

DISSERTATION

MODELLIERUNG VON KONTAKTEN KOMPLEX GEFORMTER KÖRPER IN DER MEHRKÖRPERDYNAMIK

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

O.Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Lugner
E 325

Institut für Mechanik und Mechatronik

eingereicht an der Technischen Universität Wien
Fakultät für Maschinenbau

von

Dipl.-Ing. Gerhard Hippmann
Matr.-Nr.: 9927414
Muldenstraße 14 d
82237 Steinebach in Bayern

Steinebach, im Oktober 2004

Kurzfassung

Die vorliegende Arbeit beschreibt Grundlagen, Modellierung, Implementierung und Anwendung eines Verfahrens zur Analyse von Kontaktvorgängen zwischen komplex geformten Körpern im Rahmen der Mehrkörperdynamik.

Nach einer kurzen Einführung werden einige Grundlagen der Simulation von Mehrkörpersystemen erläutert. Es folgt eine Übersicht der wichtigsten Methoden zur Kontaktanalyse, wobei zwischen Festkörper- und Oberflächenmodellen sowie eindimensionalen Modellen unterschieden wird.

Das im Rahmen dieser Arbeit entwickelte Polygonflächen-Kontaktmodell basiert auf Körperoberflächen in Polygondarstellung. Die ausführlich erläuterten Methoden und Algorithmen zur Kollisionserkennung sowie zur Approximation und Diskretisierung der Kontaktgebiete weisen daher eine enge Verwandtschaft zur Computergrafik auf. Zur Bestimmung der Kontaktspannungen wird auf die Kontaktelemente ein um viskose Dämpfung erweitertes Randschichtmodell sowie eine regularisierte Version des Coulombschen Reibgesetzes angewendet.

Vier Simulationsbeispiele unterschiedlicher Komplexität demonstrieren die Anwendung des Verfahrens. Mehrfache, mehrfach berandete und schiefere Kontakte werden robust und effizient ausgewertet. Die Arbeit schließt mit einer Bewertung der Ergebnisse und einem Ausblick.

Abstract

This thesis describes the fundamentals, modelling, implementation and application of a method for contact analysis between complexly shaped bodies in multibody dynamics.

After a brief introduction some basics of multibody system simulation are illustrated. An overview of the most important methods for contact analysis follows, whereby solid body, surface and one-dimensional models are differentiated.

The Polygonal Contact Model developed within the scope of this thesis is based on the polygonal representation of body surfaces. Therefore the comprehensively explained methods and algorithms for collision detection and contact patch approximation and discretisation are closely related to computer graphics. For determining the contact stresses the elastic foundation model, extended by viscous damping and a regularised version of Coulomb's friction law are utilised.

Four simulation examples of varying complexity present the application of the method. Multiple, multiply bordered and conforming contacts are analysed robustly and efficiently. The thesis closes with an evaluation of the results and an insight into further developments.

Inhaltsverzeichnis

1	Einführung	1
1.1	Mehrkörpersysteme	1
1.2	Kontaktmechanik	3
1.3	Motivation und Ziel	5
1.4	Grundlagen und Beitrag	5
1.5	Gliederung	7
2	Mehrkörpersystem–Simulation	9
2.1	MKS–Elemente	9
2.2	Bewegungsgleichungen	12
2.3	Numerische Integration	15
3	Kontaktmodelle	27
3.1	Grundlagen	27
3.1.1	Impenetrabilität	27
3.1.2	Adhäsionsfreiheit	28
3.1.3	Gegenwirkungsprinzip	29
3.1.4	Prinzip von St. Venant	29
3.1.5	Reibung	30
3.2	Schwierigkeiten	31
3.2.1	Kontakterkennung	31
3.2.2	Kontaktgebiet	32
3.2.3	Partielles Gleiten	32
3.2.4	Materialdämpfung	32
3.3	Festkörpermodelle	33
3.3.1	Elastizitätstheorie	33
3.3.2	Finite–Elemente–Methode	35
3.4	Oberflächenmodelle	37
3.4.1	Halbraumapproximation	37
3.4.2	Randschichtmodell	41
3.5	Eindimensionale Modelle	44

3.5.1	Kraftelemente	44
3.5.2	Starrkörperkontakt	46
3.5.3	Kontaktkinematik	48
3.6	Gegenüberstellung und Bewertung	52
4	Polygonflächen–Kontaktmodell	55
4.1	Kollisionserkennung	60
4.1.1	Anforderungen	60
4.1.2	Elementare Analyse	61
4.1.3	Hüllvolumen–Hierarchie	64
4.2	Konstruktion der Schnittvolumina	73
4.2.1	Konstruktion der Schnittpolygone	73
4.2.2	Bestimmung der aktiven Flächenelemente	77
4.3	Bestimmung der Kontaktelemente	84
4.3.1	Master–Slave–Methode	85
4.3.2	Variational–Interpolation–Methode	89
4.4	Bestimmung der Kontaktkraft	94
4.4.1	Bestimmung der Normalkraft	95
4.4.2	Bestimmung der Tangentialkraft	96
5	Simulationsergebnisse	99
5.1	Plausibilität	99
5.2	Vergleich zum Hertzschen Modell	103
5.3	Bouncing Bubbles	109
5.4	Fußgänger–Pkw–Unfall	125
6	Zusammenfassung	139
6.1	Inhalt	139
6.2	Bewertung und Einordnung	143
6.3	Ausblick	145
A	Implementierung	149
A.1	Schnittstellen	149
A.2	Struktur	154
	Literaturverzeichnis	159

Kapitel 1

Einführung

In vielen technischen Systemen spielen Kontaktvorgänge eine wichtige Rolle. Um deren dynamisches Verhalten zu untersuchen, erweist sich häufig die Mehrkörperdynamik–Simulation als vorteilhaft. Dabei wird vor allem auf zwei Disziplinen aus der technischen Mechanik zurückgegriffen: Einerseits auf die Mechanik der *Mehrkörpersysteme* (MKS), andererseits auf die *Kontaktmechanik*. In den beiden folgenden Abschnitten werden beide Begriffe kurz erläutert. Weiter werden Motivation und Aufgabenstellung der vorliegenden Arbeit vorgestellt, und ihr Beitrag anhand einer Übersicht der wichtigsten zugrundeliegenden und themenverwandten Veröffentlichungen herausgearbeitet. Die Einführung schließt mit einer kurzen Beschreibung der folgenden Kapitel.

1.1 Mehrkörpersysteme

Die numerische MKS–Simulation hat sich als bedeutende Disziplin beim rechnergestützten Entwurf technischer Systeme (engl. Computer Aided Engineering, CAE) etabliert. Sie stellt neben Finiten–Elemente–Methode (FEM), computerbasierter Konstruktion (engl. Computer Aided Design, CAD) und computergestütztem Reglerentwurf (engl. Computer Aided Control Engineering, CACE) ein wesentliches Glied im modernen Entwicklungsprozess nach den Prinzipien des *Concurrent Engineering* dar [1]. Die hohe Schlagkraft der MKS–Simulation rührt insbesondere von ihrer guten Effizienz her; mit minimalen Informationen über ein mechanisches System lassen sich bereits im Konzeptstadium, lange vor dem Bau von Prototypen, ausgiebige Parameterstudien durchführen (*Virtual Prototyping*). Darüber hinaus sind Modelldarstellung und Auswertungsmethoden etablierter MKS–Programme auf die Bedürfnisse von Berechnungsingenieuren zugeschnitten, und für be-

deutende Spezialanwendungen wie Straßen- oder Schienenfahrzeuge [2] stehen ausgereifte Bibliotheken und Module zur Verfügung [3].

Als zentrales Element eines Programms für die Mehrkörperdynamik-Simulation dient der MKS-Formalismus zur automatischen Erstellung der Bewegungsgleichungen. Durch sie wird es möglich, anhand entsprechender numerischer Verfahren das kinematische und dynamische Verhalten im Zeitbereich zu berechnen, statische Gleichgewichtszustände bzw. nominelle Schnittkräfte zu ermitteln oder nach Linearisierung Eigenwertanalysen und Stabilitätsuntersuchungen im Frequenzbereich durchzuführen. Ferner lassen sich die genannten Methoden automatisiert für Parameterstudien sowie zur Optimierung und Parameteridentifikation eines MKS einsetzen.

Die MKS-Simulation unterscheidet sich in einigen wesentlichen Punkten von der noch weitaus stärker verbreiteten FEM. Hier wird typischerweise nicht die Statik eines Einzelkörpers, sondern die Dynamik aus dem Zusammenspiel mehrerer Körper betrachtet. Neben der vollständigen Berücksichtigung aller Trägheitswirkungen zählt der Umgang mit Nichtlinearitäten infolge beliebig großer Verdrehungen zu den Stärken der Mehrkörperdynamik. Mit selten mehr als 1000 Freiheitsgraden weisen MKS-Modelle um zwei bis drei Größenordnungen weniger Zustandsgrößen auf als FEM-Modelle. Die Rechenzeiten auf aktuellen Workstations bewegen sich entsprechend im Sekunden- oder Minutenbereich, während FEM-Analysen durchaus mehrere Stunden dauern können.

In der klassischen MKS-Theorie wird das betrachtete mechanische System aus starren Körpern modelliert, deren Bewegung durch *Gelenke* eingeschränkt und durch *Kraftelemente* beeinflusst wird. In die Bewegungsgleichungen gehen nur ihre Trägheitseigenschaften und die Lage der *Koppelpunkte* (der Gelenke und Kraftelemente) ein. Die Gestalt der Körper spielt dabei keine Rolle, sodass sie sich im Rahmen ihrer Gelenkfreiheitsgrade beliebig bewegen können. Insbesondere werden Berührungsvorgänge nicht abgebildet, d. h. eine Durchdringung von Körpern des realen Systems bleibt in der Simulation wirkungslos.

In zahlreiche Anwendungen sind Berührungen der Körper jedoch essentiell für das Systemverhalten. Beispielsweise bestimmen sie ganz wesentlich die Eigenschaften von Wälzlagern [4] (Abbildung 1.1) und Kurvengetrieben. Auch bei der Analyse von greifenden Robotern [5] und Laufmaschinen [6] spielen sie eine zentrale Rolle. Ein grundlegendes Problem bei der MKS-Simulation von Fahrzeugen ist die Darstellung der Traktion zwischen Rädern und Lauffläche [7, 8]. Und auch in der Biomechanik treten vermehrt Berührungen auf, zum Beispiel in Gelenken [9] oder bei der Simulation von Unfällen [10]. Für diese und viele weitere Anwendungen werden im MKS-Modell Elemente benötigt, welche die *Kontaktmechanik* geeignet darstellen.

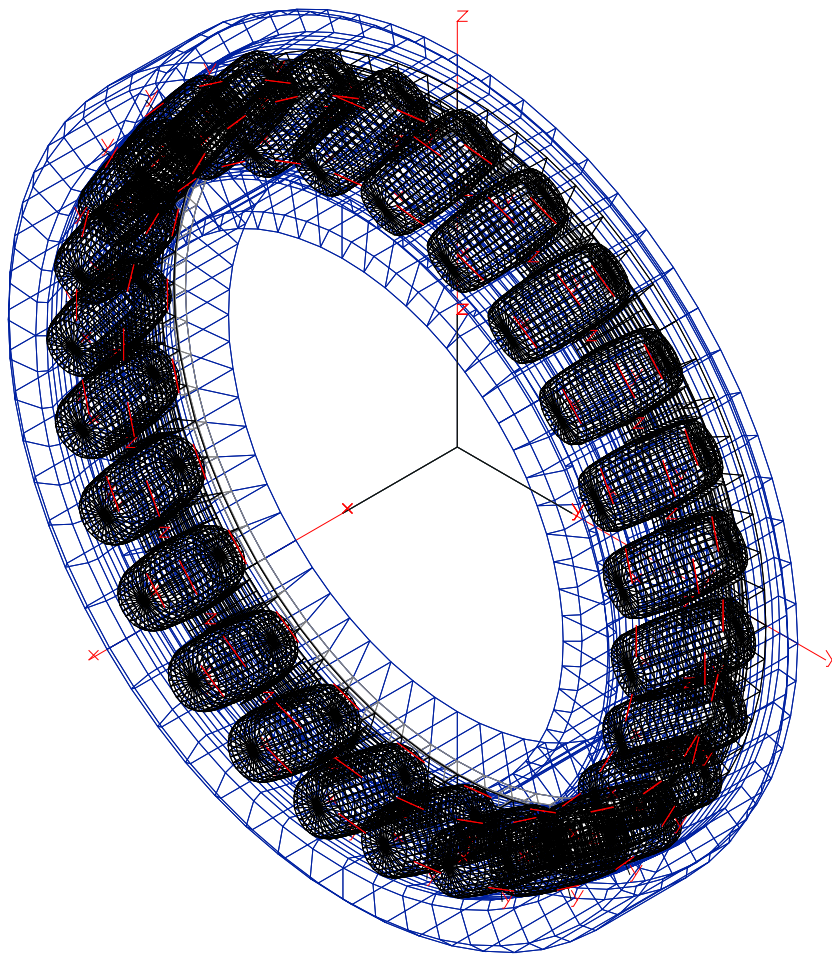


Abbildung 1.1: MKS-Modell einer Pendelrollenlagerhälfte

1.2 Kontaktmechanik

Die Kontaktmechanik versucht, Berührungsvorgänge zwischen festen Körpern zu beschreiben. Gegenwärtig konzentriert sich die Forschung auf diesem Gebiet auf experimentelle Arbeiten, die Entwicklung von *Kontaktmodellen* und die Implementierung dieser Modelle für die numerische Simulation.

Zum Thema Kontaktmechanik gibt es eine kaum überblickbare Zahl von Veröffentlichungen. Daneben kommt das ungebrochen große Interesse an dieser Disziplin durch zahlreiche Workshops und Tagungsreihen zum Ausdruck, die sich alleine oder schwerpunktmäßig mit Kontaktproblemen beschäftigen. Bei genauerer Betrachtung wird allerdings deutlich, dass der Begriff Kontaktmechanik häufig physikalische Phänomene beschreibt, die wenig mit den in

dieser Arbeit untersuchten Berührungsvorgängen in MKS gemein haben. So spielen hier etwa elektrische Kontakte, Granulatdynamik und Bruchmechanik keine Rolle.

Ein dynamischer Kontaktvorgang kann global als Prozess betrachtet werden, bei dem kinetische Energie umgewandelt wird in dissipative Energien wie Wärme, Schall und plastische Verformung, sowie wiederum kinetische Energie. Für ein MKS ist dabei von entscheidender Bedeutung, welchen Einfluss die Kontaktkräfte auf die Systemdynamik ausüben. Neben den Kräften ist insbesondere der kinematische Zustand des MKS nach dem Kontakt von Interesse. Dagegen brauchen innere Spannungen und Verformungen der sich berührenden Körper, thermodynamische Prozesse, Tribologie, Oberflächenrauheit und Verschleiß nicht näher betrachtet zu werden. Auch hydrodynamische Effekte, die bei sehr hohen Kollisionsgeschwindigkeiten auftreten, bleiben in der Regel unberücksichtigt.

Einige Begriffe der Kontaktmechanik werden in der Literatur nicht völlig einheitlich verwendet. In dieser Arbeit wird unter *Kontakt* eine Berührung der Oberflächen zweier Festkörper verstanden. Dagegen beschreibt *Separation* deren räumliche Trennung. Ein Berührungsvorgang, bei dem die Körper vom separierten zum kontaktierenden Zustand wechseln, wird *Kollision* genannt. Außerdem werden dynamische Kontaktvorgänge in Starrkörpermodellen, bei denen es zum Impulsaustausch kommt, als *Stoß* bezeichnet.

Die im Weiteren relevanten Kontaktmodelle lassen sich auf zwei verschiedene Ansätze zurückführen: Einerseits werden Verformungen und Spannungen der elastisch angenommenen Körper modelliert. Andererseits geht man — wie im klassischen MKS — von starren Körpern aus, und dynamisches Kontaktverhalten wird durch hypothetische Stoßgesetze angenähert. Zur Berücksichtigung der für viele Kontaktprobleme wesentlichen trockenen Reibung wird fast ausschließlich das Coulombsche Reibgesetz zugrunde gelegt, wenn auch häufig in regularisierter Form. Kapitel 3 befasst sich ausführlich mit den Grundlagen der Kontaktmechanik und Kontaktmodellen der beiden genannten Klassen.

Zur Implementierung eines Kontaktmodells in ein Simulationsprogramm sind viele weitere Probleme zu lösen. Effizienz und Robustheit der numerischen Simulation hängen ganz wesentlich von effektiver Datenhaltung und korrekter Numerik des Kontaktalgorithmus ab. Außerdem muss konkret entschieden werden, wie die für die Kontaktanalyse essentielle Geometrie der Körper beschrieben wird. Dabei können technisch bedingte Einschränkungen oder Verkomplizierungen die Anwendbarkeit des Programms erheblich beeinträchtigen. Schließlich ist auch auf zweckmäßige Gestaltung von Parametrierung und Schnittstellen zu achten.

1.3 Motivation und Ziel

Obwohl die Kontaktmechanik als hoch entwickelte Disziplin gelten darf, zeigen sich gängige MKS-Programme bei der Analyse von Systemen mit Berührungsvorgängen sehr eingeschränkt. So findet man einerseits ausgesprochen genaue und effiziente Methoden, die allerdings auf ganz spezielle Anwendungen beschränkt sind, wie z. B. den Rad-Schiene Kontakt. Andererseits bilden allgemein gehaltene Modelle den Kontakt in so hohem Maße vereinfacht ab, dass die Ergebnisse der Simulation häufig zu deutlich von der Realität abweichen. In der Praxis erweist es sich außerdem als sehr hinderlich, dass viele Verfahren nur bestimmte Oberflächengeometrien der Körper verarbeiten können, beispielsweise primitive Formen wie Ebenen, Zylinder und Ellipsoide oder aber ausschließlich konvexe Flächen.

Es ist das Ziel dieser Arbeit, einen allgemeinen MKS-Kontaktalgorithmus, welcher bei MKS-typischer Effizienz und Abbildungsqualität mehr Flexibilität als bekannte Lösungen bietet, zu entwickeln, zu implementieren und testweise anzuwenden.

Hierbei bezieht sich die Allgemeinheit auf den Anwendungsbereich: Gesucht ist ein Verfahren, mit dem sich möglichst viele verschiedenartige Kontaktprobleme lösen lassen. Mit der mangelnden Flexibilität verfügbarer Methoden wird insbesondere auf deren Einschränkungen hinsichtlich der Oberflächengeometrie eingegangen. Ziel ist es, einen Algorithmus zu implementieren, der mit beliebig geformten Körpern umgehen und auch schmiegende Berührungen sowie Mehrpunktkontakt verarbeiten kann. Der allgegenwärtige Zielkonflikt zwischen Effizienz und Genauigkeit der Simulation soll so gelöst werden, dass die Rechenzeiten unter Berücksichtigung von Modellkomplexität und -umfang eindeutig im MKS-üblichen Rahmen liegen, wobei eine möglichst gute Approximation realer Kontaktvorgänge erzielt wird.

1.4 Grundlagen und Beitrag

An dieser Stelle kann und soll keine umfassende Literaturübersicht zur Mehrkörperdynamik und Kontaktmechanik stehen. Es werden jedoch einige Veröffentlichungen zusammengestellt, die für die vorliegende Arbeit von besonderer Bedeutung sind, weil sie grundlegende Erkenntnisse und Methoden liefern oder eine sehr ähnliche Problemstellung behandeln.

Die frühesten Beiträge zur Kontaktmechanik lieferten Newton (1643–1727) und Poisson (1781–1840) mit ihren Stoßhypothesen sowie Coulomb (1736–1806) mit seinem Reibgesetz. Ihre Erkenntnisse sind bis heute von größter Bedeutung: Die Stoßgesetze beschreiben Kontaktvorgänge unter der

Annahme starrer Körper und finden daher insbesondere in der Mehrkörperdynamik Anwendung. Neuere Arbeiten aus diesem Blickwinkel verfassten etwa Glocker [11] und Klisch [12]. Und obwohl das Coulombsche Reibmodell ein hochkomplexes Phänomen, das noch immer nicht vollständig verstanden ist, in einfachster Weise beschreibt, konnte sich bisher keine geeignetere Modellierung trockener Reibung durchsetzen.

Auch das 1882 veröffentlichte Kontaktmodell nach Hertz [13], bei dem die berührenden Körper durch linear-elastische Halbräume approximiert werden, spielt bis in die Gegenwart eine überaus wichtige Rolle. Die für reibungsfreie Oberflächen zweiter Ordnung formulierten Gesetzmäßigkeiten wurden im Laufe des 20. Jahrhunderts u. a. von Love [14], Panagiotopoulos [15] und Kalker [16] der numerischen Behandlung zugänglich gemacht. Damit stehen Methoden zur Verfügung, um Kontakte zwischen beliebig geformten glatten Körpern unter Berücksichtigung Coulombscher Reibung zu analysieren. In [17] wird ein weiter vereinfachtes Oberflächenmodell für die effiziente Berechnung des Rad-Schiene-Kontakts verwendet, bei dem von starren Körpern ausgegangen wird, die mit einer elastischen Randschicht überzogen sind.

