sogenannten `EX File Format` unter [7].

Die Daten können mit der `OpenCMISS-Zinc` API gelesen und bearbeitet werden. Diese stellt unter Anderem Anbindungen für Python und C++ bereit, welche für diese Arbeit verwendet wurden (Version 1.3.0.20180409141131 der Linux Development-Distribution; Download und Hinweise zur Installation unter [8]).
Um mit der Zinc- Bibliothek zu arbeiten muss zunächst ein `Context` Objekt erstellt werden, welches das einzige Objekt der Bibliothek ist das von keiner Methode eines anderen Objekts erstellt wird. Daraus folgt auch, dass jedes andere benötigte Objekt direkt oder indirekt von diesem erstellt wird. Ein `Region` Objekt wird beispielsweise direkt über `Context.getDefaultRegion()` erstellt und ermöglicht es eine Datei des EX File-Formats einzulesen. Die wichtigsten Objekte der Zinc Bibliothek und wofür diese verwendet wurden werden nun kurz angeschnitten:

Context

Wie schon oben erwähnt muss in jedem Fall ein `Context` erstellt werden, um mit der Zinc- Bibliothek erbeiten zu können.

Region

Kann als Analogon einer Ordnerhierarchie angesehen werden. Falls dem Benutzer eine Region ausreicht, kann vom Context eine standard-Region erstellt werden, was auch verwendet wurde. Die Region vermag es außerdem, Dateien vom EX-File Format zu lesen und zu schreiben.

Fieldmodule

Erstellt von `Region` . Dieses Objekt ist hauptsächlich dazu da, Objekte vom Typ `Field` zu verwalten. Wird verwendet um einen Großteil der verwendeten Objekte zu erzeugen.

Fieldcache

Erstellt von `Fieldmodule`. Im Fieldcache können Orte zur späteren Auswertung

gespeichert werden. Dieser Ort kann beispielsweise ein Element inklusive lokalen ξ_i-Koordinaten oder ein Knoten sein.

Mesh/Nodeset

Erstellt von `Fieldmodule`. Ein Mesh beschreibt eine Menge an Elementen, wobei jedem Element ein Index zugeordnet ist. Besitzt Funktionen um Elemente zu erstellen oder zu löschen.
Nodeset verhält sich ähnlich, nur dass es sich, wie der Name bereits vermuten lässt, um eine Menge von Knoten handelt.

Field

Erstellt von `Fieldmodule`. Speichert die Werte der Parameter an den Knoten/Elementen. Soll beispielsweise ein Parameter eines Knotens ausgegeben werden, muss dieser Knoten im Fieldcache als Ort gespeichert werden, woraufhin das Feld an der im Fieldcache spezifizierten Stelle abgefragt wird.

Eine Dokumentation der Zinc-API ist hier zu finden:[9]

3 Praktischer Teil

3.1 Generieren feiner Gitter

Offensichtlicher Weise ist ProMesh nicht in der Lage, Dateien des EX-File Formats zu lesen. Am Ende muss also zur Visualisierung eine Umwandlung in ein von ProMesh lesbares Format geschehen. In diesem Fall wurde das *Visualisation Toolkit*, kurz VTK verwendet. Dieses bietet die Möglichkeit ein unstrukturiertes Gitter in XML-Format mit der Endung `.vtu` abzuspeichern. Dafür wurde ein Konverter geschrieben, der die erstellten Gitter in eine `.vtu`- Datei übersetzt welche anschließend von ProMesh gelesen werden kann.

