Objektorientierte FEM-basierte Simulation der Biomechanik des Kniegelenks auf parallelen Rechnerarchitekturen

Martin Wawro

Universität Dortmund - Lehrstuhl Informatik VII
Otto-Hahn-Str. 16, 44221 Dortmund
Email: wawro@ls7.cs.uni-dortmund.de

Zusammenfassung. Es wird eine auf FE-Methoden basierende Simulationsumgebung für das Kniegelenk vorgestellt. Gearbeitet wird auf Bilddaten, die *in-vivo* vom Patienten akquiriert und mittels 3D-Rekonstruktion in ein Hexaedernetz zur FE-Analyse transferiert werden. Die Simulation stützt sich auf nichtlineare FE-Analyse wobei Kollisionen zwischen verschiedenen Elementen erkannt und behandelt werden um die komplexe Roll-/Gleitbewegung des Kniegelenks realistisch nachzustellen. Die Konzeption der Simulationssoftware folgt einem objektorientiertem Ansatz um Erweiterbarkeit und Wartbarkeit der Software zu gewährleisten.

1 Motivation

Das Kniegelenk repräsentiert eines der wichtigsten und am stärksten belasteten Gelenke im menschlichen Körper, das für die Mobilität des Individuums eine tragende Rolle spielt. Dementsprechend stellen Erkrankungen oder Verletzungen (bspw. der Kreuzbandriß) eine Einschränkung der Mobilität des Patienten dar. Aus dem Kreuzbandriß, als eine häufige Folge von Sportunfällen, resultieren nicht selten permanente Instabilitäten im Kniegelenk. Als Folge von Fehlbelastungen durch die veränderte Dynamik in der Gelenkbewegung kommt es zu Abnutzungserscheinungen am Gelenkknorpel. Eine möglichst genaue Modellierung und Simulation der Dynamik des intakten Kniegelenks im Vergleich zum defekten Gelenk kann Ansatzpunkte zu einer verbesserten Verfahrensweise z.B. bei der Kreuzbandrekonstruktion liefern. Auch kann das zusätzliche Wissen über die Belastungsverhältnisse während der Bewegung in der Entwicklung von Knieendoprothesen eingesetzt werden.

Die Majorität der derzeit verwendeten Ansätze liefert nur ein kinematisches Modell der Kniegelenksbewegung (z.B. [1]). FEM-basierte Ansätze beschränken sich überwiegend auf *quasi-statische* Simulationsvorgänge ([2],[3]), ein dynamisches (aber proprietäres) Modell wird in [4] vorgestellt.

2 Methodik

Zur physikalisch-basierten Simulation verwenden wir die *Finite Elemente Analyse* (FEA), wobei die Konstruktion des Modells auf Basis von Bilddaten aus

der Magnetresonanztomographie (MRT) geschieht. Als Bildmaterial dienen 60 Sagittalschnitte eines intakten Kniegelenks mit $1.5mm$ Schichtdicke in einer Auflösung von 256×256 Bildpunkten pro Schicht. Die Schichten werden zunächst manuell segmentiert, mit einer Segmenteinteilung in a) Femur b) Tibia, c) Femurkondylenknorpel, d) Menisci, e) vorderes Kreuzband, f) hinteres Kreuzband. Die segmentierten Bilddaten werden dann mit dem *marching cubes* Verfahren zu einem 3D-Modell rekonstruiert und anschließend geglättet.