Die meisten formulierten und implementierten Kontaktalgorithmen zur Beschreibung von Kontakten in der Mehrkörperdynamik basieren auf der Annahme einer elastischen Randschicht. Häufig wird die Berührung nur in einem angenommenen Kontaktpunkt betrachtet, wo einseitige Feder-Dämpfer Elemente in normaler und regularisierte Coulombsche Reibelemente in tangentialer Richtung wirken [3, 18, 19, 20]. Andere Verfahren arbeiten mit einer Diskretisierung der Berührfläche [8, 21, 22, 23, 24]. Außerdem lassen mittlerweile die schnellen Fortschritte in der Computertechnik FEM-Kontaktanalyse während der MKS-Simulation interessant erscheinen [25].

Ein grundlegendes Charakteristikum von Modellen für komplex geformte Körper stellt die Repräsentation der Oberflächen dar. Der Algorithmus in [19] ermöglicht es, mit Flächen zu arbeiten, die aus Geraden und Kreisbögen zusammengesetzte, ebene Kurven überstreichen, wenn sie entlang einer Geraden bewegt werden. Die auf bewegten Markern basierende Methode aus [3, 18] kann Spline-interpolierte oder -approximierte sowie analytisch beschriebene Flächendarstellungen behandeln. In der Praxis erweisen sich beide Verfahren jedoch nur für bestimmte Anwendungen geeignet, sodass in jüngerer Zeit nach flexibleren Möglichkeiten gesucht wurde. So wird in [20] mit Freiformflächen gearbeitet, die durch die bekannte NURBS (Non-Uniform Rational B-Splines) Darstellung approximiert werden. Außerdem wurden einige Verfahren entwickelt, bei denen Polygonflächen Anwendung finden [21, 22, 23, 24, 26].

Die vorliegende Arbeit stellt einen neuen Kontaktalgorithmus vor, der auf die Anwendung in der MKS-Simulation zugeschnitten ist (s. Kapitel 4).

Die Körperoberflächen beliebiger Form werden durch Polygonflächen dargestellt, mit denen üblicherweise auch in der Computergrafik gearbeitet wird, sodass sie starke Verbreitung und Unterstützung finden. Die neue Methode hat darüber hinaus den Vorteil, dass die Kontaktnormalkräfte nach dem (im Rahmen einer Diskretisierung) vergleichsweise hochwertigen Randschichtmodell ermittelt werden. Die zugehörige Implementierung ist mit allgemeinen Schnittstellen zur Anbindung an beliebige MKS-Programme ausgeführt und steht als offener Quellcode zur Verfügung.

1.5 Gliederung

Das folgende Kapitel 2 erläutert kurz die Methode der numerischen MKS-Simulation. Ausführlicher wird auf Eigenschaften und die numerische Lösung der Bewegungsgleichungen eingegangen.

Im Kapitel 3 werden zuerst Grundlagen und Schwierigkeiten der Kontaktmechanik zusammengestellt. Es folgt ein detaillierter Überblick bekannter Methoden zur Kontaktanalyse. Anhand einer Einteilung in verschiedene Modelltypen und der Darstellung ihrer zugrundeliegenden Ideen und Verfahren wird ein Vergleich in Hinblick auf ihre Eignung für die MKS-Simulation angestellt.

Kapitel 4 beschreibt Ideen, Algorithmen und Implementierung eines neuartigen *Polygonflächen*-Kontaktmodells für die MKS-Simulation. Zunächst werden die geometrischen Probleme der Kollisionserkennung und -synthese, dann die Bestimmung der Kontaktkräfte behandelt.

Kapitel 5 zeigt einige Anwendungen des Polygonflächen-Kontaktmodells. In vier Simulationsbeispielen werden ausführlich Praxistauglichkeit, Robustheit und Effizienz des Verfahrens und seiner Implementierung untersucht.

Schließlich fasst Kapitel 6 die Arbeit in kurzer Form zusammen, formuliert Stärken und Schwächen des neuen Kontaktmodells und gibt einen Ausblick auf mögliche Weiterentwicklungen.

Außerdem beschreibt Anhang A Schnittstellen und Struktur der im Rahmen dieser Arbeit erstellten Software-Implementierung.

Kapitel 2

Mehrkörpersystem–Simulation

In diesem Kapitel wird in knapper Form auf einige Grundlagen und Aspekte der MKS–Simulation eingegangen, welche für die Simulation von Kontaktvorgängen von Bedeutung sind. Nach einer Beschreibung der wichtigsten MKS–Elemente werden Struktur und rechnergestützte Ermittlung der Bewegungsgleichungen behandelt. Weiter werden wichtige Begriffe und Verfahren zu deren numerischer Lösung im Zeitbereich vorgestellt.

2.1 MKS–Elemente

Um ein technisches System mit Hilfe der MKS–Simulation zu analysieren, muss zunächst ein entsprechendes Mehrkörpermodell entwickelt werden, welches die interessierenden dynamischen Effekte wiedergeben kann. Dabei werden klassische MKS, welche für die Betrachtungen in dieser Arbeit ausreichend sind, aus nur drei Elementklassen gebildet (Abbildung 2.1):

Körper werden idealisiert vollkommen starr angenommen, sodass Schwerpunkt, Masse und Trägheitstensor ihre Trägheitseigenschaften vollständig beschreiben. *Gelenke* dienen zur (masselosen und reibungsfreien) kinematischen Kopplung zweier Körper, welche deren Bewegungsfreiheit relativ zueinander einschränken. Ebenso wirken *Kraftelemente* zwischen Körpern, wobei sie eingeprägte Kräfte und Momente in Abhängigkeit derer kinematischer Konfiguration („*Messungen*“) bewirken. In den einfachsten Fällen handelt es sich um Federn oder Dämpfer, es lassen sich jedoch auch komplexe Subsysteme wie komplette Regelkreise, Reifen oder Kontaktmodelle durch Kraftelemente beschreiben.

Ferner verwendet man *Marker*, d. h. Koordinatensysteme mit körperfester Einbaulage und -orientierung, als Koppelemente für Gelenke und Kraftelemente. Als Bezugsbasis wird von einem *Inertialsystem* ausgegangen,

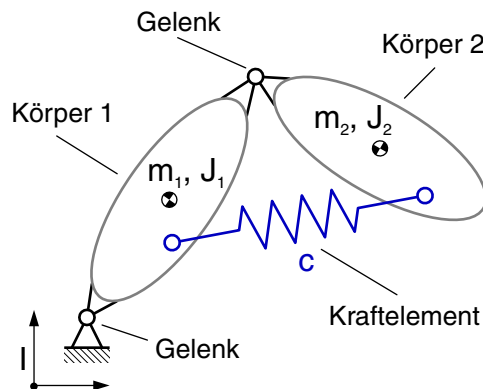


Abbildung 2.1: Elementklassen klassischer Mehrkörpersysteme

welches meist einem erdbodenfesten Punkt entspricht. Durch die Erdgravitation oder andere physikalische Effekte bewirkte *Volumenkräfte* können z. B. durch Kraftelemente beschrieben werden.

Auch Kontaktprobleme werden i. d. R. anhand entsprechend spezialisierter MKS-Elemente modelliert. Während beim Starrkörperkontakt Gelenke Anwendung finden, beschreibt man Kontakte mit elastischer Verformung durch Kraftelemente (s. Abschnitt 3.5). Ferner sind für die Analyse von Kontakten je nach Modellierung mitunter erhebliche Eingriffe in die Numerik der MKS-Simulation erforderlich, die von einfacher Begrenzung der Zeitschrittweiten bis zu Stoßdetektierung mittels Schaltfunktionen und anschließender Berechnung eines konsistenten Systemzustands (s. Abschnitt 3.5.2) reichen können.

MKS-Programme verfügen meist über umfangreiche Bibliotheken von Gelenken und Kraftelementen, um verschiedenartige technische Systeme modellieren zu können. Während die Implementierung von Gelenken i. d. R. recht schematisch verläuft, decken Kraftelemente ein ausgesprochen großes Anwendungsspektrum ab, sodass völlig unterschiedliche Ansätze und Strategien realisiert werden. In der Praxis handelt es sich daher bei benutzerimplementierten Bibliothekselementen fast ausschließlich um Kraftelemente. Auch das in den Kapiteln 4 und 5 vorgestellte Kontaktmodell stellt formal ein Kraftelement dar. Deshalb wird im Folgenden näher auf diesen MKS-Elementtyp eingegangen.

Gewöhnliche Kraftelemente wirken zwischen zwei Koppelmarkern, die auf verschiedenen Körpern oder inertial fest liegen können, in Abhängigkeit relativkinematischer Messungen der Koppelmarker. In der Praxis werden überwiegend zwei verschiedene Typen angewendet: Bei sog. *Point-to-Point-*

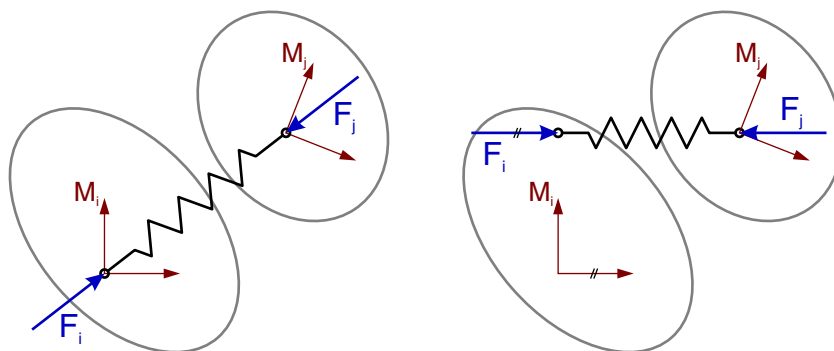


Abbildung 2.2: Point-to-Point- (links) und Component-Kraftelement bei gleicher Anordnung der Koppelmarker M_i , M_j

Elementen stimmt die Wirkrichtung mit der Verbindungslinie zwischen den Koppelmarkern überein. *Component*-Elemente sind dagegen prinzipiell unsymmetrisch, da ihre Wirkrichtung durch die Orientierung einer der beiden Koppelmarker bestimmt und der Kraftangriffspunkt am anderen Koppelmarker angenommen wird (Abbildung 2.2).

Bei der Bestimmung der kinematischen Eingangsgrößen des Kraftgesetzes ist zu beachten, dass sich aus den MKS-Zustandsgrößen nur Messungen auf Lage- und Geschwindigkeitsebene ableiten lassen. Für manche Anwendungen (z. B. Fahrermodelle) werden jedoch auch Beschleunigungen benötigt, welche erst nach vollständiger Auswertung der Systemfunktion (einschließlich aller Kraftelemente) bekannt sind. Die gleiche Problematik bringen Kraftelemente mit sich, die von Zwangskräften abhängen (z. B. Gelenkreibung). In beiden Fällen muss im Prinzip auf Formulierung als implizite Zustandsgrößen zurückgegriffen werden, die durch algebraische Nebenbedingungen bestimmt werden [3].

In den meisten Fällen beschreiben Kraftelemente technische Systeme, die zwischen zwei Körpern wirken. Als Messungen sind deshalb rein relative kinematische Größen relevant, die nicht von der gemeinsamen Führbewegung der gekoppelten Körper abhängen. Dies ist insbesondere bei der Bestimmung translatorischer Geschwindigkeiten und Beschleunigungen zu beachten [27].

Während Massenträgheiten den Körpern vorbehalten sind, werden Kraftelemente grundsätzlich masselos angenommen. Als Konsequenz muss die vektorielle Summe der eingprägten Kräfte und Momente jedes Kraftelements verschwinden. Werden passive Kraftgesetze an einem bestimmten Punkt zur Wirkung gebracht, ist ferner darauf zu achten, dass Ort und Richtung der Messungen mit denen der resultierenden Kräfte bzw. Momente übereinstimmen.

Wie im Abschnitt 2.3 erläutert wird, führen Konstellationen aus Körpern mit geringer Trägheit und Krafterelementen mit hoher Steifigkeit und/oder Dämpfung zu steifen Bewegungsgleichungen, sodass die Effizienz oder sogar Stabilität der numerischen Integration erheblich beeinträchtigt werden kann. Deshalb sollten solche Situationen möglichst schon während der Modellierung vermieden werden.

Bei der Implementierung von Krafterelementen ist zu beachten, dass im Zeitbereich nicht von streng monotoner Auswertung ausgegangen werden kann. Dies lässt sich anhand der im Abschnitt 2.3 beschriebenen Funktionsweise des Integrators DASSL nachvollziehen. Folglich macht es keinen Sinn, auf lokal gespeicherte Größen aus dem letzten Aufruf eines Krafterelements zuzugreifen, um beispielsweise das Maximum einer Funktion zu bestimmen; denn der vom Integrator vorgegebene Systemzustand ist oft Gegenstand einer Iteration und somit für das letztlich berechnete Simulationsergebnis irrelevant.

2.2 Bewegungsgleichungen

Um ein MKS rechnerisch zu untersuchen, muss es in ein entsprechendes mathematisches Modell überführt werden. Für überschaubare Systeme kann der Ingenieur diesen Schritt mit Papier und Bleistift erledigen, die meisten technischen Anwendungen überschreiten jedoch aufgrund ihrer Komplexität den dafür vertretbaren Aufwand bei Weitem. Deshalb werden die Bewegungsgleichungen, welche das dynamische Verhalten eines MKS beschreiben, in der Praxis normalerweise mittels Computerprogrammen generiert.

Die dabei verwendeten Algorithmen werden als *MKS-Formalismen* bezeichnet und unterscheiden sich teilweise erheblich hinsichtlich verschiedener Merkmale, von denen hier einige erwähnt werden sollen:

Während *symbolische* Formalismen die Bewegungsgleichungen ebenso wie ein menschlicher Bearbeiter in analytischer Form darzustellen vermögen, liefern *numerische* Verfahren lediglich ihre Auswertung für einen gegebenen Systemzustand. Erstere erlauben also auch Schlüsse über Kopplungen und strukturelle Zusammenhänge, führen jedoch bei umfangreichen Systemen mit Unstetigkeiten zu kaum mehr beherrschbarem Verwaltungs- und Handhabungsaufwand.

Großen Einfluss auf den Umfang und die Eigenschaften des resultierenden Gleichungssystems hat die Wahl der Systemvariablen. Ein weit verbreiteter Ansatz beschreibt die Bewegung der Körper durch *Absolutkoordinaten*, d. h. jeweils sechs Variablen, die Position und Orientierung gegenüber dem Inertialsystem angeben. Kinematische Bindungen durch Gelenke werden

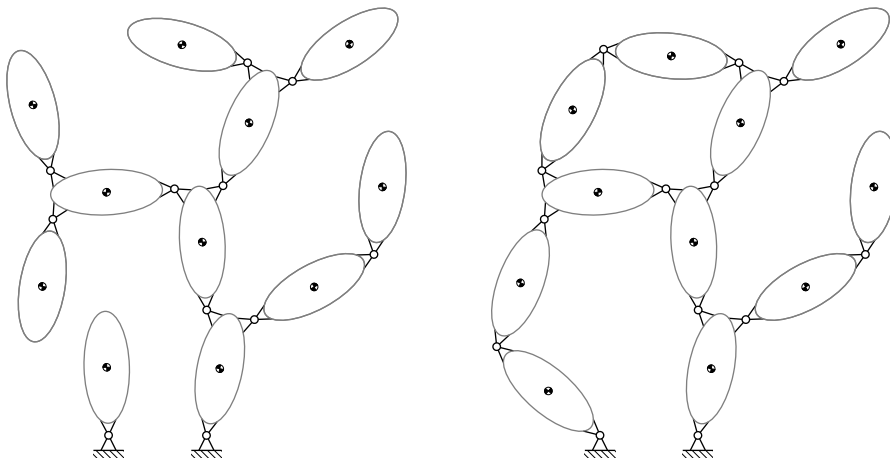


Abbildung 2.3: MKS in Baumstruktur (links) und mit kinematischen Schleifen, die durch Verkettung mit Gelenken gebildet werden

durch Zwangsbedingungen berücksichtigt, welche in entsprechenden Zwangskräften und -momenten resultieren [28].

Andererseits lässt sich der Zustand eines MKS auch durch *generalisierte Koordinaten* darstellen. Dabei beschreiben die Systemvariablen meist die relativen kinematischen Freiheitsgrade der Gelenke. Enthält das System allerdings geschlossene kinematische Schleifen (Abbildung 2.3), stellt die automatische Generierung der Bewegungsgleichungen i. Allg. ein schwieriges Problem dar [29]. Deshalb wird in der Praxis mit *Relativkoordinaten in Deskriptorform* gearbeitet, wobei das MKS durch baumstrukturierte kinematische Ketten beschrieben wird, welche durch schleifenschließende Gelenke verbunden werden können. Letztere ergeben analog zur Formulierung in Absolutkoordinaten kinematische Zwangsbedingungen, die für zusätzliche Zwangskräfte und -momente in den dynamischen Bewegungsgleichungen sorgen [30].

Im Wesentlichen ist die Darstellung in Absolutkoordinaten einfacher zu handhaben und führt auf vergleichsweise große differential-algebraische Gleichungssysteme, deren einfache Struktur sich bei der numerischen Lösung gut ausnützen lässt [31]. Dagegen gestaltet sich der Umgang mit in Deskriptorform formulierten Systemen anspruchsvoller, wenn kinematische Schleifen (Abbildung 2.3) vorliegen. Das resultierende Gleichungssystem weist aber in der Regel erheblich weniger Zustandsgrößen und algebraische Nebenbedingungen auf; MKS mit kinematischer Baumstruktur ergeben sogar nur ein gewöhnliches Differentialgleichungssystem. Eine ausführliche Diskussion der Koordinatenwahl für MKS findet man z. B. in [3].

Gelenke bestimmen neben der durch ihre Anordnung gegebenen Topologie auch durch die Art ihrer kinematischen Bindungen wichtige Eigenschaften des betrachteten MKS [32, 33]. Gehen ausschließlich Zustandsgrößen auf Lageebene ein, spricht man von *holonomen* Bindungen. Zu dieser Klasse zählen Standardgelenktypen wie z. B. Schiebe-, Dreh-, Kardan- und Kugelgelenke. Gelenke mit *nichtholonomen* Bindungen sind darüber hinaus durch (nicht mittels Integration in holonome Bindungen überführbare) Zwangsbedingungen auf Geschwindigkeitsebene gekennzeichnet. Diese Klasse tritt in der Praxis eher selten auf, beispielsweise bei Kontaktproblemen mit idealisiert modellierter Coulombscher Haftreibung (s. Abschnitt 3.5.2). Weiter nennt man Bindungen, bei denen explizit die Systemzeit eingeht, *rheonom*, während sie sonst als *skleronom* bezeichnet werden.

Die Bewegungen der einzelnen Starrkörper eines (kinetisch bestimmten) MKS sind durch die Newtonschen und Eulerschen Gleichungen bestimmt. Eine weitere Unterscheidung von MKS-Formalismen ist durch das zugrundeliegende *Prinzip der Mechanik* [32, 33, 34] gegeben, welches die Anwendung der kinetischen Grundgleichungen auf kinematisch gebundene MKS ermöglicht. Dabei wird in den meisten Fällen vom *D’Alembertschen Prinzip* oder den *Lagrangeschen Gleichungen zweiter Art* Gebrauch gemacht. Während ersteres die Bewegungsänderung der freigeschnittenen Körper als Wirkung der angreifenden Kräfte und Momente beschreibt, gehen in letztere die Ableitungen der kinetischen und potentiellen Energie nach den generalisierten Lage- und Geschwindigkeitskoordinaten ein (wobei sie nur für holonome Systeme anwendbar sind).

Als Grundlage für die in den Kapiteln 4 und 5 beschriebenen Implementierungen und Simulationen wurde das kommerzielle MKS-Programmpaket SIMPACK [3, 18] verwendet. Dieses basiert auf einem Formalismus mit Freiheitsgradproportionalem Rechenaufwand ($\mathcal{O}(n)$ -Formalismus) [35, 36], der mit generalisierten Koordinaten nach dem D’Alembertsche Prinzip arbeitet.

Unabhängig vom angewendeten mechanischen Prinzip ergibt sich i. Allg. ein differential-algebraisches Gleichungssystem erster Ordnung mit folgender Struktur [37]:

$$\dot{\mathbf{p}} = T(\mathbf{p})\mathbf{v} \quad (2.1a)$$

$$M(\mathbf{p})\dot{\mathbf{v}} = \mathbf{f}(\mathbf{p}, \mathbf{v}) + G^T(\mathbf{p})\lambda \quad (2.1b)$$

$$\mathbf{g}(\mathbf{p}) = \mathbf{0} \quad (2.1c)$$

Hierbei stellen die Lagekoordinaten $\mathbf{p} \in \mathbb{R}^{n_p}$, die Geschwindigkeiten $\mathbf{v} \in \mathbb{R}^{n_v}$ und die Lagrangeschen Multiplikatoren $\lambda \in \mathbb{R}^{n_\lambda}$ die Systemvariablen dar. \mathbf{p} und \mathbf{v} werden über die Koeffizientenmatrix $T \in \mathbb{R}^{n_p \times n_v}$, welche häufig der

Einheitsmatrix entspricht, verknüpft. Die generalisierte Massenmatrix $M \in \mathbb{R}^{n_v \times n_v}$ enthält die wirksamen Massen und Trägheitsmomente der Körper, $\mathbf{f} \in \mathbb{R}^{n_v}$ die generalisierten eingepprägten Kräfte sowie Zentrifugal- und Corioliskräfte.