Zuständig für das Erstellen feinerer Gitter ist die das Klasse `mesh_generator`. Diese bietet die Möglichkeit über verschiedene `set`- Funktionen die Anzahl der Verfeinerungen in die drei Raumrichtungen für das Herz und die Muskelfasern zu bestimmen, sowie die Namen der verwendeten Dateien zu ändern.
Der `mesh_generator.set_refinement(v,h,r)` Befehl nimmt drei Integer ≥ 0 entgegen, und setzt die Verfeinerung auf v- mal in vertikale Richtung, h- mal in horizontale Richtung und r- mal in radiale Richtung.
(Analoges gilt für die `mesh_generator.set_refinement_fibers(v,h,r))` -Funktion.)
Die Verfeinerung beruht auf dem Prinzip, dass aus einem gegebenen Element neue Knoten über Interpolation generiert werden, welche wiederum neue Elemente bilden. Da jedes Element auf ein lokales, orthonormales Koordinatensystem abbildet, bei dem die jeweiligen Eckpunkte eines Würfels mit Kantenlänge 1 die Knoten des Elements darstellen (vergleiche Figure 1.3.2), kann die Verfeinerung auf diesem lokalen Koordinatensystem stattfinden. Anschließend muss lediglich eine Rücktransformation auf die globalen Koordinaten stattfinden. Dies ist in zwei Dimensionen anschaulich in Figure 3.1 dargestellt. Dort findet in beide Raumrichtungen eine einfache Verfeinerung statt. Aus einem Element werden also vier neue Elemente generiert. Grundsätzlich ist es so, dass jede Verfeinerungsstufe in eine Raumrichtung die Anzahl der Elemente in dieser Raumrichtung verdoppelt. Im dreidimensionalen Fall des Herzens nimmt also die Anzahl der Elemente mit Faktor $2^{(v+h+r)}$ zu, wobei v, h und r die Anzahl der Verfeinerungen in die Raumrichtungen sind. Da zu Beginn 60 Elemente existieren, gilt für die Gesamtanzahl an Elementen nach der Verfeinerung:

$$\#\text{Elemente } = 60 \cdot 2^{(v+h+r)}$$

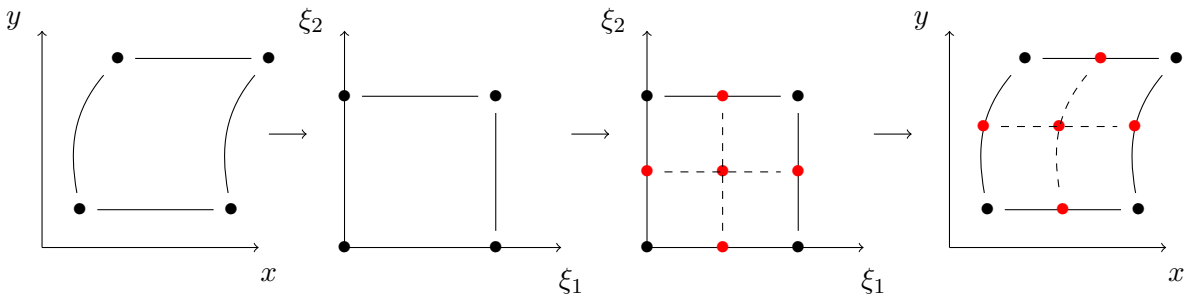

Figure 3.1: Graphische Darstellung der Verfeinerung eines verzerrten Elements. Zunächst wird das Element in lokalen Koordinaten umgewandelt, dort werden neue Knoten generiert (Als rote Punkte dargestellt). Das Element wird mit den neuen Knoten zurück in globale Koordinaten transformiert. Zu erkennen ist, dass die Elementgrenzen der neuen Elemente ebenfalls verzerrt sind.

Der Algorithmus zur Verfeinerung des Gitters iteriert über alle Elemente des Gitters und unterteilt jedes Element gemäß der gewählten Verfeinerungsstufen in neue Elemente mit gegebenenfalls neuen Knoten. Unter Betrachtung von Figure 3.2 lässt sich feststellen, dass ein Knoten v_i am Rand eines Elements die gleichen globalen Koordinaten wie ein bereits generierter Knoten v_j am Rand eines benachbarten Elements haben kann. Da die Anforderung an die Stetigkeit über Elementgrenzen hinweg erfüllt ist, müssen v_i und v_j deshalb die gleichen Parameter haben, also gilt $u(v_i) = u(v_j)$ und es kann v_j für v_i verwendet werden.