2.1 Modellbildung

Das Knotennetz für die FE-Analyse wird durch volumetrische Elemente strukturiert. Diese Hexaederlemente werden mit jeweils acht Knoten und linearen Ansatzfunktionen in einem 3D-Referenzkoordinatensystem realisiert. Durch die Wahl von volumetrischen Grundelementen lassen sich die im Knie vorhandenen Strukturen gut modellieren, insbesondere die Modellierung der Kreuzbänder, die in der Vergangenheit häufig durch Bündel von 1D-Linienelementen angenähert wurden, wird damit verbessert. Den Elementen werden Materialeigenschaften zugewiesen, welche experimentell an Gewebeproben bestimmt werden können. Die Generierung eines zusammenhängenden, nicht-degenerierenden, Hexaedernetzes aus den Oberflächendaten wurde semi-automatisch mittels einer Meshing-Software bewerkstelligt, dabei mußten z.T. Strukturen miteinander verschmolzen werden (Femur/Kondylenknorpel, Menisci/Tibia), die resultierenden Elemente erhielten jedoch die Materialeigenschaften für die entsprechenden Einzelgewebe. Es wird ein linear-elastisches isotropes Materialverhalten angenommen, für die Beurteilung der simulierten makroskopischen Bewegung im kleinen Rahmen, wie zunächst geschehen, ist diese Annäherung suffizient.

2.2 Simulation

Die Bewegungssimulation von artikulierenden Strukturen stellt zusätzliche Anforderungen an die verwendete FE-Software. Da die physiologische Bewegung des Kniegelenks eine kombinierte Roll-/Gleitbewegung von Femur und Tibia ist, muß das Simulationssystem in der Lage sein, *Kollisionen* zu erkennen und zu berücksichtigen. Eine lineare Formulierung des Problems kommt daher aus zwei Gründen nicht mehr in Betracht: a) Kollisionen, b) große Verschiebungen/Deformationen. Das Aufstellen der dynamischen Bewegungsgleichungen ergibt ein Differentialgleichungssystem:

$$\mathbf{M}\Delta\ddot{\mathbf{u}} + \mathbf{C}\Delta\dot{\mathbf{u}} + \mathbf{K_t}(\mathbf{u})\Delta\mathbf{u} = \mathbf{q_e} - \mathbf{q_i} \tag{1}$$

Wobei $\mathbf{M}$ und $\mathbf{C}$ die Massen- bzw. Dämpfungsmatrix darstellen und $\mathbf{K_t}(\mathbf{u})$ die Tangentensteifigkeitsmatrix, die sich aus der Linearisierung des nichtlinearen Problems ergibt. Der Vektor $\mathbf{u}$ enthält die gesuchten Verschiebungen, während die Vektoren $\mathbf{q_i}$ und $\mathbf{q_e}$ sich auf die externen und internen Kräfte beziehen, die auf das Modell einwirken. Zur Linearisierung des Gleichungssystem wurde das modifizierte Newton-Raphson Iterationsschema verwendet, wobei der Vektor $\Delta\mathbf{u}$

die Änderung des Gesamtverschiebungsvektors $\mathbf{u}$ beinhaltet, diese wird in jeder Iteration errechnet. Bei Erreichen eines Konvergenzkriteriums ist eine stationäre Lösung gefunden und der nächste Zeitschritt kann berechnet werden. Als numerisches Zeitintegrationsverfahren kommt das implizite *Newmark* Schema zur Anwendung ([5]).

Die Tangentensteifigkeitsmatrix $\mathbf{K_t}$ ist abhängig von den Materialeigenschaften und der aktuellen Deformation des Systems, sie ergibt sich als Diskretisierung der zweiten Variation der Deformationsenergie des Modells:

$$\Pi = \int_\Omega \frac{1}{2}\boldsymbol{\varepsilon}^{\boldsymbol{T}}\boldsymbol{\sigma}\ d\Omega \rightarrow \delta V_i = \int_\Omega \mathbf{B_{nl}^T C B_{nl}} + \mathbf{G^T \hat{S} G}\ \mathbf{d}\Omega = \mathbf{K_t} \tag{2}$$

wobei $\boldsymbol{\varepsilon}$ und $\boldsymbol{\sigma}$ die Dehnungen und Spannungen in einem Element darstellen. Das Einsetzen von (2) in (1) liefert für jeden Zeitschritt/Iterationsschritt ein lineares Gleichungssystem welches iterativ mit dem Verfahren der konjugierten Gradienten (CG) gelöst wird.