Die Terme $G^T(\mathbf{p})\lambda$ und $\mathbf{g}(\mathbf{p})$ treten nur auf, wenn Gelenke durch Zwangsbedingungen beschrieben werden. Dabei enthält $\mathbf{g}(\mathbf{p}) \in \mathbb{R}^{n_\lambda}$ die Zwangsbedingungen auf Lageebene, und die Transponierte der Zwangsmatrix $G(\mathbf{p}) \in \mathbb{R}^{n_\lambda \times n_v}$ mit

$$G(\mathbf{p}) = \frac{\partial \mathbf{g}(\mathbf{p})}{\partial \mathbf{p}} T(\mathbf{p}) \quad (2.2)$$

projiziert den Vektor der Lagrangeschen Multiplikatoren λ auf die Gelenkzwangskräfte.

2.3 Numerische Integration

Die am häufigsten angewendete Analyseverfahren für MKS ist die Zeitintegration: Für einen gegebenen Anfangszustand werden die Bewegungsgleichungen aufintegriert, sodass sich die Systemdynamik im Zeitbereich ergibt. Dabei wird ausschließlich mit numerischen Integrationsverfahren gearbeitet. Diese können sich hinsichtlich Komplexität, Genauigkeit und Effizienz erheblich unterscheiden [38, 39].

Integrationsverfahren

Sind in den zu integrierenden Bewegungsgleichungen keine kinematischen Zwangsbedingungen enthalten, ergibt sich mit dem Zustandsvektor

$$\mathbf{x} = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \end{pmatrix} \quad (2.3)$$

und der Systemfunktion

$$\phi(\mathbf{x}) = \begin{pmatrix} \mathbf{v} \\ M^{-1}(\mathbf{p})\mathbf{f}(\mathbf{p}, \mathbf{v}) \end{pmatrix} \quad (2.4)$$

ein gewöhnliches Differentialgleichungssystem (ODE, *Ordinary Differential Equation*) erster Ordnung:

$$\dot{\mathbf{x}} = \phi(\mathbf{x}) \quad (2.5)$$

Die Integration kann hier durch vergleichsweise einfache Verfahren erfolgen [38].

Enthalten die Bewegungsgleichungen dagegen Zwangsbedingungen, ist größerer Aufwand erforderlich. Mit der Definition

$$\mathbf{F}(\mathbf{p}, \mathbf{v}, \lambda) = \begin{pmatrix} -\dot{\mathbf{p}} + T(\mathbf{p})\mathbf{v} \\ -M(\mathbf{p})\dot{\mathbf{v}} + \mathbf{f}(\mathbf{p}, \mathbf{v}) + G^T(\mathbf{p})\lambda \\ \mathbf{g}(\mathbf{p}) \end{pmatrix} \quad (2.6)$$

und dem erweiterten Zustandsvektor

$$\mathbf{y} = \begin{pmatrix} \mathbf{p} \\ \mathbf{v} \\ \lambda \end{pmatrix} \quad (2.7)$$

lässt sich das differential-algebraische Gleichungssystem (DAE, *Differential Algebraic Equation*) (2.1) in impliziter Form darstellen:

$$\mathbf{F}(t, \dot{\mathbf{y}}, \mathbf{y}) = \mathbf{0} \quad (2.8)$$

Ein wichtiges Merkmal von DAEs ist ihr *Differentiationsindex*. Er gibt an, wie viele Differentiationen erforderlich sind, um das System prinzipiell mit Hilfe algebraischer Umformungen in ein ODE-System zu überführen. Dies ist für (2.1) durch dreimaliges Ableiten der Nebenbedingungen (2.1c) möglich, sodass es sich um ein Index-3-System handelt [37].

Der Index ist ein Maß für die Komplexität des Problems. So stehen nur für DAE bis zum Index 2 effiziente Integrationsmethoden zur Verfügung. Um die MKS-Bewegungsgleichungen numerisch lösen zu können, ist deshalb eine *Indexreduktion* erforderlich. Dies lässt sich formal durch Differentiation von (2.1c), d. h. Formulierung der Zwangsbedingungen auf Geschwindigkeitsebene

$$\mathbf{0} = \dot{\mathbf{g}}(\mathbf{p}, \mathbf{v}) = \frac{\partial \mathbf{g}(\mathbf{p})}{\partial \mathbf{p}} \dot{\mathbf{p}} = G(\mathbf{p})\mathbf{v} \quad (2.9)$$

oder Beschleunigungsebene

$$\mathbf{0} = \ddot{\mathbf{g}}(\mathbf{p}, \mathbf{v}, \dot{\mathbf{v}}) = G(\mathbf{p})\dot{\mathbf{v}} + \dot{G}(\mathbf{p}, \mathbf{v})\mathbf{v} \quad , \quad (2.10)$$

erreichen [37].

Allerdings entstehen dadurch neue Schwierigkeiten. Neben größerem Aufwand bei der Formulierung der Nebenbedingungen und zur Bestimmung eines konsistenten Anfangszustands [37] ist hierbei der *Drift-off-Effekt* zu nennen: Da die Zwangsbedingungen nur im Rahmen der numerischen Genauigkeitsschranken gelöst werden, führen die verbleibenden Restgeschwindigkeiten bzw. Restbeschleunigungen im Laufe der Integration zu inakzeptablen Kriechbewegungen in den zugehörigen Gelenken. Abhilfe kann z. B. durch

Integrationsverfahren	Typ	Ordnung p	Fkt.–Ausw. je Schritt
Euler explizit	exp. Einschnitt	1	1
Euler implizit	imp. Einschnitt	1	≥ 1
Runge–Kutta explizit	exp. Einschnitt	≥ 1	$\geq p$
Runge–Kutta implizit	imp. Einschnitt	≥ 1	$\geq p/2$
Adams–Bashforth	exp. Mehrschritt	≥ 1	1
Adams–Moulton	imp. Mehrschritt	≥ 2	≥ 2
BDF (Gear)	imp. Mehrschritt	≥ 1	≥ 1

Tabelle 2.1: Eigenschaften einiger wichtiger Integrationsverfahren; abgesehen von den Eulerverfahren existieren jeweils Varianten mit verschiedener Ordnung

das Stabilisierungsverfahren nach Baumgarte [40] oder Indexreduktion mittels Gear–Gupta–Leimkuhler–Formulierung [41] geschaffen werden.

Bei der numerischen Integration von (2.5) liefern die Bewegungsgleichungen die zeitliche Ableitung zu einem gegebenen Systemzustand \mathbf{x} in Form einer Auswertung der Systemfunktion $\phi(\mathbf{x})$. Die Effizienz der Simulation wird bestimmt durch den Rechenaufwand je Auswertung sowie die vom Integrationsverfahren benötigte Anzahl an Funktionsauswertungen.

Integrationsmethoden lassen sich in explizite und implizite, in Einschnitt- und Mehrschritt- sowie in ODE- und DAE–Verfahren einteilen (Tabelle 2.1). Außerdem sind Ordnung und Stabilitätsgebiete als wichtige Merkmale zu nennen [38, 39, 42].

Für Systeme gewöhnlicher Differentialgleichungen (2.5) ist das *explizite Eulerverfahren*

$$\mathbf{x}_{n+1} := \mathbf{x}_n + h \cdot \phi(\mathbf{x}_n) \quad (2.11)$$

die einfachste denkbare Variante: Der aktuelle Systemzustand \mathbf{x}_n wird anhand der durch die Funktionsauswertung gewonnenen Ableitung $\phi(\mathbf{x}_n)$ linear um die Schrittweite $h := t_{n+1} - t_n$ entlang der Zeitachse verändert. Weitere wichtige Vertreter der expliziten Einschnittmethoden sind die Verfahren vom *Runge–Kutta*-Typ; hier werden zunächst an verschiedenen Punkten im zu integrierenden Zeitschritt $[t_n, t_{n+1}]$ die Ableitungen näherungsweise durch verallgemeinerte Eulerschritte bestimmt, um dann \mathbf{x}_{n+1} anhand von Quadraturformeln zu bestimmen [42].

Bei Mehrschrittverfahren werden zur Berechnung eines Zeitschritts nicht nur die Ableitung an dessen Beginn, sondern zusätzlich Funktionsauswertungen aus benachbarten Schritten berücksichtigt. Als bekannte Beispiele sind die expliziten *Adams–Bashforth–Verfahren* zu nennen.

Da Mehrschrittverfahren normalerweise mit weniger Auswertungen der Systemfunktion auskommen, arbeiten sie i. d. R. effizienter. Das gilt aber nicht für Systeme mit starken Änderungen im Lösungsverhalten, wo Einschrittverfahren flexibler reagieren können und daher überlegen sind [39].

Während sich bei expliziten Integrationsmethoden die Berechnungsvorschrift für den nächsten Schritt direkt angeben lässt, enthält sie bei impliziten Verfahren die zunächst unbekannte Ableitung $\phi(\mathbf{x}_{n+1})$ am zu berechnenden zukünftigen Zeitpunkt t_{n+1} , z. B. für das *implizite Eulerverfahren*:

$$\mathbf{x}_{n+1} := \mathbf{x}_n + h \cdot \phi(\mathbf{x}_{n+1}) \quad (2.12)$$

Damit erfordert jeder Zeitschritt die Lösung eines i. Allg. nichtlinearen algebraischen Gleichungssystems für \mathbf{x}_{n+1} . Zur Lösung wird i. d. R. das iterative *Newton–Raphson–Näherungsverfahren* verwendet, welches die *Jacobimatrix* $\partial\phi/\partial\mathbf{x}$ der Systemfunktion benötigt.

Meist wird die Jacobimatrix durch numerische Differenzenapproximation berechnet, welche i. Allg. jeweils $n_x + 1$ Auswertungen der Systemfunktion erfordert und damit den Rechenaufwand erheblich erhöht. Deshalb wird versucht, die Berechnung durch Strukturausnutzung zu vereinfachen oder wenn möglich ganz zu vermeiden, indem man auf das Ergebnis des letzten Integrationsschritts zugreift [3, 37, 43].

Wichtige implizite Integrationsmethoden sind die impliziten Runge–Kutta Einschrittverfahren, die *Adams–Moulton–Verfahren* und das BDF–Verfahren (*Backwards Differentiation Formulae*) nach Gear. Letzteres unterscheidet sich prinzipiell dadurch von den Adams–Verfahren, dass in der Berechnungsvorschrift auf die voran gegangenen Werte des Zustandsvektors \mathbf{x} anstelle der Systemfunktion $\phi(\mathbf{x})$ zurückgegriffen wird.

Ordnung und Stabilität

Das explizite Eulerverfahren ist besonders einfach zu implementieren und kommt mit einer einzigen Auswertung der Systemfunktion pro Zeitschritt aus. Weil damit der Rechenaufwand je Zeitschritt besonders gering und genau definiert ist, wird es in der Praxis häufig für Echtzeitanwendungen eingesetzt. Allerdings müssen hierbei erhebliche Nachteile hinsichtlich Genauigkeit und Stabilität der Integration in Kauf genommen werden.

Die aufwändigeren Methoden zeichnen sich dagegen durch geringere numerische Fehler aus, was sich anhand der *Ordnung* p beziffern lässt [39, 42]: Ist die Systemfunktion $p + 1$ -mal stetig differenzierbar, gilt für den globalen Fehler

$$\|\varepsilon_n\| = \mathcal{O}(h^p) \quad , \quad (2.13)$$

d. h. Verfahren höherer Ordnung konvergieren mit der p -ten Potenz der Schrittweite gegen die exakte Lösung. Während also das explizite Eulerverfahren mit der Ordnung $p = 1$ für eine um zwei Stellen genauere Lösung eine hundertmal kleinere Schrittweite erfordert, genügt bei einem Verfahren fünfter Ordnung etwa der Faktor 0,4.

Da (2.13) nur bei $p + 1$ -mal stetiger Differenzierbarkeit von $\phi(\mathbf{x})$ gilt, zahlt sich der Mehraufwand von Integrationsmethoden mit höherer Ordnung i. d. R. allerdings nur für entsprechend glatte Lösungsverläufe aus. Sonst muss in jedem Fall mit kleiner Schrittweite gerechnet werden, sodass Verfahren niedriger Ordnung effizienter arbeiten.

Ein sehr wichtiger Aspekt bei der Integration der Bewegungsgleichungen ist die Stabilität. Dazu ist zunächst die *physikalische Stabilität* des gegebenen MKS zu betrachten [3, 42], die sich im nichtlinearen Fall z. B. nach Linearisierung anhand der Eigenwerte $\lambda_i \in \mathbb{C}$ der Jacobimatrix $\partial\phi/\partial\mathbf{x}(\mathbf{x}^*)$ analysieren lässt. Hierbei muss \mathbf{x}^* einen Gleichgewichtszustand mit $\phi(\mathbf{x}^*) = \mathbf{0}$ darstellen.

Jeder komplexe Eigenwert charakterisiert eine Eigenform des Systems, wobei $\text{Re } \lambda_i$ die Dämpfung und $\text{Im } \lambda_i$ die Frequenz der entsprechenden Schwingung angibt. Reale mechanische Systeme werden i. d. R. physikalisch stabil ausgelegt, sodass sämtliche Eigenschwingungen abklingen und die Realteile aller Eigenwerte negativ sind. In der komplexen Zahlenebene liegen dabei schwach gedämpfte Eigenwerte nahe der imaginären Achse, während weit entfernte starker Dämpfung entsprechen. Analog steigt mit dem Abstand zur reellen Achse die zugehörige Frequenz.

Systeme mit stark unterschiedlichen Beträgen der Eigenwerte werden als *steif* bezeichnet. Zur Quantifizierung dient der Steifigkeitsindex

$$\kappa = \frac{|\lambda|_{max}}{|\lambda|_{min}} \quad , \quad (2.14)$$

wobei $\kappa \geq 1000$ als Faustwert für die Zuordnung zur Klasse der steifen Probleme gilt [42]. Steife Bewegungsgleichungen ergeben sich für MKS, bei denen Kräftelemente mit großen Steifigkeiten und/oder Dämpfungen die Bewegung von Körpern mit relativ kleiner Trägheit bestimmen.

Die *numerische Stabilität* der Zeitintegration ist gegeben, wenn die Zustandsgrößen während der Simulation beschränkt bleiben [3, 39, 42]. Das ist dann der Fall, wenn für sämtliche (physikalischen) Eigenwerte des Systems das Produkt $h\lambda_i$ aus Schrittweite und Eigenwert innerhalb des sog. (numerischen) *Stabilitätsgebiets* S liegen [44]. Zu einem vorgegebenen Integrationsverfahren lässt sich dieses Stabilitätsgebiet durch Analyse der Zeitintegration einer skalaren Testgleichung bestimmen. Für die Integration der

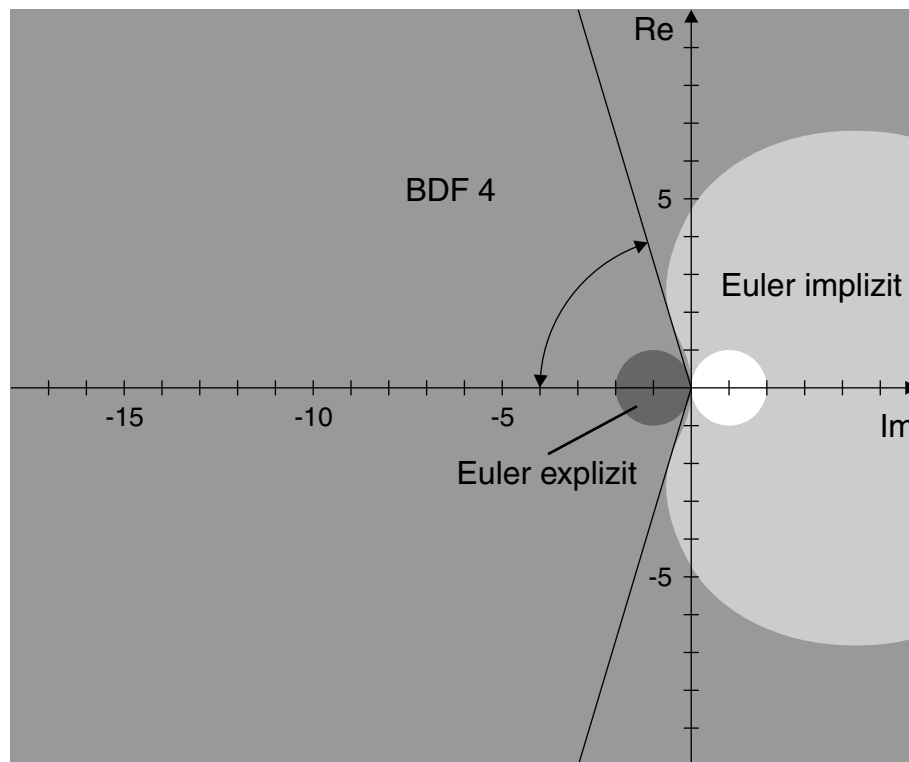


Abbildung 2.4: Stabilitätsgebiete des expliziten und impliziten Eulerverfahrens sowie des BDF-Verfahrens vierter Ordnung ($A(\alpha)$ -stabil mit $\alpha = 73,35^\circ$)

Bewegungsgleichungen eines MKS ist h stets so zu wählen, dass $h\lambda_i$ für alle seine Eigenwerte im Inneren von S liegen:

$$h \cdot \lambda_i \in S \quad (2.15)$$

Beim kreisförmigen Stabilitätsgebiet des expliziten Eulerverfahrens (Abbildung 2.4) erfordern daher stark gedämpfte und/oder hochfrequente Eigenwerte extrem kleine Schrittweiten, was in der Praxis nicht nur zu sehr hohem Rechenaufwand führt, sondern aufgrund der damit in den Vordergrund rückenden numerischen Rundungsfehler oft nicht mehr praktikabel ist.

Dagegen erstreckt sich das Stabilitätsgebiet impliziter Methoden häufig über den größten Teil der linken komplexen Zahlenebene. Der dabei auftretende Öffnungswinkel α eines Sektors um die negative reelle Achse gibt den Bereich an, in dem die Eigenwerte unabhängig von der Schrittweite stets (2.15) erfüllen. Die Verfahren werden daher $A(\alpha)$ -stabil genannt [3, 44]. Im Idealfall ist $\alpha = 90^\circ$ und man spricht von A -Stabilität. Der relativ große

numerische Aufwand impliziter Integrationsverfahren wird also durch ihre Eignung für in der Praxis häufig anzutreffende steife mechanische Systeme gerechtfertigt.

Ziel der numerischen Integration ist es, das gegebene Problem in der geforderten Genauigkeit bei möglichst geringem Rechenaufwand zu lösen. Aus dem bisher Gesagten wird deutlich, dass die Wahl des optimalen Integrationsverfahrens stark von den Systemeigenschaften abhängt. Der Berechnungsingenieur ist hierbei oft überfordert. Außerdem kommt es in praktischen Anwendungen häufig vor, dass sich die Eigenschaften des MKS im Laufe der Simulation erheblich verändern. Beispielsweise können harte Anschläge von Fahrwerken dafür sorgen, dass sich die Steifheit der Bewegungsgleichungen zeitweise erheblich verändert.

Diese Konflikte werden oft erfolgreich durch den Einsatz adaptiver Integratoren gelöst, die z. B. Schrittweite und Ordnung während der Rechnung in Abhängigkeit des Integrationsverlaufs verändern, sodass optimale Effizienz erreicht wird.

Der Integrator DASSL

Der Integrator DASSL [45] hat sich in der Praxis besonders für die MKS-Simulation bewährt. Hierbei handelt es sich um ein BDF-basiertes Verfahren, das mit variabler Schrittweite und Ordnung ($p = 1..5$) arbeitet. Ohne auf Details einzugehen, soll im Folgenden kurz die Funktionsweise beschrieben werden (Abbildung 2.5).

DASSL arbeitet nach dem *Prädiktor-Korrektor-Prinzip*, um DAE-Systeme in der Form (2.8) zu lösen: Um den nächsten Systemzustand \mathbf{y}_{n+1} zu berechnen, wird zunächst ein Polynom $\omega_{n+1}^P(t)$ bestimmt, das die numerischen Lösungen der vergangenen $p + 1$ Schritte interpoliert:

$$\omega_{n+1}^P(t_{n-i}) = \mathbf{y}_{n-i} \quad i = 0..p \quad (2.16)$$

Dabei entspricht p der aktuellen Ordnung.