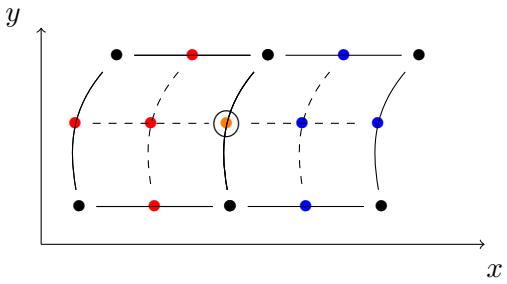

Figure 3.2: Zwei benachbarte, verfeinerte Elemente. Zu beachten ist, dass der markierte Knoten beim Bearbeiten beider Elemente entsteht, falls keine weiteren Vorkehrungen getroffen werden.

Die Herausforderung besteht nun darin, Knoten gleicher Position zu erkennen und anschließend zu verwenden. Ein naiver Ansatz wäre dafür jeden potentiell zu erstellenden Knoten mit jedem bereits verwendeten Knoten zu vergleichen und erst wenn der Vergleich mit allen Knoten negativ war, diesen zu erstellen. Dies führt aber für große Knotenmengen zu nicht akzeptabler Rechenzeit.

Die verwendete Methode um doppelte Knoten zu vermeiden greift auf den naiven Ansatz zurück, führt diesen allerdings auf Knotenmengen aus, die durch ein vorsortieren sehr viel kleiner sind als.
Vor dem eigentlichen Verfeinern wird jedem Knoten v_i ein Wert $w(v_i)$ zugeordnet, welcher der Summe seiner x, y und z Koordinaten entspricht. Daraufhin werden 10.000 Behälter erstellt und jedem Behälter B_j wird ein gleichgroßes Intervall I_j zugeordnet, wobei das Minimum (Maximum) aller $w(v_i)$ im Intervall des ersten (letzten) Behälters liegt.
Anschließend kann für alle Knoten bestimmt werden: Knoten v_i ist genau dann Behälter B_j zuzuordnen, wenn gilt: $w(v_i) \in I_j$.

Sei nun v_{neu} ein potentieller Knoten, mit $w(v_{\text{neu}}) \in I_j$. Es wird nun v_{neu} mit allen Knoten $v_i \in B_{j-1} \cup B_j \cup B_{j+1}$ verglichen. Sollte einer dieser Knoten mit v_{neu} übereinstimmen, wird dieser Knoten verwendet. Andernfalls wird v_{neu} erstellt und B_j zugeordnet.
Für den Fall dass der Wert eines Knotens nahe einer Intervallgrenze eines Behälters liegt und Rundungsfehler nicht ausgeschlossen werden können, wird in der Vereinigung des Behälters mit seinen beiden benachbarten Behältern gesucht.

Bibliography

[1] Martin L Buist Andrew J Pullan, Leo K Cheng. *Mathematically Modelling the Electrical Activity of the Heart*. Wiley, 2005.

[2] Jennifer E. Fairman Clare O'Connor, Jill U. Adams. *Essentials of Cell Biology*. Scitable, 2014.

[3] Hue-Teh Shih. Anatomy of the action potential in the heart. *Texas Heart Institute Journal*, 21(1):30–41, 1994.

[4] Gülistan Meşe, Gabriele Richard, and Thomas W. White. Gap junctions: Basic structure and function. *Journal of Investigative Dermatology*, 127:2516–2524, 2007.

[5] Ian LeGrice, Peter Hunter, Alistair Young, and Bruce Small. The architecture of the heart: a data-based model. *the royal society*, 359, 2001.

[6] P. M. F. Nielsen, I. J. Le Grice, B. H. Smaill, and P. J. Hunter. Mathematical model of geometry and fibrous structure of the heart. *American Journal of Physiology-Heart and Circulatory Physiology*, 260:H1365–H1378, 1991.

[7] The ex file format. `http://opencmiss.org/documentation/data_format/ex_file_format.html`. Accessed: 2018-07-22.

[8] Opencmiss - getting started. `http://opencmiss.org/documentation/getting_started_oclibs.html`. Accessed: 2018-07-22.

[9] Opencmiss-zinc c++ api documentation. `http://opencmiss.org/documentation/apidoc/zinc/latest/index.html`. Accessed: 2018-07-22.