2.3 Kollisionen

Die Kollisionsbehandlung gliedert sich in die *Kollisionsdetektion* und die *Kollisionsantwort* des Systems. Die Detektion wird unter Benutzung von "*bounding boxes*" realisiert ([6]). Nach Feststellung der Kollisionspunkte auf den Oberflächen werden mittels des *Penalty* Verfahrens ([7]) Zwangsbedingungen auf das Gleichungssystem gelegt, welches dann erneut gelöst wird. Anschaulich entspricht dieser Ansatz dem Einfügen von Federn an den Kontaktpunkten, welche Rückstellkräfte auf die beiden kollidierenden Oberflächen ausüben. Die von uns realisierte Kollisionbehandlung ([8]) benötigt kein *a priori* Wissen über Kollisionen und auch keine speziellen Kontaktelemente.

2.4 Parallelisierung

Da die Lösung großer Gleichungssysteme ein rechenaufwendiges Prozedere darstellt, ist eine Parallelisierung dieses Vorgangs naheliegend. Die rechenintensiveren Schritte innerhalb der FE-Software sind a) Lösung des linearen Gleichungssystems, b) Aufstellen der Steifigkeitsmatrix. Die Parallelisierung von b) erweist sich als einfach, da die Steifigkeitsmatrizen aus unabhängigen Elementmatrizen aufgebaut werden und keinerlei Kommunikation (bis auf die einmalige Distribution der Elemente) notwendig ist, womit die zu erwartende Beschleunigung linear mit der Zahl der Prozessoren zunimmt. Die Beschleunigung beim iterativen Löser hingegen ist abhängig von der Besetzung der Matrix. Im Wesentlichen benutzt das CG Verfahren zur Lösungsfindung Matrix/Vektor-Multiplikationen und Skalarprodukte. Jeder Prozessor erhält einen zeilenweisen Ausschnitt der Matrix, bei Besetzung der Matrix an Positionen, deren Spaltenindex außerhalb des Zeilenbereiches des Prozessors liegt, muß das entsprechende Element des zu multiplizierenden Vektors von einem anderen Prozessor transferiert werden. Ein geschicktes Partitionieren der Matrix hat somit starken Einfluß auf den

Kommunikationsüberhang. Wir benutzen einen hierarchischen graphbasierten Ansatz ([9]) zur Partitionierung des Gesamtproblems, welcher den Kommunikationsüberhang (lokal) minimiert.

2.5 Technische Realisation

Da frei verfügbare Standardpakete zur FE-Analyse zum Großteil auf FORTRAN Implementationen beruhen, andererseits heutzutage mit modernen Methoden der Softwaretechnologie gearbeitet wird (oder werden sollte), wurde ein eigenes FEA-Werkzeug entwickelt ([8]). Die Konzeption der Simulationssoftware erfolgte in einem objektorientierten Prozeßmodell (OMT) unter Benutzung der UML-Notation. Bei der Konzeptionierung wurde besonderer Wert auf die Erweiterbarkeit aller Komponenten gelegt, die zusammen ein bibliotheksähnliches System bilden. Die Implementation erfolgte mittels C++ unter dem Betriebssystem Linux, die parallele Version der Software benutzt die `MPI` Bibliothek zur Kommunikation zwischen den Prozessoren.

3 Ergebnisse

Als Testplattform dienten neun Linux Rechner mit Intel-Celeron Prozessoren (400MHz) und jeweils 128MB Speicher. Die Rechner waren mit einem 100MBit/s Ethernet Netzwerk über einen Switch vernetzt. Das erstellte Kniegelenkmodell bestand aus 8540 Elementen und 10840 Knoten. Das Modell wurde am Femur fixiert und es wurde eine Kraftwirkung auf die Tibia im Bereich der *M. biceps femoris* Ansätze simuliert, welche das Kniegelenk in eine Flexionsbewegung versetzt. Die simulierte Bewegung im Kniegelenkmodell hinterließ makroskopisch einen validen Eindruck.