Durch Auswerten des Prädiktorpolynoms zum Endzeitpunkt $t_{n+1} := t_n + h$ des zu berechnenden Zeitschritts (mit der aktuellen Schrittweite h) gewinnt man einen Startwert für die folgende Korrektoriteration:

$$\mathbf{y}_{n+1}^{(0)} := \omega_{n+1}^P(t_{n+1}) \quad (2.17)$$

Anschließend wird iterativ ein Korrektorpolynom $\omega_{n+1}^K(t)$ berechnet, das durch p gleichverteilte Punkte des Prädiktorpolynoms sowie — im Rahmen

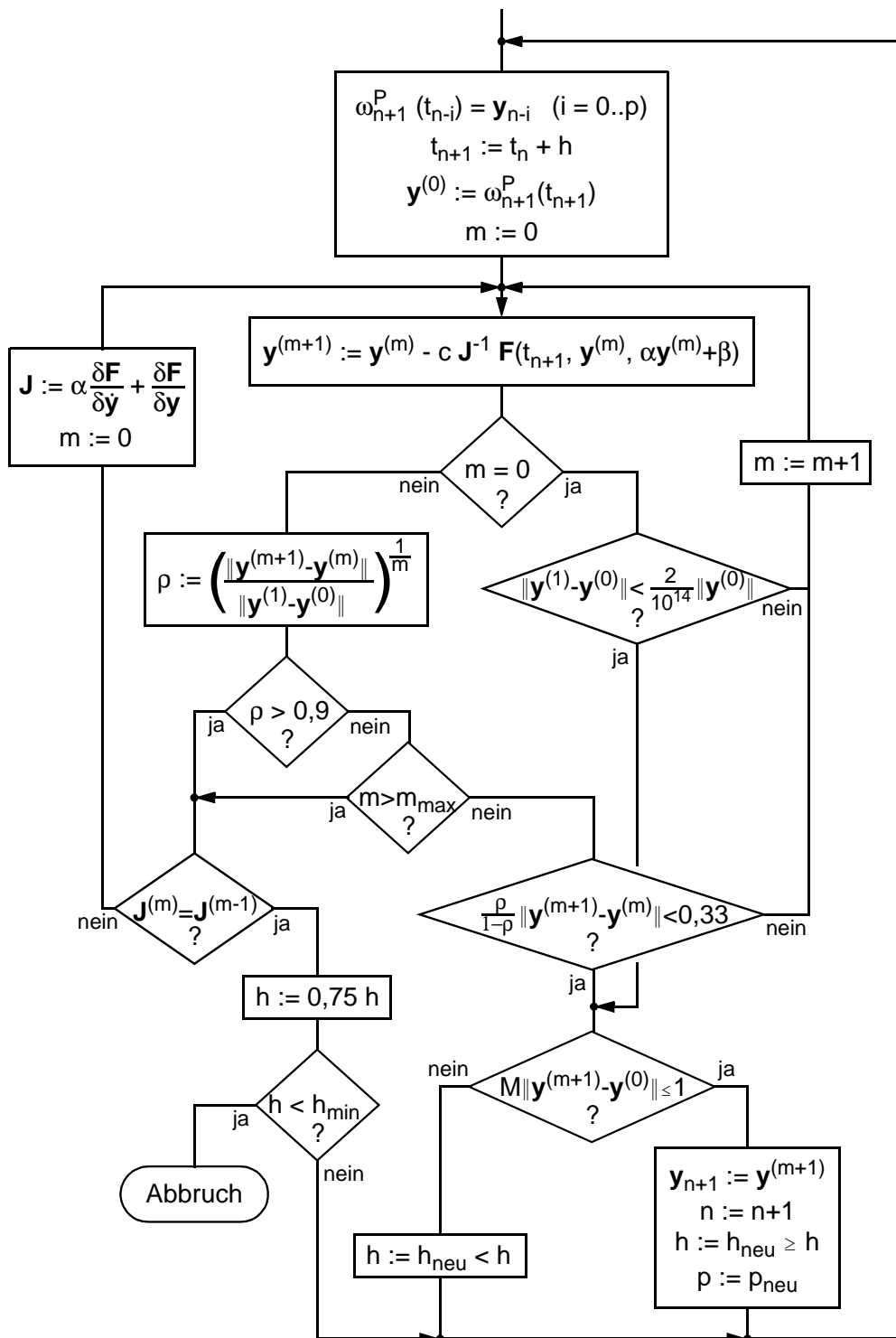


Abbildung 2.5: Flussdiagramm des DAE-Integrators DASSL

der vom Benutzer vorgegebenen Fehlertoleranzen — den gesuchten Systemzustand \mathbf{y}_{n+1} verläuft und die Gleichungen des gegebenen DAE-Systems zum Zeitpunkt t_{n+1} erfüllt:

$$\omega_{n+1}^K(t_{n+1} - ih) = \omega_{n+1}^P(t_{n+1} - ih) \quad i = 1..p \quad (2.18a)$$

$$\omega_{n+1}^K(t_{n+1}) = \mathbf{y}_{n+1} \quad (2.18b)$$

$$\mathbf{F}(t_{n+1}, \dot{\omega}_{n+1}^K(t_{n+1}), \omega_{n+1}^K(t_{n+1})) = \mathbf{0} \quad (2.18c)$$

Unter Verwendung von dividierten Differenzen wird (2.18) durch eine modifizierte Newton-Iteration mit der Berechnungsvorschrift

$$\mathbf{y}_{n+1}^{(m+1)} = \mathbf{y}_{n+1}^{(m)} - c \cdot \mathbf{J}^{-1} \cdot \mathbf{F}(t_{n+1}, \mathbf{y}_{n+1}^{(m)}, \alpha \mathbf{y}_{n+1}^{(m)} + \beta) \quad (2.19)$$

gelöst [45]. Dabei wird die Jacobimatrix

$$\mathbf{J} = \alpha \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{y}}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}} \quad (2.20)$$

benötigt, deren Berechnung mittels numerischer Differenzenapproximation $n_y + 1$ Auswertungen der Systemfunktion \mathbf{F} erfordert und daher i. d. R. den größten Teil des Rechenaufwands verursacht. Sie wird deshalb nur dann neu berechnet, wenn die Korrektoriteration nicht mehr oder zu langsam konvergiert. Die skalare Größe α wird in Abhängigkeit von der aktuellen Schrittweite vorgegeben, und β ergibt sich aus $\beta := \omega_{n+1}^{iP}(t_{n+1}) - \alpha \mathbf{y}_{n+1}^{(0)}$.

Unterscheidet sich das Ergebnis der ersten Iteration ($m = 0$) nur geringfügig vom Startwert aus dem Prädiktorschritt, wird es als Ergebnis akzeptiert. Ansonsten erfolgt eine weitere Korrektoriteration nach (2.19), die anhand der Konvergenzrate

$$\rho = \left(\frac{\|\mathbf{y}_{n+1}^{(m+1)} - \mathbf{y}_{n+1}^{(m)}\|}{\|\mathbf{y}_{n+1}^{(1)} - \mathbf{y}_{n+1}^{(0)}\|} \right)^{\frac{1}{m}} \quad (2.21)$$

bewertet wird. Bei ausreichender Konvergenzgeschwindigkeit dient die Bedingung

$$\frac{\rho}{1 - \rho} \cdot \|\mathbf{y}_{n+1}^{(m+1)} - \mathbf{y}_{n+1}^{(m)}\| < 0,33 \quad (2.22)$$

als Abbruchkriterium der Korrektoriteration. Ist dagegen die Konvergenzrate zu groß ($\rho > 0,9$) oder die Obergrenze m_{max} für die Anzahl der Iterationen erreicht, wird vor dem nächsten Iterationsschritt die Jacobimatrix \mathbf{J} neu berechnet. Falls auch damit keine ausreichende Konvergenzrate ρ erreicht wird, beginnt die gesamte Iteration mit verminderter Schrittweite von vorne.

Schließlich wird die so ermittelte Lösung \mathbf{y}_{n+1} einem Fehlertest unterzogen, indem die Bedingung

$$M \cdot \|\mathbf{y}_{n+1} - \mathbf{y}_{n+1}^{(0)}\| \leq 1 \quad (2.23)$$

überprüft wird. Hierbei werden der lokale Diskretisierungsfehler der Integration sowie der Interpolationsfehler zur Bestimmung des Systemzustands innerhalb der Integrationsschritte abgeschätzt, indem die Norm des Differenzvektors zwischen dem Prädiktor- und dem Korrektorzustand mit einer von Ordnung und Schrittweite abhängigen Konstante M multipliziert wird [45]. Je nach Ergebnis des Fehlertests wird die Integration mit gleicher oder größerer Schrittweite fortgesetzt oder aber \mathbf{y}_{n+1} verworfen und mit verminderter Schrittweite neu berechnet.

Auch die Ordnungssteuerung greift nach der Korrektoriteration ein. Dabei werden die Abweichungen zwischen dem Prädiktor- und dem Korrektorpolynom anhand ihrer Taylorentwicklung abgeschätzt. Bei wachsenden Differenzen über die letzten Schritte wird die Ordnung erhöht, bei sinkenden reduziert.

Wichtig zum Verständnis des Algorithmus ist, dass die verwendete Norm durch die vom Anwender vorgegebenen Toleranzen $atol$ und $rtol$ für den absoluten und relativen Fehler gewichtet wird:

$$\|\mathbf{y}^{(m+1)} - \mathbf{y}^{(m)}\| := \sqrt{\frac{1}{n_y} \sum_{i=1}^{n_y} \left(\frac{y_i^{(m+1)} - y_i^{(m)}}{atol_i + rtol_i \cdot \max(|y_i^{(m+1)}|, |y_i^{(m)}|)} \right)^2} \quad (2.24)$$

Faktisch werden also Komponenten mit betragsmäßig sehr kleinen Werten mit dem Faktor $1/atol_i$, und die restlichen mit $1/(|y_i|rtol_i)$ skaliert.

Für die Implementierung von MKS-Elementen ist von besonderem Interesse, dass sie als Teil der Systemfunktion \mathbf{F} in zwei unterschiedlichen Stufen des Integrationsverfahrens ausgewertet werden. Dies geschieht einerseits jeweils genau einmal zur Berechnung von Newton-Schritten nach (2.19), andererseits — falls erforderlich — jeweils mehrfach mit geringfügig variierten Systemzuständen zur Berechnung der Jacobimatrix (2.20) durch Differenzenapproximation.

Unstetigkeiten

Die Voraussetzung ausreichend häufiger Differenzierbarkeit für (2.13) macht bereits deutlich, dass die Glattheit des zu berechnenden Lösungsverlaufs erheblichen Einfluss auf die Robustheit und Effizienz der numerischen

Integration hat. Auch in den Polynomapproximationen (2.16) und (2.18a), (2.18b) durch DASSL spiegelt sich dieser Zusammenhang wider.

Andererseits führt das erwähnte Beispiel der Fahrwerksanschlüge zu sehr großen Gradienten oder gar Sprüngen in den Kräften und damit auch in den Beschleunigungen des MKS. Werden Starrkörperkontakte mit Stößen modelliert, so treten gar Sprünge in den Geschwindigkeiten auf (s. Abschnitt 3.5.2). Darüber hinaus kann sich das Verhalten bestimmter MKS-Elemente diskret ändern, etwa beim Übergang von Haft- zu Gleitreibung eines Kraftelements; man spricht dann von einem *strukturvarianten* System.

Um trotz solcher systembedingter Unstetigkeiten eine stabile und effiziente Integration zu ermöglichen, arbeiten viele Verfahren mit *Schaltfunktionen* [3, 39, 46]. Die Idee ist hierbei, die Berechnung bei entsprechenden Ereignissen zu unterbrechen, um nach Neuinitialisierung des Integrators und der schaltbaren MKS-Elemente fortzusetzen. Somit werden also abschnittsweise glatte bzw. stetige Systeme betrachtet.

In der Praxis erweist sich die optionale Verwendung von Schaltfunktionen nicht immer als vorteilhaft. Denn treten während der Simulation ständig Unstetigkeiten in sehr geringem zeitlichen Abstand auf, verläuft die numerische Integration wegen fortwährender Initialisierung mit entsprechend kleiner Schrittweite und Ordnung. Für viele strukturvariante Systeme sind Schaltfunktionen allerdings unverzichtbar.

Kapitel 3

Kontaktmodelle

In diesem Kapitel werden zunächst die wichtigsten Grundlagen und Schwierigkeiten bei der Modellierung von Kontaktvorgängen zusammengestellt. Weiter sollen die Abschnitte 3.3 bis 3.5 einen Überblick bekannter Verfahren zur Berechnung von Kontaktkräften geben. Dabei wird von komplexen Methoden mit hoher Abbildungsgüte zu einfachen, stark abstrahierenden Modellen vorgegangen. Der Umfang der einzelnen Abschnitte richtet sich allein nach ihrer Bedeutung für die weiteren Kapitel, und es steht stets die Tauglichkeit für die MKS-Simulation im Mittelpunkt. In diesem Sinne wird im letzten Abschnitt bewertet und verglichen.

3.1 Grundlagen

Um Kontaktvorgänge in der numerischen Simulation darstellen zu können, muss ein geeignetes physikalisches Modell entwickelt werden. Dazu gilt es zunächst, die Prinzipien dieses Phänomens zu verdeutlichen. Zur Formulierung der Grundlagen werden zwei kontaktierende Körper 1 und 2 betrachtet (Abbildung 3.1). Die durchgezogenen, dicken Linien zeigen ihre durch die Kontaktspannungen verformten Oberflächen; die gestrichelten, dünnen Linien stellen ihre gedachten unverformten Profile dar.

3.1.1 Impenetrabilität

Das Prinzip der Impenetrabilität besagt, dass sich die Körper nicht durchdringen. Dies kann durch eine sog. *Gap-Funktion*

$$g = u_{n1} + u_{n2} - d \geq 0 \quad (3.1)$$

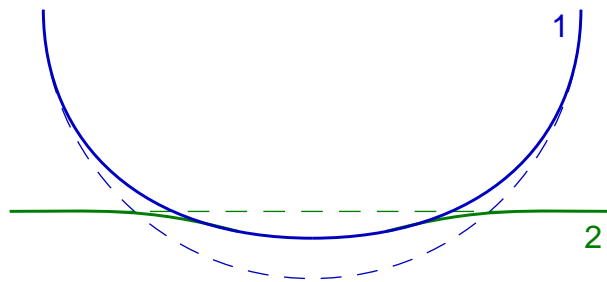


Abbildung 3.1: Kontaktierende Körper

beschrieben werden, in der sich der Abstand der Oberflächen g aus den Verformungen u_{n1} und u_{n2} in Kontaktnormalenrichtung und der Durchdringung d der unverformten Profile ergibt (Abbildung 3.2).

3.1.2 Adhäsionsfreiheit

Adhäsionsfreiheit bedeutet, dass senkrecht zur Kontaktfläche keine Zugspannungen auftreten. Bezeichnet man mit positivem p Druckspannungen (Abbildung 3.2), gilt also:

$$p \geq 0 \quad (3.2)$$

Sonderfälle durch klebrige Oberflächen werden hier nicht betrachtet.

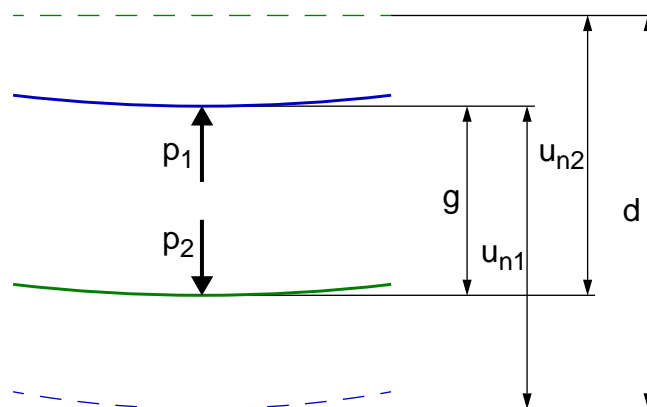


Abbildung 3.2: Gap-Funktion und Kontaktnormalenspannung

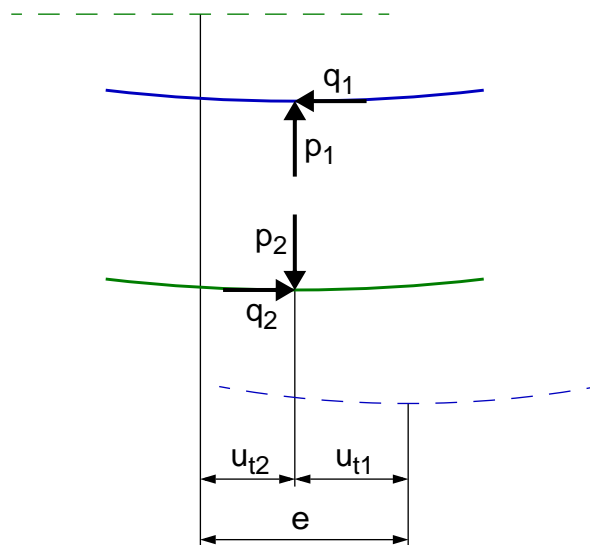


Abbildung 3.3: Gegenwirkungsprinzip und Tangentialverformung

3.1.3 Gegenwirkungsprinzip

Ein weiterer wichtiger Grundsatz ist das Gegenwirkungsprinzip („*actio gleich reactio*“) der Kontaktspannungen zwischen gegenüberliegenden Flächenelementen im Berührgebiet (Abbildung 3.3):

$$p_1 + p_2 = 0 \quad (3.3)$$

$$\mathbf{q}_1 + \mathbf{q}_2 = \mathbf{0} \quad (3.4)$$

Dabei sind die tangential wirkenden Verschiebungen \mathbf{u}_t und Spannungen \mathbf{q} als Vektoren formuliert, weil sie im räumlichen Fall eine Richtung innerhalb der Tangentialebene besitzen.

3.1.4 Prinzip von St. Venant

Nach dem *Prinzip von St. Venant* hängen Spannungen und Verformungen eines elastischen Körpers in hinreichender Entfernung vom Angriffsbereich äußerer Kräfte nicht mehr von ihrer speziellen Verteilung, sondern nur noch von ihrer Resultierenden ab [47]. Für kontaktmechanische Problemstellungen in MKS bedeutet dies, dass die Kontaktspannungen berührender Körper, welche typischerweise in einem relativ kleinen Gebiet auftreten, zum resultierenden Kraftwinder aufintegriert bzw. aufsummiert werden können, um als eingetragte Kräfte und Momente berücksichtigt zu werden. Darüber hinaus

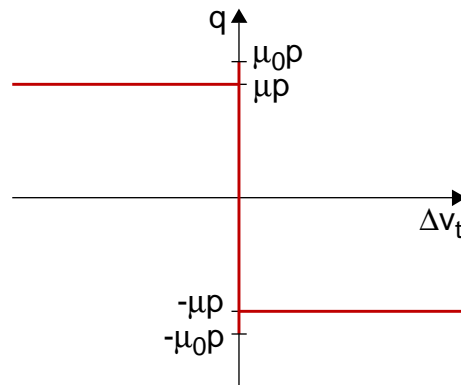


Abbildung 3.4: Coulombsches Reibgesetz

wird im Weiteren davon ausgegangen, dass Veränderungen von Schwerpunktlage und Trägheitstensor infolge relativ kleiner Verformungen durch Kontakt der Körper vernachlässigbar sind.

3.1.5 Reibung

Reibung zwischen festen Körpern ist ein Phänomen, welches durch physikalische Modelle approximiert werden kann. Sie stellt Kraftwirkungen dar, die tangential zur Kontaktfläche wirken. Sind die Oberflächen trocken, wird in den weitaus meisten Fällen das *Coulombsche Reibgesetz* (Abbildung 3.4) angewendet. Dieses unterscheidet zwischen zwei Zuständen:

Liegt *Haften* (auch *Adhäsion* oder *Stick* genannt) vor, tritt keine relative Tangentialverschiebung in der Kontaktfläche auf. Bezeichnet man entsprechend Abbildung 3.3 mit \mathbf{u}_{t1} und \mathbf{u}_{t2} die tangentialen Verformung und mit \mathbf{e} die entsprechende Verschiebung der unverformten Profile, kann die Haftreibung durch einen der Gap-Funktion (3.1) analogen Zusammenhang beschrieben werden:

$$\mathbf{e} - (\mathbf{u}_{t1} + \mathbf{u}_{t2}) = 0 \quad (3.5)$$

Der zweite Zustand heißt *Gleiten* (oder auch *Slip*) und besagt, dass sich bei Tangentialbewegung der Kontaktflächen eine der Relativgeschwindigkeit $\Delta \mathbf{v}_t$ entgegenwirkende Traktionsspannung \mathbf{q} einstellt, welche proportional mit der Normalspannung p wächst:

$$\mathbf{q} = -\mu \cdot p \cdot \frac{\Delta \mathbf{v}_t}{|\Delta \mathbf{v}_t|} \quad (3.6)$$

Die Proportionalitätskonstante μ wird als *Gleitreibungszahl* bezeichnet und hängt in erster Linie von der Materialpaarung und der Oberflächenbeschaffenheit ab. Ihr Wertebereich liegt zwischen null und typischerweise kaum mehr als eins.

Für den Übergang zwischen den Zuständen gelten zwei verschiedene Kriterien. Von Gleiten nach Haften ist die tangentielle Relativgeschwindigkeit maßgeblich; sobald sie verschwindet, kann Adhäsion auftreten. Umgekehrt löst sich die feste Bindung, wenn die *Haftkraftreserve*

$$|\mathbf{q}| = \mu_0 \cdot p \quad (3.7)$$

überschritten wird. Der Koeffizient μ_0 wird *Haftreibungszahl* genannt und ist stets etwas größer als μ .

Es sei darauf hingewiesen, dass in der Praxis ein recht großer Teil von Kontaktproblemen auch ohne Berücksichtigung der Reibung zufriedenstellend gelöst werden kann.

3.2 Schwierigkeiten

Die vorgestellten Grundlagen der Kontaktmechanik erscheinen schlüssig und keineswegs besonders abstrakt oder komplex. Trotzdem stößt man bei der Analyse auf schwierige Probleme, die häufig erst durch die Abbildung der Phänomene und Modelle auf die numerische Simulation entstehen oder verschärft werden. Im Folgenden werden einige typische Schwierigkeiten vorgestellt.

3.2.1 Kontakterkennung

Wie bereits erwähnt, werden die räumliche Ausdehnung und Oberfläche eines Körpers in einem klassischen MKS-Modell nicht abgebildet. Für die Untersuchung von Kontaktproblemen ist deshalb ein Verfahren vorzusehen, welches anhand der relativen Lage und Orientierung der n Körper ihren *Kontaktstatus* ermittelt. I. Allg. muss für jede der $\frac{n(n-1)}{2}$ möglichen Körperpaarungen untersucht werden, ob sich ihre (unverformten) Oberflächen schneiden, sodass überhaupt erst ein Kontaktproblem vorliegt. Ansonsten ist keine weitere kontaktmechanische Analyse erforderlich. Je nach Darstellung der Körperoberflächen bedarf eine effiziente Kontakterkennung mitunter aufwändiger Verfahren (s. Abschnitt 4.1).

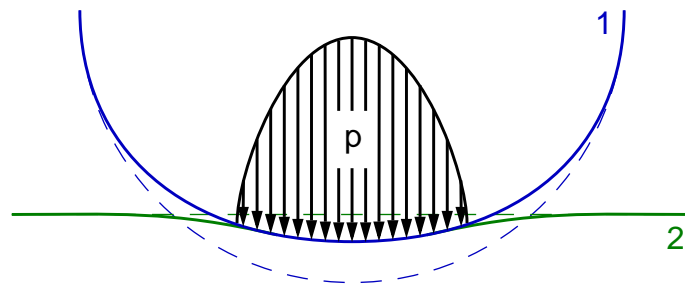


Abbildung 3.5: Kontaktgebiet und Normalspannungsverteilung

3.2.2 Kontaktgebiet

Beim Kontakt zweier Körper treten aufgrund der Impenetrabilität und ihrer Elastizität Verformungen auf (Abbildung 3.5). In der Regel verschieben sich die Oberflächen in Richtung der Kontaktnormalspannungen nach innen. Scherspannungen im Körperinneren sorgen dafür, dass sich die Verformung auch in der näheren Umgebung neben dem Berührgebiet fortsetzt. Daraus ergibt sich das Problem, dass Größe und Gestalt des Kontaktgebiets zunächst nicht bekannt sind. Eine erste Näherung stellt die Schnittkurve der unverformten Oberflächen dar, doch das tatsächliche Berührgebiet ist normalerweise kleiner. Diese Problematik kann bei der numerischen Analyse durch iterative Algorithmen oder Variationsrechnung gelöst werden (s. Abschnitt 3.4.1).

3.2.3 Partielles Gleiten

Durch die Elastizität der Körper und die zwischen ihnen wirksame Reibung können sich in der Berührfläche Gebiete ausbilden, in denen neben einander Haft- und Gleitreibung auftreten. Beispielsweise stellt sich beim Radkontakt eines Schienenfahrzeugs eine Verteilung wie schematisch in Abbildung 3.6 dargestellt ein. Infolge dessen können sich selbst bei statischen Betrachtungen in kinematisch identischen Kontaktsituationen unterschiedliche Reibungs- und Spannungsverteilungen einstellen, die von der *Belastungsgeschichte* abhängen [48]. Um den Zustand des partiellen Gleitens zu ermitteln, sind auch hier iterative Methoden oder Variationsrechnung zur numerischen Auswertung erforderlich (s. Abschnitt 3.4.1).

3.2.4 Materialdämpfung

Die Elastizität jedes Materials geht mit Dämpfung einher, welche sein dissipatives Verhalten bei Verformung darstellt. Da in MKS-Simulationen oft ho-

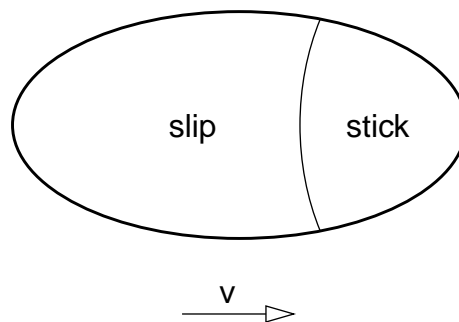


Abbildung 3.6: Partielles Gleiten im Rad–Schiene Kontaktgebiet (angetriebenes Rad)

he Kontaktgeschwindigkeiten auftreten, muss die Materialdämpfung mit geeigneter Komplexität modelliert werden. In der einschlägigen Literatur fällt jedoch auf, dass ihre dynamischen Effekte insbesondere bei aufwändigeren Kontaktmodellen häufig gar nicht oder nur stark vereinfacht Berücksichtigung finden.

3.3 Festkörpermodelle

Kontaktkräfte treten an der Oberfläche von festen Körpern auf und befinden sich im Gleichgewicht mit deren inneren Spannungen. Der klassische Ansatz zur Analyse von Berührungsvorgängen basiert daher auf der Festkörpermechanik. Die grundlegenden Zusammenhänge bei der Verformung elastischer Körper beschreibt die Elastizitätstheorie. Für kontaktmechanische Untersuchungen werden darüber hinaus geeignete Gesetzmäßigkeiten für die Übertragung von Spannungen im Berührgebiet benötigt (s. Abschnitt 3.1), welche hier als Randbedingungen eingehen.

3.3.1 Elastizitätstheorie

Die Verformung eines elastischen Körpers kann durch seinen *Verzerrungszustand* dargestellt werden. Die Dehnung ε beschreibt die Längenveränderung und die Scherung γ die Winkelveränderung eines Volumenelements. Für kartesische Koordinaten gilt mit den Verschiebungen u , v und w in x -, y - und z -Richtung:

$$\begin{aligned}
 \varepsilon_x &= \frac{\partial u}{\partial x} & \gamma_{xy} = \gamma_{yx} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\
 \varepsilon_y &= \frac{\partial v}{\partial y} & \gamma_{xz} = \gamma_{zx} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\
 \varepsilon_z &= \frac{\partial w}{\partial z} & \gamma_{yz} = \gamma_{zy} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}
 \end{aligned}
 \tag{3.8}$$

Durch geometrische Betrachtung und Umformung lässt sich leicht zeigen, dass die Elemente des Verzerrungstensors den *Kompatibilitätsbedingungen* genügen müssen [47]:

$$\begin{aligned}
\frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial y} & 2 \frac{\partial^2 \varepsilon_x}{\partial y \partial z} &= \frac{\partial}{\partial x} \left(-\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right) \\
\frac{\partial^2 \varepsilon_y}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial y^2} &= \frac{\partial^2 \gamma_{yz}}{\partial y \partial z} & 2 \frac{\partial^2 \varepsilon_y}{\partial z \partial x} &= \frac{\partial}{\partial y} \left(-\frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial x} \right) \\
\frac{\partial^2 \varepsilon_z}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial z^2} &= \frac{\partial^2 \gamma_{zx}}{\partial z \partial x} & 2 \frac{\partial^2 \varepsilon_z}{\partial x \partial y} &= \frac{\partial}{\partial z} \left(-\frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} \right)
\end{aligned} \tag{3.9}$$

Auch die den Verzerrungen entsprechenden Spannungen σ und τ sind nicht beliebig, sondern müssen den *dynamischen Grundgleichungen* mit den Volumenkräften f und der Massendichte ρ folgen:

$$\begin{aligned}
\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + f_x &= \rho \ddot{u} \\
\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + f_y &= \rho \ddot{v} \\
\frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + f_z &= \rho \ddot{w}
\end{aligned} \tag{3.10}$$

Um den Verzerrungs- und Spannungszustand eines Körpers bestimmen zu können, wird neben diesen allgemein gültigen partiellen Differentialgleichungen ein Materialgesetz benötigt, welches die Verformung mit den Spannungen verknüpft. In den meisten Fällen, auch in der Kontaktmechanik, wird das *Hookesche Gesetz* verwendet, welches linear-elastisches Materialverhalten modelliert:

$$\begin{aligned}
\varepsilon_x &= \frac{1}{E} \left[\sigma_x - \left(\frac{E}{2G} - 1 \right) (\sigma_y + \sigma_z) \right] & \gamma_{xy} &= \frac{1}{G} \tau_{xy} \\
\varepsilon_y &= \frac{1}{E} \left[\sigma_y - \left(\frac{E}{2G} - 1 \right) (\sigma_z + \sigma_x) \right] & \gamma_{yz} &= \frac{1}{G} \tau_{yz} \\
\varepsilon_z &= \frac{1}{E} \left[\sigma_z - \left(\frac{E}{2G} - 1 \right) (\sigma_x + \sigma_y) \right] & \gamma_{zx} &= \frac{1}{G} \tau_{zx}
\end{aligned} \tag{3.11}$$

Dabei wird von homogenem, isotropem Material ausgegangen, dessen Elastizität alleine durch den Elastizitätsmodul E und den Schubmodul G beschrieben wird; Materialdämpfung und Hystereseeffekte werden vernachlässigt.

Die klassische Aufgabenstellung bei der Anwendung der Elastizitätstheorie auf Kontaktprobleme ist die Bestimmung des Verzerrungs- und Spannungsfelds in einem *elastischen Halbraum* bei gegebenen Verschiebungen oder Spannungen (Randbedingungen erster bzw. zweiter Art) an seiner Oberfläche (Abbildung 3.7). Bei dreidimensionaler Betrachtung lassen sich mit Hilfe der *Cerutti-Potentialfunktion* für nur wenige, sehr einfache Fälle geschlossene Lösungen angeben, z. B. achsensymmetrische Probleme, Einzelkräfte, Hertzscher Kontakt und linearer Spannungsverlauf in polygonalem Gebiet [48].

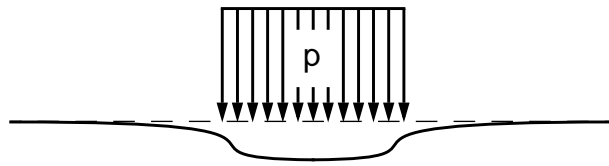


Abbildung 3.7: Druckverteilung auf elastischem Halbraum

3.3.2 Finite-Elemente-Methode

Die Methode der Finiten Elemente (FEM) ist ein weit verbreitetes Verfahren zur numerischen Lösung partieller Differentialgleichungen [49]. Eine klassische Anwendung ist die Spannungsanalyse von Bauteilen, bei der die erwähnten Gleichungen der Elastizitätstheorie für gegebene Belastungen bzw. Randbedingungen ausgewertet werden.

Als grundlegende Idee der FEM wird das betrachtete Gebiet in einfache Elemente wie z. B. Tetraeder oder Parallelepipede unterteilt, und die resultierenden partiellen Differentialgleichungen wandelt man in äquivalente Variationsgleichungen um. Für jedes Finite Element wird eine einfache Ansatzfunktion definiert, die den Verlauf des gesuchten Felds — hier der Verschiebungen u , v , w aus der Ruhelage — approximiert. Bei der Analyse des gesamten Bereichs sind nur die Knotenpunkte der Elemente relevant; in ihnen müssen die Ansatzfunktionen konform sein, d. h. im problemspezifisch erforderlichen Maße stetig verlaufen, oder zumindest den sog. Patch-Test [50] bestehen. Setzt man die durch die Ansatzfunktionen approximierten Feldgrößen in die Variationsgleichungen ein, erhält man für statische Probleme ein algebraisches Gleichungssystem, für dynamische ein Differentialgleichungssystem für die Knotenvariablen bzw. deren Zeitverlauf. Dieses wird bestimmt und damit numerisch lösbar, wenn Rand- und ggf. Anfangsbedingungen vorgegeben werden.

Die FEM stellt zweifellos das mächtigste Werkzeug zur Analyse von Kontaktvorgängen dar. Dazu werden elastische FE-Modelle der kontaktierenden Körper erstellt und bei Berührung die Randbedingungen verändert, sodass das Prinzip der Impenetrabilität (s. Abschnitt 3.1.1) sowie ggf. Tangentialspannungen durch Reibung berücksichtigt werden [50, 51]. So lassen sich nahezu beliebig geformte Oberflächen darstellen, Spannungen und Wellenausbreitung innerhalb der Körper studieren und mit entsprechenden Materialgesetzen und speziellen, nichtlinearen Algorithmen auch plastische Verformungen analysieren. Im Vergleich zur MKS-Simulation sind typischerweise wesentlich größere Gleichungssysteme zu lösen, die allerdings eine einfachere Struktur aufweisen. Dennoch betragen die Rechenzeiten zur dynamischen

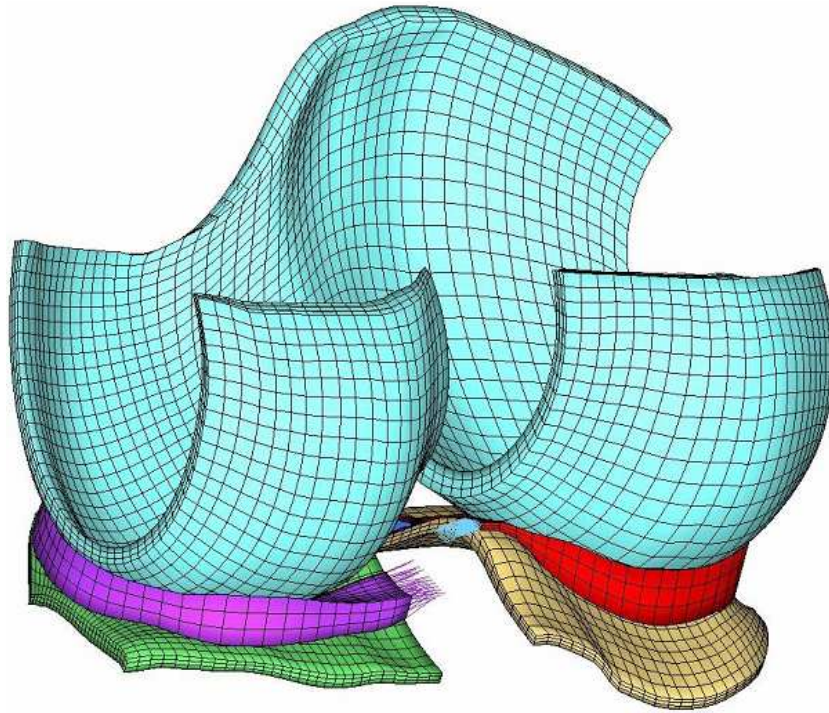


Abbildung 3.8: FEM–Modell eines Kniegelenks [52]

Simulation dreidimensionaler Kontaktprobleme auf aktuellen Workstations üblicherweise mehrere Stunden.

P. Eberhard befasst sich in seiner Habilitationsschrift [25] sehr ausführlich mit der FEM–Analyse von Kontaktvorgängen innerhalb von MKS–Simulationen. Die Arbeit macht deutlich, dass solche hybriden Methoden und Algorithmen extrem komplex und aufwändig sind: Neben tiefem Verständnis für MKS und FEM müssen auch nichtlineare Strukturmechanik, Geometrie und Numerik beherrscht werden. Daraus entsteht der nicht zu unterschätzende Nachteil, dass ein unerfahrener Anwender nicht mehr in der Lage sein kann, unerwartete Ergebnisse zu interpretieren. Darüber hinaus machen die Angaben zur Rechenzeit für ein relativ einfaches 2D–Beispiel deutlich, dass die FEM–Kontaktanalyse den üblichen Rahmen von MKS–Simulationen bei Weitem sprengt.

3.4 Oberflächenmodelle

In diesem Abschnitt werden Verfahren beschrieben, die Berührungen durch Modellierung der Vorgänge im Kontaktgebiet an den Oberflächen der Körper darstellen. Dabei geht man stets davon aus, dass das Körperinnere im Sinne des Prinzips von St. Venant (s. Abschnitt 3.1.4) als unbeteiligt bzw. starr betrachtet werden kann, und dynamische Effekte aufgrund von Beschleunigungen bei der Verformung werden vernachlässigt.

3.4.1 Halbraumapproximation

Die Halbraumapproximation [16, 17, 48, 53, 54] ist für annähernd homogene, isotrope, linear-elastische Körper (deren Materialverhalten allein durch zwei Stoffkonstanten, z. B. Schubmodul G und Querkontraktionszahl ν , beschrieben wird) anwendbar, wenn das Kontaktgebiet relativ klein gegenüber ihren charakteristischen Abmessungen ist. Dann lässt sich das elastische Feld im Nahbereich der Berührfläche in guter Näherung durch einen elastischen Halbraum ersetzen. Man geht vom (im Abschnitt 3.1.1 definierten) Abstandsprofil $g(x, y)$ der unverformten Oberflächen in einem Koordinatensystem aus, dessen z -Achse normal zur angenommenen Berührebene verläuft. Im Weiteren wird ein Ersatzmodell betrachtet, in dem ein starrer Körper mit dem Abstandsprofil als Oberfläche in einen linear-elastischen Halbraum (Abbildung 3.9) mit kombinierten Stoffkonstanten eindringt [14]¹:

$$G^* = \frac{1}{\frac{1}{G_1} + \frac{1}{G_2}} \quad (3.12)$$

$$\nu^* = \left(\frac{\nu_1}{G_1} + \frac{\nu_2}{G_2} \right) G^* \quad (3.13)$$

Der Elastizitätsmodul lässt sich am einfachsten mit dem bekannten Zusammenhang $E^* = 2G^*(1 + \nu^*)$ ermitteln. Ferner dient die Größe

$$\kappa = \left(\frac{1 - 2\nu_1}{G_1} - \frac{1 - 2\nu_2}{G_2} \right) \frac{G^*}{4} \quad (3.14)$$

als Maß für die elastische Abweichung der Materialien ($|\kappa| \leq 1$): Ist $\kappa = 0$, nennt man die Paarung *quasiidentisch*, d. h. eine gegebene Spannungsverteilung bewirkt dasselbe Tangentialverformungsfeld im Kontaktgebiet.

¹In [16, 17] wird mit den doppelt so großen Werten $G = 2G^*$, $\nu = 2\nu^*$ gearbeitet

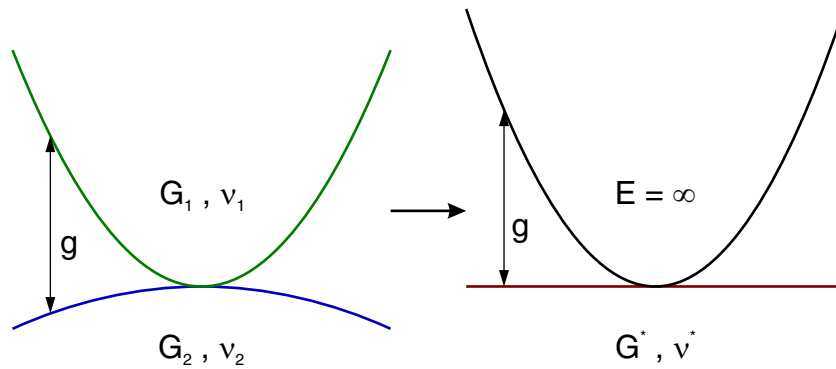


Abbildung 3.9: Übergang von der Kontaktpaarung zur Halbraumapproximation

Gesucht werden die im Ersatzmodell auftretenden Verformungen und Spannungen, um sie auf das Kontaktproblem zurückzuübertragen. Als theoretische Grundlage dient die Deformation des Halbraums aufgrund einer an der Oberfläche wirkenden Normalkraft P entsprechend der Elastizitätstheorie (s. Abschnitt 3.3.1):

$$u_n(x, y) = \frac{1 - \nu}{2\pi G} \frac{P}{r} \quad (3.15)$$

Dabei steht $r = \sqrt{(x - x_P)^2 + (y - y_P)^2}$ für den Abstand zwischen dem betrachteten Ort und dem Angriffspunkt (Abbildung 3.10). Ersetzt man P durch das Produkt einer Normalspannung p mit einem Flächenelement dA , lässt sich u_n für eine beliebige Druckverteilung im Wirkgebiet S aufintegrieren [14]:

$$u_n(x, y) = \frac{1 - \nu}{2\pi G} \iint_S \frac{p(x_p, y_p)}{r} dx_p dy_p \quad (3.16)$$

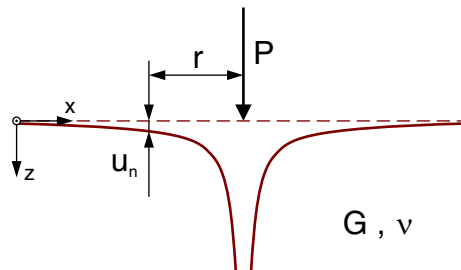


Abbildung 3.10: Deformation u_n eines elastischen Halbraums infolge einer Normalkraft P

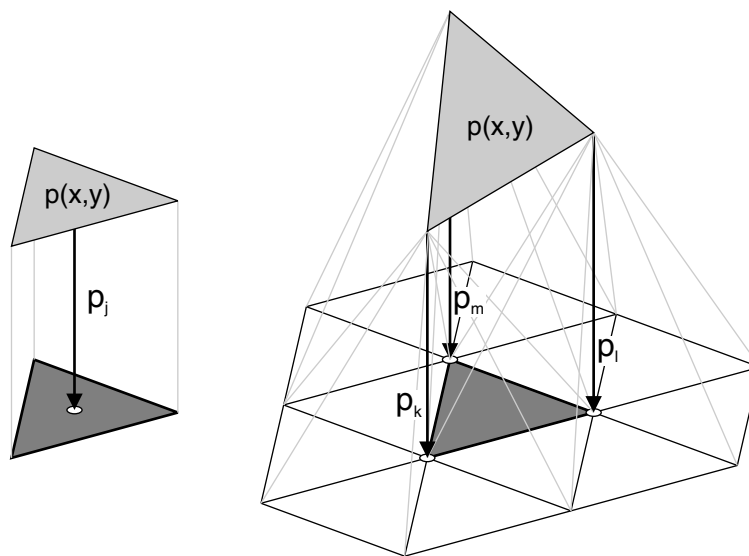


Abbildung 3.11: Traktionselement als gleichseitiges Dreieck mit konstanter bzw. linearer Spannungsverteilung

Ähnlich wie bei der FEM wird das Gebiet zur numerischen Analyse räumlich diskretisiert. Die dabei entstehenden *Traktionselemente* sind jedoch zweidimensionale Gebilde, die an der Oberfläche der Halbebene liegen. Gebräuchlich sind Dreiecke oder Rechtecke, deren Schwer- oder Eckpunkte als *Matching Points* definiert werden.