Einige Rechenzeiten und die erreichte Beschleunigung sind in Tabelle 1 aufgeführt. Eine theoretische Abschätzung der Beschleunigung ([8]) ergab bspw. für den Fall von acht Prozessoren eine untere Schranke von 4.51 und eine obere Schranke von 6.16 für die von uns verwendeten Umgebungs- und Modelldaten. Diese Abschätzung liefert für alle Konfigurationen valide Bereiche, womit sich für vorgegebene Probleme die Zahl der effektiv einzusetzenden Prozessoren im voraus ermitteln läßt. Die hohe Latenzzeit des Ethernet Netzes wirkte sich negativ auf die erzielte Beschleunigung aus, mit schnelleren Netzen (z.B. `SCI`) dürfte der Beschleunigungsfaktor besser sein. Die Parallelisierung stellt in diesem Falle ein geeignetes Mittel dar um die Rechenzeiten auf ein erträgliches Maß zu senken, die erreichte Beschleunigung spricht für sich. Auf Basis dieser Software ist es möglich, derartige Simulationen in Clustern von kostengünstigen Linux-Workstations zu rechnen anstatt direkt eine massiv-parallele Architektur einsetzen zu müssen (obgleich massiv-parallele Architekturen performanter sind, jedoch auch wesentlich teurer und nicht überall verfügbar).

# Prozessoren	Rechenzeit	Beschleunigung	Speicher/Knoten
1	3234.7s	1.00	361 MB
2	1848.9s	1.88	168 MB
4	1082.8s	3.39	100 MB
8	671.2s	5.59	51 MB

Tabelle 1. Rechenzeiten / Beschleunigungen (1 Zeitschritt)

4 Ausblick

Eine Erweiterung des Materialmodells auf nichtlineare anisotrope Materialien ist derzeit in Entwicklung, ebenso wie eine Erweiterung der Elementbibliothek. Mit diesen Materialmodellen kann man eine Analyse der Spannungen/Dehnungen in den beteiligten Strukturen erstellen um damit stark belastete Punkte an den Strukturen bei Fehlbelastungen oder auch bei Aufprällen zu identifizieren. Die Generierung des Hexaeder-Netzes aus einem gegebenen Oberflächennetz ist mit derzeitig verfügbarer Software nur mühsam zu realisieren und muß in Zukunft deutlich verbessert werden. Als Folge könnte die Detailtreue des erstellten Modells erhöht werden, um z.B. die *Menisci* korrekt als bewegliche, aufliegende Strukturen auf der Tibia zu modellieren. Der Einfluß verschiedener Lösungsverfahren (z.B. Multigrid Methoden) auf die Konvergenzgeschwindigkeit und Parallelisierbarkeit ist ebenfalls Gegenstand weiterer Untersuchungen.

Literatur

1. Garg, A., Walker, P.: *Prediction of Total Knee Motion Using a Three-Dimensional Computer-Graphics Model.* J. of Biomechanics, 23(1), 45-48, 1990.
2. Perie, D., Hobartho, M.: *In-vivo Determination of Contact Areas and Pressure of the Femorotibial Joint Using Non-Linear Finite Element Analysis.* Clinical Biomechanics, vol 13., 394-402, 1998.
3. Bendjaballah, M.: *Biomechanical Response of the Passive Human Knee Joint Under Anterior-Posterior Forces.* Clinical Biomechanics, vol. 13, 625-633, 1998.
4. *Knees-Up, HPCN enabled Simulation of the Human Knee (Public Final Report).* HPCN TTN Network, Nov. 1998.
5. Crisfield, M.: *Non-linear Finite Element Analysis of Solids and Structures.* John Wiley & Sons, Chichester, 1997, vol 1 & 2.
6. van den Bergen, G.: *Efficient Collision Detection of Complex Deformable Models using AABB Trees.* Tech. Bericht, TU Eindhoven, 1998.
7. Zienkiewicz, O.: *Methode der finiten Elemente.* McGraw-Hill, Maidenhead, 1977.
8. Wawro, M.: *An Object-Oriented Framework for the Simulation of Knee-Joint Biomechanics in Parallel Environments.* Diplomarbeit, Lehrstuhl Informatik VII, Universität Dortmund, 2000.
9. Karypis G., Kumar V.: *A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs.* Tech. Bericht, TR 95-033, Univ. of Minnesota, 1995.