Außerdem ist eine mathematische Darstellung zur Approximation des Spannungsverlaufs in den Elementen festzulegen. Das günstigste Verhältnis zwischen Näherungsgüte und Aufwand weisen i. Allg. Elemente mit konstanter Spannung auf (Abbildung 3.11 links), die zu einer stufenförmigen Approximation des gesamten Gebiets führen [16, 17]. Ein stetiger Spannungsverlauf kann durch Elemente mit linearer Verteilung erreicht werden [54]. Abbildung 3.11 rechts zeigt z. B., wie sich p bei einer aus gleichseitigen Dreiecken gebildeten Diskretisierung durch Superposition von pyramidenförmig über den Eckpunkten angenommenen Druckverläufen ergibt [53].

Damit sind die Voraussetzungen geschaffen, um mit Hilfe von (3.16) die Verformung u_i an einem Matching Point I aufgrund der Spannung p_j an einem weiteren Matching Point J in der Form

$$u_i = c_{ij} \cdot p_j \quad (3.17)$$

zu berechnen. Für die üblichen Traktionselemente lassen sich die *Einflusskoeffizienten* c_{ij} analytisch darstellen, die Ausdrücke sind jedoch recht unhandlich

[16, 53, 54]. Wegen der Linearität des Modells kann man die Verformungen u_i aufsummieren und erhält in Matrixschreibweise:

$$\mathbf{u} = \mathbf{C} \cdot \mathbf{p} \quad (3.18)$$

Der Vektor \mathbf{p} enthält die Spannungen und \mathbf{u} die Verformungen in den Matching Points. \mathbf{C} ist die Matrix der Einflusskoeffizienten c_{ij} , die allein von Typ und Abmessungen der gewählten Traktionselemente abhängt und somit nur einmal berechnet werden muss (solange die Diskretisierung unverändert bleibt).

Das Kontaktproblem wird fortan nur noch in den Matching Points betrachtet. (3.18) gibt die Einsinkungen für eine gegebene Normalspannungsverteilung an. Gesucht sind jedoch die Spannungen, die sich aufgrund der relativen Lage zwischen dem Starrkörper und der Halbebene einstellen. Damit geht die Ermittlung des Berührgebiets mit $p_j > 0$ einher, während außerhalb $p_j = 0$ als Randbedingung gilt.

Eine Möglichkeit zur Lösung des Problems basiert auf Inversion der Matrix \mathbf{C} , sodass (3.18) die Spannungen für einen angenommenen Verformungszustand liefert [48, 53]. Als Startwert für \mathbf{u} bietet sich die Eindringtiefe in die unverformte Halbebene an. Da das tatsächliche Kontaktgebiet i. d. R. kleiner ist (Abbildung 3.5), ergeben sich am Rand Zugspannungen. In weiteren Iterationsschritten setzt man dort $p_j = 0$, sodass die entsprechenden Matching Points außerhalb des Kontaktgebiets angenommen werden. Die Lösung ist gefunden, wenn sich ein konsistenter Zustand ohne Verletzung der Impenetrabilität oder Adhäsionsfreiheit einstellt (s. Abschnitt 3.1). Dafür sind typischerweise zwei oder drei Iterationsschritte erforderlich. Bei sehr feiner Diskretisierung wird \mathbf{C} jedoch nahezu singulär und bei der Lösung des Gleichungssystems treten numerische Schwierigkeiten auf [48].

Eine elegantere Methode bedient sich der Variationsrechnung [16, 17, 48, 53, 54]: Unter Berücksichtigung der Nebenbedingungen Impenetrabilität (3.1) und Adhäsionsfreiheit (3.2) wird eine Spannungsverteilung $p(x, y)$ bestimmt, die die Kontaktenthalpie

$$V = \frac{1}{2} \iint_S p(x, y) u_n(x, y) \, dA + \iint_S p(x, y) d(x, y) \, dA \quad (3.19)$$

minimiert. Hierbei steht der erste Summand für die innere Verformungsenergie des Halbraums und der zweite für die äußere Energie, die zum Einpressen des Körpers benötigt wird (wobei d die Eindringtiefe in den unverformten Halbraum darstellt). In diskretisierter Form erhält man mit (3.17):

$$V = a \left(\frac{1}{2} \sum_{i,j} p_i c_{ij} p_j A_i + \sum_i p_i d_i A_i \right) \quad (3.20)$$

Der Faktor a entspricht dem Verhältnis aus Volumen und Grundfläche der gewählten Traktionselemente. Durch (3.20) lassen sich Druckverteilung und Kontaktgebiet mit einem Standardverfahren zur Quadratischen Optimierung direkt und eindeutig berechnen.

Die Halbraumapproximation ist grundsätzlich auch für Kontaktanalysen mit Berücksichtigung Coulombscher Reibung geeignet [17, 48]. Dabei sind zusätzlich Traktionselemente in tangentialer Richtung zu definieren, und analog zur Ermittlung des Kontaktgebiets müssen anhand von (3.5), (3.6) Haft-/Gleitzustände und ggf. Gleitrichtungen bestimmt werden (s. Abschnitt 3.2). Sind die elastischen Eigenschaften der kontaktierenden Körper nicht quasi-identisch ($\kappa \neq 0$ nach (3.14)), erhöht sich die Komplexität des Problems weiter durch Kopplung der Normal- und Tangentialspannungen. Zur numerischen Lösung sind nur noch iterative Algorithmen praktikabel, die auf der Variationsrechnung basieren [17].

Abgesehen von hochspezialisierten Modulen für den Radkontakt von Schienenfahrzeugen [17] sind dem Autor keine Implementierungen des Halbraummodells in MKS-Programmen bekannt. Die Methode erscheint jedoch aufgrund ihrer recht hohen Allgemeinheit bei moderater Komplexität interessant für dieses Umfeld. Als Schwachpunkt ist die nicht näher spezifizierte Wahl der Berührebene zu nennen, durch die implizit die Richtung der Kontaktkraft festgelegt wird.

3.4.2 Randschichtmodell

Ähnlich wie bei der Halbraumapproximation wird beim Randschichtmodell [17, 53] mit einem Ersatzmodell gearbeitet, bei dem ein starrer Körper mit dem aufsummierten Profil der Kontaktpaarung in eine elastische Ebene eindringt. Man geht jedoch nicht von einem elastischen, sondern ebenfalls starren Halbraum aus, der mit einer dünnen, linear-elastischen Randschicht überzogen ist. Eine erhebliche Vereinfachung ergibt sich durch Vernachlässigung der Scherspannungen: Anstelle einer Einflussfunktion (3.16), bei der die Verformung durch Integration über das gesamte Berührgebiet ermittelt werden muss, besteht ein linearer Zusammenhang zwischen Einsinkung und Druckspannung am selben Ort:

$$u_n(x, y) = \frac{b}{K} \cdot p(x, y) \quad (3.21)$$

Dabei steht b für die Dicke und

$$K = \frac{1 - \nu}{(1 + \nu)(1 - 2\nu)} \cdot E \quad (3.22)$$

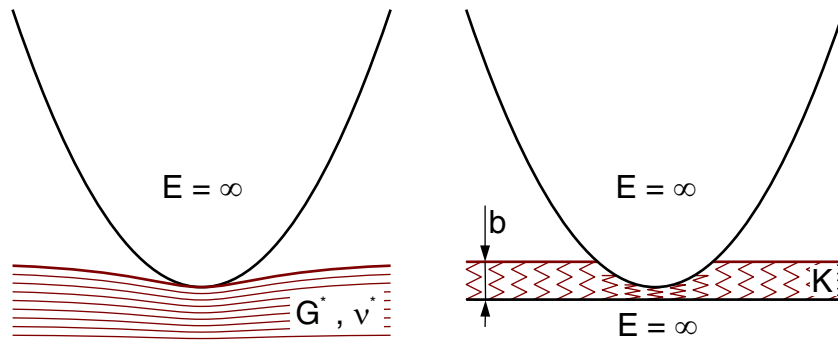


Abbildung 3.12: Vergleich von Halbraumapproximation und Randschichtmodell

für den elastischen Modul der Randschicht, die man sich vorstellen kann wie eine Matratze, deren Federn unabhängig voneinander wirken. Abbildung 3.12 veranschaulicht den wesentlichen Unterschied zur Halbraumapproximation.

Mit (3.21) lässt sich auch unmittelbar die Spannung aufgrund der Verformung, welche hier mit der Eindringtiefe des unverformten Profils übereinstimmt, berechnen. Es muss jedoch beachtet werden, dass das Modell nur für Kontaktpaarungen mit $\nu < 0,45$ sinnvolle Ergebnisse liefert [17]. Weiter resultiert die fehlende Absenkung des Kontaktbereichs (Abbildung 3.12) in einer fehlerhaften Versteifung, die sich bei bekanntem Profil durch einen Korrekturfaktor für K vermeiden lässt [53]; beispielsweise liefert ein Vergleich mit dem Hertzschen Kontakt in [17] in etwa die doppelte Gesamtsteifigkeit.

Das Randschichtmodell ist völlig analog auch in tangentialer Richtung anwendbar, sofern eine quasiidentische Materialpaarung ($\kappa = 0$ nach (3.14)) vorliegt [17]. Durch entsprechende Anpassung von (3.21) kann man außerdem in einfacher Weise andere rheologische Modelle zugrunde legen, um etwa viskoelastisches oder plastisches Materialverhalten darzustellen. Wie bei der Halbraumapproximation ist aber auch hier unklar, wie die Berührebene für beliebige Oberflächengeometrien zu definieren ist. Darüber hinaus muss der Faktor b/K durch Versuche oder Abschätzung bestimmt werden, sofern nicht tatsächlich eine dünne elastische Randschicht vorliegt.

In [26] wird ein heuristisch motivierter Algorithmus für die MKS-Simulation vorgestellt, welcher die Kontaktanalyse in Anlehnung an das Randschichtmodell implementiert. Hier wird das Schnittvolumen V der unverformten Körperoberflächen, die durch Polygonflächen beschrieben werden, zur Ermittlung der Kontaktnormalkraft herangezogen:

$$F_n = c_V V + d_V \dot{V} \quad (3.23)$$

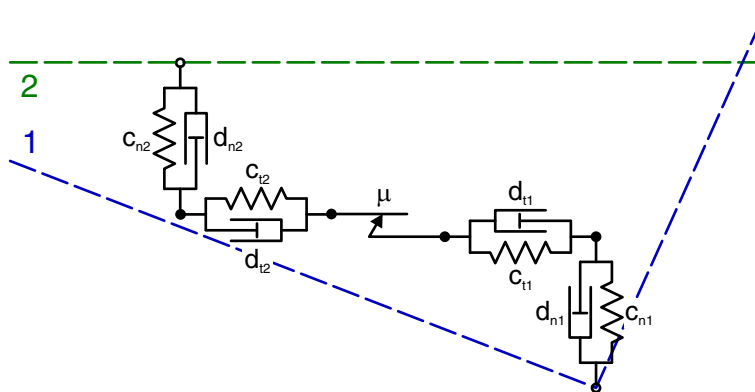


Abbildung 3.13: Beim Kontaktmodell nach [21, 22] werden aufwändige Kraftelemente zwischen eingedrungene Ecken (Körper 1) und mittels Eigendynamik bestimmte Punkte (Körper 2) geschaltet

Mit $V = \iint_S u_n dA$ und $c_V = K/b$ lässt sich der elastische Anteil als Randschichtmodell nach (3.21) interpretieren, und der Term $d_V \dot{V}$ sorgt analog für viskose Dämpfung. Als Angriffspunkt für den Kraftvektor dient der Schwerpunkt des Schnittvolumens; seine Richtung wird allerdings ohne Berücksichtigung physikalischer Zusammenhänge durch die relative Orientierung der konvexen Hüllen der Oberflächen bestimmt.

Auch das in [21, 22] beschriebene Verfahren basiert auf der Idee starrer Körper mit viskoelastischer Randschicht und geht von Körpern in Form konvexer Polyeder aus. Es werden die Kombinationen Ecke–Kante und Kante–Kante zwischen den Schnittgebieten der unverformten Polygonflächen betrachtet. Wenn eine neue Paarung dieser Typen auftritt, erzeugt der Algorithmus ein Kraftelement (s. Abschnitt 3.5.1), in dem Normalkräfte durch in Reihe geschaltete lineare Feder–Dämpfer und Tangentialkräfte durch eine Reihenschaltung von Elementen, die Feder–Dämpfer, Coulombsche Reibung und wiederum Feder–Dämpfer repräsentieren, dargestellt werden (Abbildung 3.13). Da für diese Kontaktkraftelemente mit Eigendynamik bis zu vier Zustandgrößen (zur Beschreibung der Längen der Feder–Dämpfer) benötigt werden, erhält man ein strukturvariantes MKS, zu dessen Lösung ein Integrationsverfahren mit zustandsabhängigen Schaltfunktionen erforderlich ist (s. Abschnitt 2.3). Dieser Aufwand des ansonsten interessanten Ansatzes erscheint in Anbetracht des fehlenden quantitativen Zusammenhangs zu (3.21) unverhältnismäßig groß.

Eine vereinfachte und somit ausgewogenere Variante, die auch konkav geformte Körperoberflächen verarbeiten kann, wird in [23, 24] vorgestellt. Hier

wird das viskoelastische Materialverhalten der Kontaktpaarung durch ein nichtlineares Feder–Dämpfer–Element in Normalenrichtung und das durch Reihenschaltung von Feder–Dämpfer und Coulombscher Reibung gebildete Bürstenmodell (s. Abschnitt 3.5.1) in Tangentialrichtung beschrieben.

Die zuvor genannten Kontaktalgorithmen sind dem in Kapitel 4 entwickelten Verfahren ähnlich. Darüber hinaus wird das Randschichtmodell sehr erfolgreich zur Beschreibung des Radkontakts von Schienenfahrzeugen angewendet [17].

3.5 Eindimensionale Modelle

Während im Abschnitt 3.3 Kontaktmodelle behandelt wurden, bei denen die beteiligten Körper in ihrer Gesamtheit beschrieben werden, beschränken sich die Verfahren aus Abschnitt 3.4 auf Betrachtung ihrer Oberflächen. Im Folgenden werden Methoden vorgestellt, die noch weiter vereinfachen, indem sie den Kontakt anhand eindimensionaler physikalischer Gesetzmäßigkeiten darstellen. Voraussetzung für ihre Anwendung ist ein repräsentativer Angriffspunkt mit zugehöriger Wirkrichtung. Im Abschnitt 3.5.3 wird auf dieses Problem der Kontaktkinematik eingegangen.

3.5.1 Kraftelemente

Zur Analyse von Kontaktvorgängen in MKS kommen meist Component–Krafterelemente (s. Abschnitt 2.1) zum Einsatz, deren richtungsbestimmendes Koppelsystem M_i an der Oberfläche eines der beiden Körper liegt und so orientiert ist, dass eine bestimmte Achse in Normalenrichtung verläuft. Der zweite Koppelpunkt M_j befindet sich auf der Oberfläche des anderen Körpers und auf der Normale von M_i (Abbildung 3.14).

Die Kontaktnormalkraft wird anhand des Abstands a und der Relativgeschwindigkeit \dot{a} der beiden Koppelpunkte entlang der Normalenrichtung bestimmt:

$$F_n = f(a, \dot{a}) \quad (3.24)$$

Dieses Kraftgesetz gestaltet sich besonders einfach, wenn die Normalkraft als einseitige Parallelschaltung einer linearen Feder und eines linearen Dämpfers modelliert wird:

$$F_n = \begin{cases} 0 & : a \geq 0 \\ ca + d\dot{a} & : a < 0 \end{cases} \quad (3.25)$$

Eine bessere Approximation kann man mit nichtlinearen Kennlinien oder -feldern erreichen. Sind etwa die Hauptkrümmungen der Oberflächen an den Koppelpunkten bekannt, kann der elastische Anteil anhand der Formeln zur

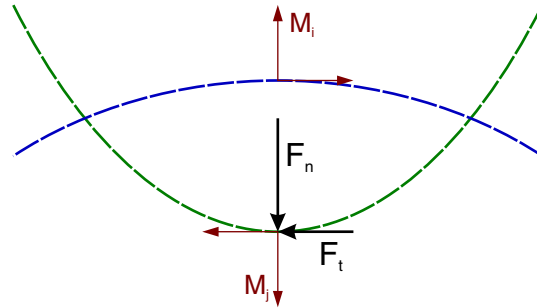


Abbildung 3.14: Anordnung eines Kontakt-Kraftelements; auf den oberen Körper wirken die Kräfte am selben Ort in entgegengesetzter Richtung

Hertzschen Pressung bestimmt werden [19]. Der dämpfende Anteil wird oft so modifiziert, dass beim Durchlaufen von $a = 0$ kein Kraftsprung und beim Ausfedern keine Zugkraft auftreten kann [18, 19, 55, 56].

Auch Reibungskräfte lassen sich anhand eines wie in Abbildung 3.14 angeordneten Kraftelements darstellen. Als zustandsabhängige Eingangsgrößen dienen dabei der tangentialer Anteil der Relativgeschwindigkeit v_t und die mit (3.23) ermittelte Normalkraft F_n :

$$F_t = f(v_t, F_n) \quad (3.26)$$

Der Reibungskraftvektor mit dem Betrag F_t wird so in die beiden Körper eingepreßt, dass er der tangentialen Relativbewegung entgegenwirkt.

Eine ideale Umsetzung Coulombscher Reibung (3.1) ist mit Kraftelementen prinzipiell nicht möglich, weil der Haftzustand nur durch Zwangskräfte, wie sie in Gelenken vorkommen, darstellbar ist. Für (3.26) gibt es daher Ansätze verschiedener Komplexität, die die Charakteristik aus Abbildung 3.4 näherungsweise abbilden. Abbildung 3.15 zeigt beispielsweise Approximationen anhand einer skalierten Arcustangensfunktion, einer hyperbolischen Tangensfunktion [19] sowie mit linearer [18] bzw. quadratischer Interpolation unterhalb einer kleinen Gleitgeschwindigkeit v_ε . Hierbei werden die Zwangskräfte im Haftbereich durch Dämpfung ersetzt, welche von der entsprechenden Gleitreibungskraft μF_n und der Grenzgeschwindigkeit v_ε bestimmt wird.

Ein aufwändigeres Verfahren implementieren Reibkraftelemente mit Strukturumschaltung [18, 24]: Nach Unterschreitung der Grenzgeschwindigkeit v_ε wirkt hier ein Point-to-Point-artiges Feder-Dämpfer-Element zwischen den Koppelmarkern. Wenn die Haftkraftreserve $\mu_0 F_n$ überschritten wird, wird wieder in den Gleitzustand zurückgeschaltet. Dieses Kraftgesetz wird als *Bürstenmodell* mit *Mikroschlupf* bezeichnet und eignet sich auch zur Darstellung von dauerhaften Haftzuständen und *Stick-Slip*-Effekten.

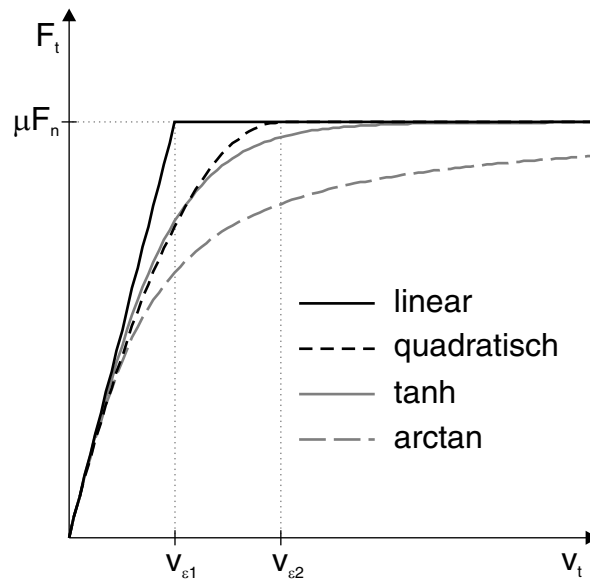


Abbildung 3.15: Funktionen zur Approximation Coulombscher Reibung, Skalierung mit gleicher Steigung bei $v_t = 0$

Die Kontaktmodellierung mit Kräftelementen bleibt qualitativ natürlich deutlich hinter Oberflächen- oder gar Festkörpermodellen zurück. In vielen Fällen gestalten sich die Reduktion der Berührfläche auf einen Punkt und die Bestimmung der Kraftgesetzparameter problematisch. Dennoch kann sie für spezielle Anwendungen (z. B. Hertzscher Kontakt mit reiner Gleitreibung) optimale Ergebnisse bei minimalem Aufwand liefern. Die starke Vereinfachung erleichtert zudem erheblich die Interpretation von Simulationsergebnissen.

3.5.2 Starrkörperkontakt

Bei der am weitesten abstrahierenden Darstellung von Kontaktvorgängen wird von völlig starren Körpern ausgegangen. d. h. es gibt keine Verformungen der Oberflächen [11, 12, 25, 55]. Daraus folgen *mengenwertige* Kraftgesetze, die durch „eckige“ Kennlinien (Abbildung 3.16) charakterisiert sind: In Normalenrichtung stellt sich bei Kontakt ($a = 0$) eine Zwangskraft ein, die das Eindringen der Körperoberflächen verhindert. Im Sinne der MKS-Sprechweise liegt ein einseitiges Gelenk vor.

Beim Starrkörperkontakt wird auch in tangentialer Richtung mit kinematischen Bindungen gearbeitet. Damit lässt sich das Coulombsche Reibgesetz (Abbildung 3.4) direkt umsetzen: Für $v_t = 0$ wirkt eine Zwangskraft, die

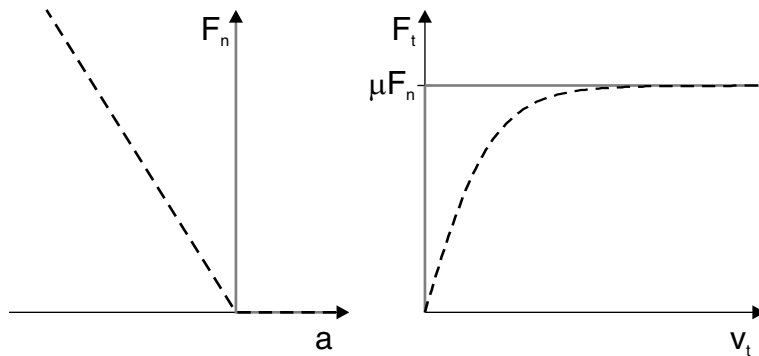


Abbildung 3.16: Kennlinien für Starrkörperkontakt (voll) und Kontaktkraft-elemente (unterbrochen)

Gleitbewegungen verhindert, solange die Haftkraftreserve $\mu_0 F_n$ nicht überschritten wird, sonst greift eine eingepreßte Kraft μF_n entgegen der Gleitbewegung an. Weil die Bindung im Haftzustand auf Geschwindigkeitsebene wirksam ist, liegt dann ein nichtholonomes Gelenk vor (s. Abschnitt 2.2).

Finden bei einer Simulation im Zeitbereich Übergänge zwischen Kontakt und Separation bzw. Haften und Gleiten statt, ändert sich die Anzahl der Bindungen. Damit wird das MKS und die zugehörigen Bewegungsgleichungen strukturvariant, was bei der Implementierung zu einem deutlich erhöhten Verwaltungsaufwand führt. Außerdem muss das Integrationsverfahren mit zustandsabhängigen Schaltfunktionen umgehen können (s. Abschnitt 2.3).

Eine besondere Behandlung erfordern Kollisionen, d. h. Zustandsänderungen von Separation zu Kontakt. Da die Körper sich nicht elastisch verhalten und bei Annäherung Kontaktkräfte aufbauen können, muss die Kontaktdynamik im Augenblick der Kollision als *Stoß* ohne zeitliche Ausdehnung beschrieben werden. Zur Modellierung greift man auf eine Stoßhypothese zurück; beispielsweise gibt das *Poissonsche Stoßgesetz* das Verhältnis der Normalimpulse der Kompressions- und Expansionsphase der Kollision an [55]:

$$\frac{\Lambda_{Exp}}{\Lambda_{Kom}} = \varepsilon \quad (3.27)$$

Das elastische Verhalten und die Dämpfung der Kontaktpaarung wird also allein durch den Stoßkoeffizienten ε beschrieben.

Durch die Zusammenfassung der Kollision in unendlich kurze Zeit gehen die Kontaktkräfte in Kraftimpulse über, welche sprunghafte Änderungen in den Geschwindigkeiten der zusammenprallenden Körper bewirken. Die numerische Integration der Bewegungsgleichungen muss daher unterbrochen und

nach Anwendung des Stoßgesetzes mit entsprechend verändertem Systemzustand auf Geschwindigkeitsebene fortgesetzt werden.

Treten im MKS mehrere Starrkörperkontakte auf, muss außerdem nach jedem Strukturwechsel eine konsistente Kombination der Kontaktzustände bestimmt werden, denn ein Impuls kann z. B. zum Abheben oder Losreißen aus dem Haftzustand eines anderen Kontakts führen. Diese Aufgabe kann im zweidimensionalen Fall als *lineares Komplementaritätsproblem* (LCP) formuliert und mit Standardalgorithmen gelöst werden [11].

Obwohl Starrkörperkontakt die am stärksten vereinfachende Modellierung von Kontaktvorgängen darstellt, ist seine numerische Implementierung wesentlich komplexer als etwa die Verwendung von Kraftelementen. Für die Übergänge zwischen den Kontaktzuständen müssen bei der Integration Schaltfunktionen berücksichtigt werden, die strukturvarianten Bewegungsgleichungen ziehen einen deutlich erhöhten Verwaltungsaufwand nach sich und zur Bestimmung konsistenter Kontaktzustände wird ein LCP-Löser benötigt. Darüber hinaus können sich selbst bei zweidimensionaler Betrachtung einfacher Systeme singuläre Zustände einstellen [11, 55], und zur allgemeinen Behandlung räumlicher Probleme mit Reibung fehlt bisher das theoretische Fundament. In vielen Anwendungen verbietet sich die Anwendung des Starrkörperkontakts auch grundsätzlich, weil die Annäherung der Körper einen wesentlichen Einfluss auf die Systemdynamik hat. So bewegt sich z. B. die elastische Einfederung in einem hoch belasteten Wälzlager in derselben Größenordnung wie das Lagerspiel. Als wesentlicher Vorteil gegenüber den nachgiebigen Modellen ist allerdings zu nennen, dass keine hohen Steifigkeiten auftreten, die die Effizienz der numerischen Zeitintegration erheblich beeinträchtigen (s. Abschnitt 2.3).

3.5.3 Kontaktkinematik

Um eindimensionale Kontaktmodelle überhaupt anwenden zu können, muss man auf den berührenden Körpern *Kontaktpunkte* und eine zugehörige Normalenrichtung bzw. Tangentialebene vorgeben. Diese geometrischen Annahmen haben erheblichen Einfluss auf die Systemdynamik, weil sie sowohl die Grundlage zur Bestimmung der kinematischen Eingangsgrößen für das Kraft- oder Stoßgesetz darstellen, als auch Angriffspunkt und Wirkrichtung der Kontaktkräfte festlegen.

Zahlreiche Arbeiten und Algorithmen basieren auf der Ermittlung *extremaler* Kontaktpunkte gemäß Abbildung 3.17: Auf den Körperoberflächen wird jeweils derjenige Punkt bestimmt, der den minimalen Abstand (bei Separation) bzw. die maximale senkrechte Eindringtiefe (bei Überschneidung) ergibt. Im Grenzfall (Abstand $a = 0$) entsprechen beide Punkte dem Berühr-

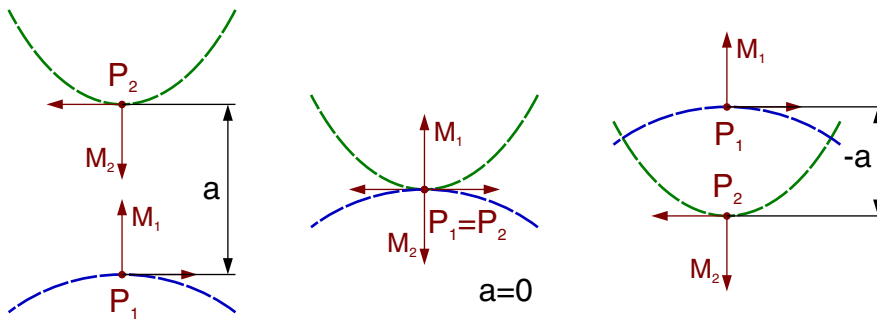


Abbildung 3.17: Lage extremaler Kontaktpunkte bei Separation, Berührung und Überschneidung

punkt, und die Normalenrichtung verläuft senkrecht zur Berührebene. Da die extremalen Punkte auch zwischen separierten Körpern definiert sind, werden sie mitunter als *potentielle* Kontaktpunkte bezeichnet; ihr Abstand a wird meist mit negativem Vorzeichen versehen, wenn sich die Oberflächen überschneiden.

Haben die Oberflächen primitive Geometrien wie z. B. Ebenen, Kugel-, Zylinder- oder Kegelflächen, lassen sich die Positionen der extremalen Kontaktpunkte durch einfache geometrische Überlegungen direkt analytisch berechnen [19]. Ansonsten greift man auf parametrisch dargestellte Flächen zurück, bei denen die Lage des Kontaktpunkts durch eine Abbildung $\mathbf{x} = f(u, v)$ vom zweidimensionalen Parameterraum (u, v) in den euklidischen, körperfesten Bildraum beschrieben wird [55, 57]. In der Regel werden stetig differenzierbare Flächen vorausgesetzt, sodass zu jedem Punkt ein begleitendes Dreibein $(\mathbf{n}, \mathbf{s}, \mathbf{t})$ existiert, das die zugehörige Normale und Tangentialebene definiert (Abbildung 3.18). Dann genügen extremale Punkte folgenden Bedingungen:

$$\begin{aligned}
 \mathbf{r}_{P_1P_2} \cdot \mathbf{s}_1 &= 0 \\
 \mathbf{r}_{P_1P_2} \cdot \mathbf{t}_1 &= 0 \\
 \mathbf{n}_2 \cdot \mathbf{s}_1 &= 0 \\
 \mathbf{n}_2 \cdot \mathbf{t}_1 &= 0
 \end{aligned} \tag{3.28}$$

Die gleichen Beziehungen gelten für vertauschte Indizes 1 und 2. Um die vier Flächenparameter der extremalen Punkte für gegebene Relativlage und -orientierung der Körper zu bestimmen, sind die Gleichungen (3.28) jedoch ausreichend. Zur numerischen Lösung kann ein Algorithmus für nichtlineare Gleichungssysteme verwendet werden. Als effizientere Alternative bietet sich die Berücksichtigung der Gleichungen (3.28) als algebraische Nebenbedingungen eines DAE-Integrationsverfahrens an [3, 20, 58].

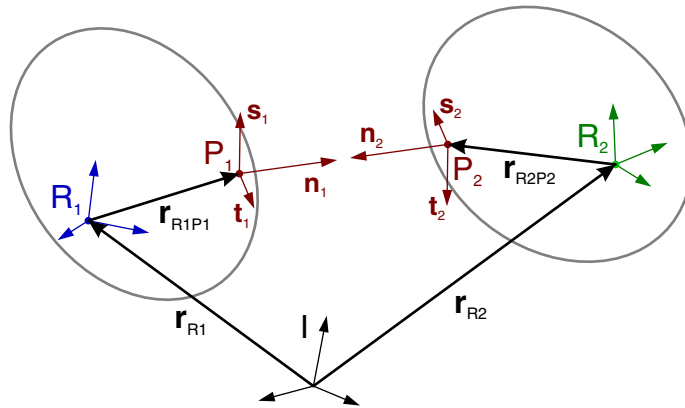


Abbildung 3.18: Kinematik extremer Kontaktpunkte

Dieses Verfahren wurde bereits für viele Flächentypen implementiert. In [18] findet man Spline-interpolierte und -approximierte sowie ganzrationale Flächen in kartesischen und Zylinderkoordinaten, rotierte Spline-Kurven und benutzerdefinierte Parametrierung. Recht aufwändig erweist sich die Verwendung von *NURBS*-Flächen [20, 59], die häufig zur Beschreibung von Freiformflächen in CAD-Programmen benutzt werden. In [60] werden neben parametrischen auch implizite und Polygonflächen behandelt.

Bei der Ermittlung der Relativgeschwindigkeit ist zu beachten, dass die Bewegung der Kontaktpunkte auf ihrem Körper rein zur Beobachtung dient, aber keine physikalische Bedeutung hat: Zu jedem Zeitpunkt markieren sie einen Ort auf der Oberfläche, der starr mit dem Körper geführt wird [12]. Deshalb darf bei der Berechnung ihrer Geschwindigkeit (maßgeblich für Dämpfungs-, Reibungs- und Stoßgesetze) nur ihre Starrkörpergeschwindigkeit, nicht aber ihre lokale Geschwindigkeit berücksichtigt werden:

$$\mathbf{v}_P = \mathbf{v}_R + \omega_R \times \mathbf{r}_{RP} + \underbrace{\mathbf{v}_{RP}}_{=0} \quad (3.29)$$

In der Praxis erweist sich die Bestimmung der extremalen Kontaktpunkte als sehr effizient, mitunter aber als schlechte Approximation und wenig robust oder gar nicht anwendbar. Die folgenden Schwierigkeiten waren eine wesentliche Motivation für die vorliegende Arbeit.

Liegt eine stark unsymmetrische Spannungsverteilung vor, kann der tatsächliche Angriffspunkt der resultierenden Kontaktkraft deutlich vom extremalen Kontaktpunkt abweichen (Abbildung 3.19).

Erhebliche Probleme können auftreten, wenn die Oberflächen nicht streng konvex geformt sind, denn dann lassen sich die Gleichungen (3.28) nicht im-

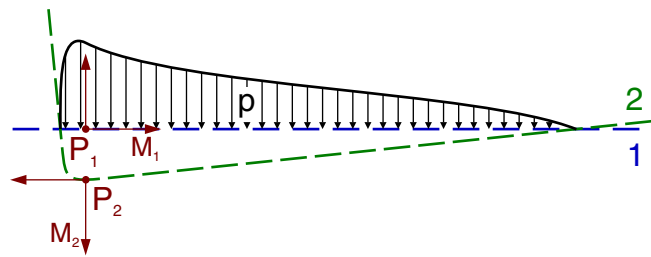


Abbildung 3.19: Bei unsymmetrischer Spannungsverteilung kann die resultierende Kraft erheblich von den extremalen Kontaktpunkten abweichen

mer eindeutig lösen [10, 61]. Abbildung 3.20 veranschaulicht solche Situationen als *Mehrpunktkontakt* bzw. *Kontaktpunktsprung* und *Schmiegung*. Selbst bei konvexen Oberflächen sind singuläre Zustände nicht auszuschließen, wenn relativ große Überschneidungen vorkommen [11]. Treten solche

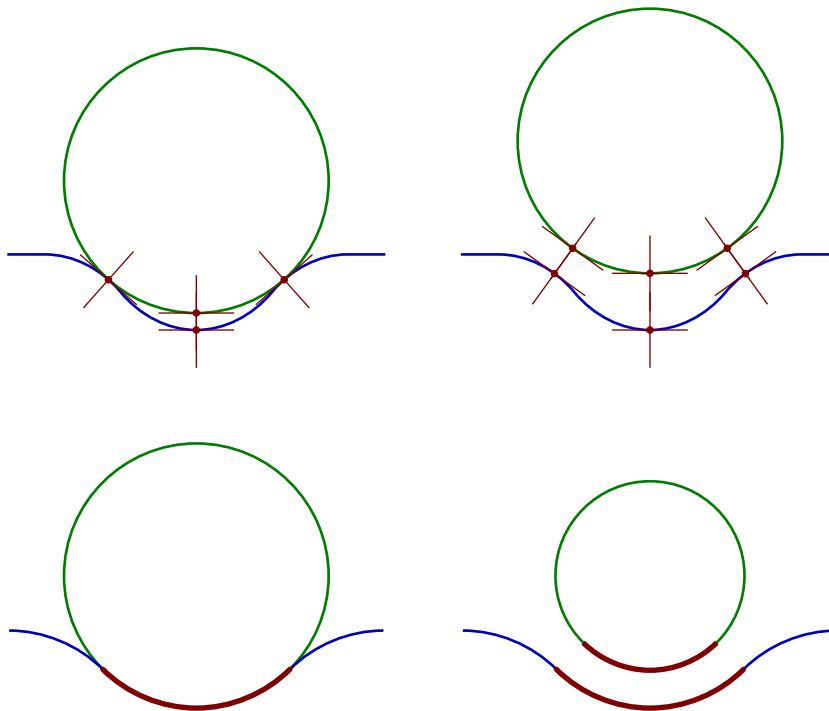


Abbildung 3.20: Bei konkaven Oberflächen sind extremale Kontaktpunkte i. Allg. nicht eindeutig definiert: Mehrpunktkontakt (oben) und Schmiegung (unten) können sowohl mit (links) als auch ohne (rechts) Kontakt auftreten

Konfigurationen während der Simulation auf, kann die Lage der extremalen Kontaktpunkte nicht berechnet werden. Die Zeitintegration bricht dann scheinbar grundlos ab, selbst wenn sämtliche Kontakte des MKS separiert sind.

Außerdem erfordert die numerische Lösung von (3.28) mit dem Newtonverfahren stetig differenzierbare Flächen bis zur zweiten Ordnung, was die Modellierung komplexer Oberflächen mit großen Krümmungsgradienten erheblich erschwert — insbesondere, weil selbst kleinste Oszillationen der Approximation lokal zu konkaven Abschnitten führen können.

3.6 Gegenüberstellung und Bewertung

In den vorangegangenen Abschnitten wurden verschiedene Kontaktmodelle vorgestellt. Tabelle 3.1 stellt sie hinsichtlich ihrer Eignung für die MKS-Simulation gegenüber. Das Bewertungskriterium *Qualität* gibt an, wie nahe ein Modell den physikalischen Phänomenen realer Kontaktvorgänge kommt. Die *Effizienz* steht für den Aufwand der numerischen Lösung und steht in engem Zusammenhang mit der *Komplexität*, die aber auch Interpretierbarkeit der Ergebnisse sowie Implementierungsaufwand und Wartbarkeit eines Programmcodes entscheidend beeinflusst. Mit der *Robustheit* wird die numerische Stabilität, aber auch die Empfindlichkeit gegenüber kritischen Parametern oder sonstigen Daten beurteilt. Und *Flexibilität* meint die Breite des Anwendungsbereichs eines Modells.

Die *Elastizitätstheorie* stellt die Grundlage aller anderen Modelle mit Ausnahme des Starrkörperkontakts dar. Analytische Methoden erlauben für wenige, sehr spezielle Probleme die Bestimmung der Verformungs- und Spannungsfelder in als linear-elastisch angenommenen, kontaktierenden Körpern in geschlossener Form. Solche Beziehungen werden in der entsprechenden Spalte aus Tabelle 3.1 bewertet. Dieser Ansatz lässt sich aufgrund der mangelnden Flexibilität jedoch nicht als allgemeines Kontaktmodell in der MKS-Simulation anwenden.

Auch die *Finite-Elemente-Methode* erscheint in dieser Hinsicht nicht sehr interessant. Zwar kann sie beinahe beliebige Kontaktprobleme in bestmöglicher Qualität darstellen; doch der dafür erforderliche immense Rechenaufwand, insbesondere bei dynamischen Simulationen von Modellen mit zahlreichen Kontakten, stellt ein Ausschlusskriterium für die Anwendung in der MKS-Welt dar. Darüber hinaus erscheint die hohe Komplexität eines hybriden MKS-/FEM-Algorithmus nachteilig.

Die beiden Oberflächenmodelle ergeben ein recht ausgewogenes Bild. Gegenüber der in MKS-Programmen üblichen eindimensionalen Modellie-

Bewertung	Elastizitätstheorie	FEM	Halbraumapproximation	Randschichtmodell	Kraftelemente	Starrkörperkontakt
Qualität	++	++	+	o	-	-
Effizienz	++	--	-	o	++	++
Komplexität	o	--	o	+	++	o
Robustheit	++	o	o	+	-	--
Flexibilität	--	++	+	+	-	-

Tabelle 3.1: Bewertung der Kontaktmodelle hinsichtlich Implementierung in MKS-Programme

rung liegt ihr wesentlicher Vorteil in ihrer „flächigen“ Wirkungsweise, durch die auch unsymmetrische Spannungsverteilungen, Mehrpunktkontakt und Schmiegun korrekt behandelt werden. Während die *Halbraumapproximation* ausgesprochen hohe Modellqualität bei eindeutiger Parametrierung bietet, sprechen Effizienz, Komplexität und Robustheit für das *Randschichtmodell*.

Für bestimmte Anwendungen stellen eindimensionale Modelle die optimale Kontaktmodellierung dar. Die Verwendung von *Kraftelementen* ist im MKS-Bereich Standard und sehr weit entwickelt. Der *Starrkörperkontakt* weist ein vergleichsweise schlechtes Verhältnis zwischen Aufwand und Qualität auf: Die Annahme starrer Körper und Stoßhypothesen sind oft unzulässige Vereinfachungen und gehen mit erheblichen Schwierigkeiten durch ein strukturvariantes System und die Herstellung konsistenter Kontaktzustände einher. Beide Methoden werfen außerdem mitunter erhebliche Probleme bei der Ermittlung der Kontaktpunkte und -normalenrichtung auf.

Kapitel 4

Polygonflächen–Kontaktmodell

In diesem Kapitel wird eine neuartige Methode zur Beschreibung von Kontaktvorgängen in Mehrkörpersystemen vorgestellt. Als Motivation dienen die zuvor erläuterten typischen Schwächen bekannter Verfahren:

Spezialisierung Aus der Literatur und aus Implementierungen sind zahlreiche Kontaktmodelle und -algorithmen bekannt, die jeweils für eine spezielle Anwendung maßgeschneidert sind. Das neue Verfahren soll hingegen möglichst universell einsetzbar sein, sodass es ein möglichst breites Spektrum von Kontaktproblemen lösen kann und sich als Modul in beliebigen MKS–Programmen verwenden lässt.

Effizienz Ein entscheidender Vorteil der MKS–Simulation ist ihr vergleichsweise niedriger Rechenaufwand. Hingegen erfordert die numerische Auswertung von Kontakten etwa mittels der Finite–Elemente–Methode um Größenordnungen höhere Rechenzeiten. Das neue Verfahren soll einen für die Mehrkörperdynamik typischen Aufwand haben.

Robustheit Mangelnde Stabilität der Berechnungsalgorithmen ist ein häufiges Problem bei der numerischen Lösung von Kontaktproblemen. Das neue Verfahren soll sich in dieser Hinsicht robust verhalten, sodass bei der Simulation Effizienzprobleme und Programmabbrüche vermieden werden.

Modellgüte Insbesondere in kommerziellen MKS–Programmen werden meist lediglich die stark vereinfachenden eindimensionalen Kontaktmodelle bereitgestellt, die die physikalischen Vorgänge häufig nur unzureichend abbilden können. Das neue Verfahren soll die Kontaktmechanik in besserer Qualität darstellen.

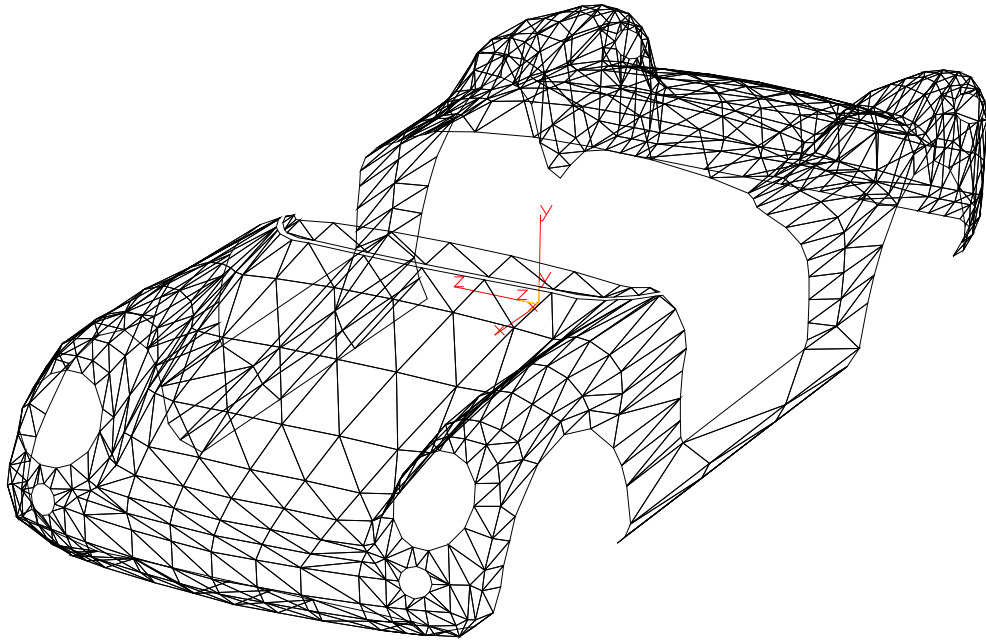


Abbildung 4.1: Pkw-Karosserie als aus Dreiecken zusammengesetzte Polygonfläche

In der Praxis wird häufig eine einfache und flexible Modellierung der für die Kontaktanalyse maßgeblichen Körperoberflächen gefordert. Das neue Polygonflächen-Kontaktmodell (*Polygonal Contact Model, PCM*)¹ basiert daher auf Dreiecksflächen (Abbildung 4.1), die die einfachste Darstellung von Freiformflächen repräsentieren.

Praktisch jede Software zur Verarbeitung räumlicher Flächen, insbesondere auch CAD-Programme, verfügt über eine Option zum Export als Polygonfläche. Darüber hinaus gibt es zahlreiche umfangreiche Bibliotheken von 3D-Modellen, die als Polygonflächen oder in einem leicht konvertierbaren Format vorliegen. Da zur Darstellung dreidimensionaler Grafik auch rechnerintern mit Dreiecksflächen gearbeitet wird, existieren außerdem zahllose Veröffentlichungen und Algorithmen für diesen Flächentyp. Flächen, die sich aus ebenen Polygonen mit mehr als drei Ecken zusammen setzen, kann man sehr einfach durch Einfügen zusätzlicher Kanten in Dreiecksflächen umwandeln.

Im Idealfall ermöglicht es das PCM also, ein CAD-Modell eines Bauteils oder eine 3D-Fläche aus einer Bibliothek im Internet als Körperoberfläche

¹Zur Vereinfachung steht die Bezeichnung PCM sowohl für die physikalische Modellierung als auch die Algorithmen und die Implementierung des im Rahmen dieser Arbeit entwickelten Verfahrens

in der MKS-Simulation zu verwenden. Die im Abschnitt 4.2 beschriebenen Methoden zur Konstruktion der Schnittvolumina erfordern allerdings orientierbare, lückenlose Flächen, wie man sie in der Praxis nicht immer vorfindet. Auch sind nicht vollständig geschlossene Flächen (wie z. B. in Abbildung 4.1) prinzipiell problematisch, wenn Kontakt an den Randlinien auftritt. Zur Beseitigung solcher Mängel sowie für viele weitere Manipulationen wie etwa Kombination, Umformung oder Vergrößerung stehen leistungsfähige Programme (z. B. [62]) zur Verfügung.

Das PCM soll über die bei MKS-Programmen häufig anzutreffenden ein-dimensionalen Kontaktmodelle hinaus gehen, um eine höhere Modellgüte zu bieten. Aus der bewertenden Gegenüberstellung der Kontaktmodelle in Abschnitt 3.6 ging das Randschichtmodell als leistungsfähigere Alternative mit angemessenem Aufwand hervor. Deshalb wurde es als kontaktmechanische Grundlage für das PCM gewählt.

Die Programmiersprache FORTRAN bewährt sich aufgrund ihrer besonders einfachen, wenig fehlerträchtigen Implementierung und der hohen Effizienz des resultierenden Maschinencodes seit Jahrzehnten in numerisch dominierten Anwendungen. Dennoch fiel für das PCM die Wahl auf die ebenfalls prozedurale Sprache C, weil sie sich durch besonders flexiblen Umgang mit komplexen, dynamisch allozierbaren Datenstrukturen auszeichnet. Die Programmierschnittstellen wurden dabei so formuliert, dass das aufrufende MKS-Programm sowohl in FORTRAN als auch C oder C++ implementiert sein kann (s. Anhang A).

Programmablauf

Abbildung 4.2 stellt die wesentlichen algorithmischen Schritte der Kontaktanalyse mit dem PCM dar. Als Eingang wird der Zustandsvektor \mathbf{y} des Mehrkörpersystems nach (2.7) verwendet. Aus ihm kann einfach die relative Lage und Orientierung der betrachteten Kontaktpaarung bestimmt werden [33]. Mit den Körperreferenzsystemen R_1 und R_2 und den zugehörigen Flächenreferenzsystemen E und F mit jeweils konstanter Einbaulage und -orientierung (Abbildung 4.3) ergibt sich:

$$\mathbf{r}_{EF}(\mathbf{y}) = \mathbf{r}_{R_2}(\mathbf{y}) + \mathbf{r}_{R_2F} - \mathbf{r}_{R_1}(\mathbf{y}) - \mathbf{r}_{R_1E} \quad (4.1)$$

$$A_{EF}(\mathbf{y}) = A_{R_1E}^T \cdot A_{IR_1}^T(\mathbf{y}) \cdot A_{IR_2}(\mathbf{y}) \cdot A_{R_2F} \quad (4.2)$$

Die erste zu lösende Teilaufgabe ist eine Kollisionserkennung. Sie stellt fest, ob sich die gegebenen Polygonflächen im aktuellen Systemzustand schneiden. Falls dies nicht der Fall ist, gibt es keine Kontaktkraft, und die Analyse kann beendet werden.

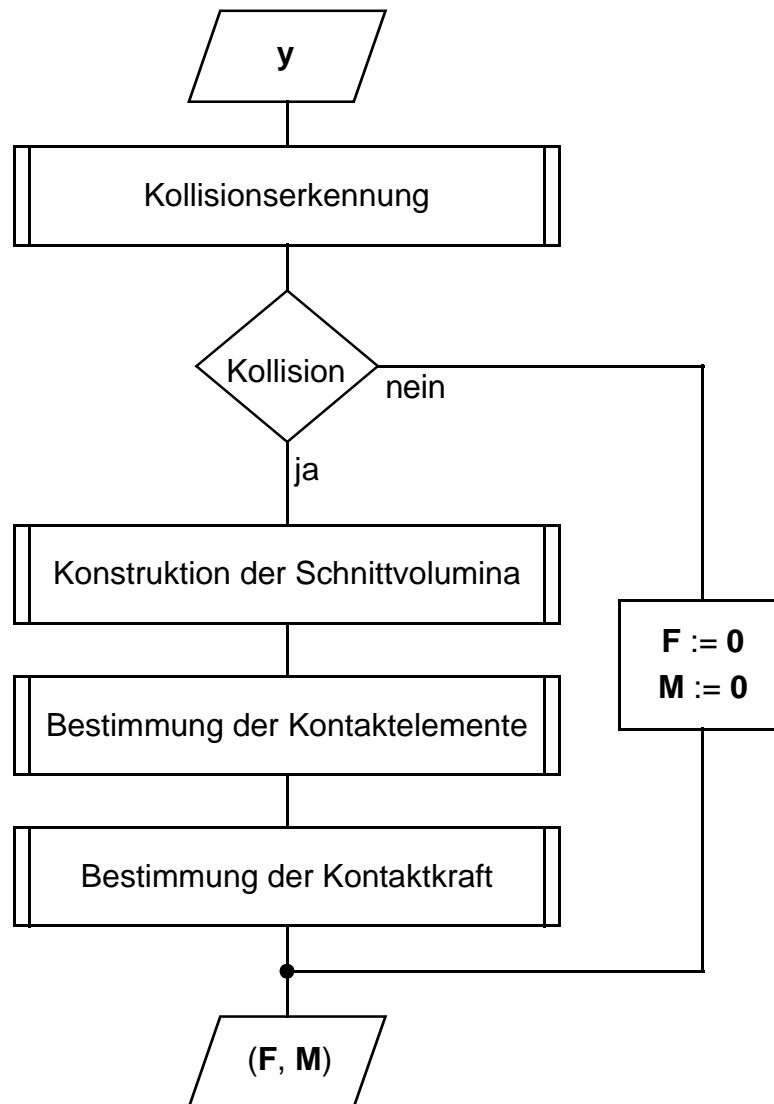


Abbildung 4.2: Programmablaufplan des PCM: An der Schnittstelle zum MKS-Programm treten die Relativkinematik der Körper (ableitbar aus dem Systemzustand \mathbf{y}) als Eingangsgröße, und der resultierende Kontaktkraftwinder (\mathbf{F}, \mathbf{M}) als Rückgabewert auf

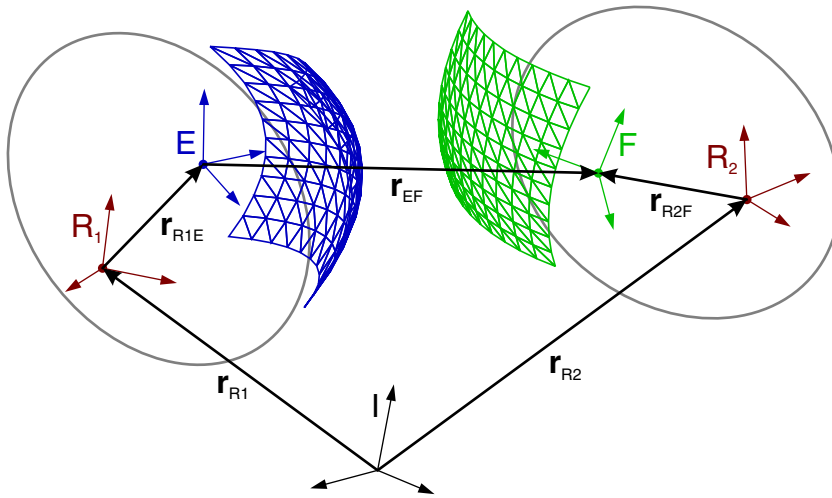


Abbildung 4.3: Relativkinematik der Flächenreferenzsysteme E und F zweier Körper 1 und 2

Ansonsten ist im nächsten Schritt das durch die sich durchdringenden Flächen gebildete Schnittvolumen zu bestimmen. Dazu sind zunächst die durch die Kollisionserkennung ermittelten Schnittlinien zwischen Dreiecken der beiden Oberflächen zu einem Schnittpolygonzug — oder bei mehrfachem Kontakt auch mehreren Schnittpolygonzügen — mit definierter Orientierung zusammenzufügen. Dann werden jeweils die „aktiven“ Dreiecke ermittelt, die innerhalb des Schnittpolygonzugs liegen oder von ihm durchlaufen werden. In Anlehnung an die numerische Halbraumapproximation wird das Kontaktgebiet anschließend durch dreieckige *Kontaktelemente* diskretisiert, die durch zwei alternative Methoden näherungsweise an der Berührfläche platziert werden.

Bis zu diesem Punkt handelt es sich um eine rein geometrische Problemstellung. Die darauf folgende Berechnung der Kontaktnormalkraft setzt hingegen ein physikalisches Modell voraus. Das Gleiche gilt für die gegebenenfalls in Abhängigkeit davon zu ermittelnde Tangentialkraft. In der vorliegenden Implementierung wird das Randschichtmodell und eine regularisierte Version des Coulombschen Reibgesetzes umgesetzt. Die Summe (\mathbf{F}, \mathbf{M}) der so erhaltenen Kraftwinder wird schließlich als eingeprägte Kraft–Momenten–Paarung in das MKS eingeleitet.

In den folgenden Abschnitten wird ausführlich auf die genannten Teilaufgaben eingegangen.

4.1 Kollisionserkennung

Die Kollisionserkennung stellt fest, ob sich zwei Flächen im Raum schneiden. Für Polygonflächen wurden zahlreiche leistungsfähige Verfahren entwickelt, die teilweise auch in Form von Software zur Verfügung stehen. Hervorzuheben sind die Arbeiten von Hubbard [63], Klosowski [64, 65] und Zachmann [66, 67, 68], sowie die Algorithmen SOLID [69], RAPID [70] und QuickCD [71].

4.1.1 Anforderungen

Das Thema Kollisionserkennung wird typischerweise für *Virtual-Reality*-Anwendungen aufgegriffen. Beispielsweise spielt es eine zentrale Rolle in Computerspielen, in der computergestützten Konstruktion (CAD) und bei der Bahnplanung für Roboter. Nur sehr wenige Arbeiten befassen sich damit hingegen unter dem Aspekt der MKS-Simulation [22, 72]. Deshalb ist sorgfältig zu klären, welchen Anforderungen die Kollisionserkennung genügen muss.

Genauigkeit Wie in Abbildung 4.2 ersichtlich ist, wird die Kollisionserkennung im Kontaktmodell als Teil eines Kraftelements eingesetzt. Es ist bekannt, dass hierbei mit hoher Genauigkeit gerechnet werden muss. Selbst nur vereinzelt auftretende Variablen mit einfacher Genauigkeit können nämlich bei Verwendung eines BDF-Integrators zu fatalen Konvergenzproblemen im Korrektorschritt führen (s. Abschnitt 2.3). Deshalb dürfen in der Kollisionserkennung keinesfalls prinzipbedingte Ungenauigkeiten auftreten, und es kommen nur Verfahren hoher Genauigkeit in Betracht.

Effizienz *Virtual-Reality*-Anwendungen stellen wegen ihres Echtzeitcharakters häufig extrem hohe Anforderungen an die Effizienz der Kollisionserkennung [68]. Auch in der MKS-Simulation ist eine schnelle Analyse wünschenswert, aber Genauigkeit und Allgemeinheit haben höhere Bedeutung. Hier sind relativ (zeit)aufwändige, einmalige Vorabrechnungen (*Pre-Processing*) akzeptabel, während die im Laufe der Zeitintegration sehr häufig durchzuführende Analyse möglichst wenig Rechenzeit in Anspruch nehmen sollte.

Robustheit Die in der Praxis verwendeten Polygonflächen sind häufig fehlerhaft [68]. Beispielsweise treten Lücken zwischen benachbarten Polygonen (*Cracks*) auf, oder deren Orientierung, welche durch den Umlaufsinn der Eckpunkte festgelegt wird, ist nicht konsistent. Bei Polygonen mit mehr als drei Ecken kann es auch vorkommen, dass die Eckpunkte

nicht in einer Ebene liegen. Da diese Probleme nicht in direktem Zusammenhang mit dem Kontaktmodell stehen, wird in dieser Arbeit nicht darauf eingegangen. Vielmehr wird von fehlerfreien Polygonflächen ausgegangen, die ausschließlich aus Dreiecken gebildet werden.

Allgemeinheit Bei vielen Algorithmen zur Kollisionserkennung werden Einschränkungen der Allgemeinheit der Flächen gemacht. Häufig können etwa nur konvexe Flächen verarbeitet werden [69]. Die MKS-Simulation stellt in dieser Hinsicht sehr hohe Anforderungen: Abgesehen von Selbstdurchdringungen muss die Form der Polygonflächen beliebig sein, und für den Einsatz auf elastischen Körpern ist ein Verfahren wünschenswert, welches auch mit verformbaren Flächen umgehen kann.

Neben den genannten Hauptmerkmalen unterscheiden sich die verschiedenen Implementierungen auch im Speicherplatzbedarf und im Abstraktionsgrad. Selbstverständlich sind geringer Ressourcenverbrauch und hohe Anschaulichkeit positiv zu bewerten.

4.1.2 Elementare Analyse

Ein großer Teil der Algorithmen zur Kollisionserkennung basiert auf einer Prüfung, ob sich zwei gegebene Polygone im Raum schneiden. Diese elementare Analyse kann auf verschiedene Weise realisiert werden. Für Dreiecke erweist sich die Methode von Möller [73] als besonders effizient [68].

Gegeben seien die Ortsvektoren der Eckpunkte $\mathbf{e}_{1..3}$, $\mathbf{f}_{1..3}$ zweier Dreiecke e und f . Zunächst bestimmt man die Ebene, in der das Dreieck e liegt:

$$\mathbf{n}_e \cdot \mathbf{r} + m_e = 0 \quad (4.3)$$

mit

$$\begin{aligned} \mathbf{n}_e &= (\mathbf{e}_2 - \mathbf{e}_1) \times (\mathbf{e}_3 - \mathbf{e}_1) \\ m_e &= -\mathbf{n}_e \cdot \mathbf{e}_1 \end{aligned}$$

Aus (4.3) lassen sich einfach die Abstände $a_{f1..3}$ der Eckpunkte $\mathbf{f}_{1..3}$ zur Ebene von e berechnen:

$$a_{f1..3} = \mathbf{n}_e \cdot \mathbf{f}_{1..3} + m_e$$

Falls alle drei Abstände von null verschieden sind und das gleiche Vorzeichen besitzen, liegt das Dreieck e außerhalb der Ebene von f . Folglich schneiden sich die Dreiecke nicht, und die Analyse wird mit diesem Ergebnis beendet.

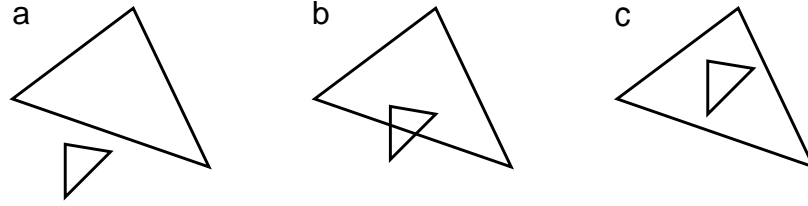


Abbildung 4.4: Schnitt zweier Dreiecke in einer gemeinsamen Ebene

Sonst wird das gleiche Verfahren mit vertauschten Dreiecken angewendet und wiederum erkannt, dass keine Kollision vorliegt, falls f nicht in der Ebene von e liegt. Bis zu diesem Punkt werden bereits alle Dreieckspaarungen aussortiert, für die es keine Gerade in Richtung $\mathbf{n}_e \times \mathbf{n}_f$ gibt, welche durch beide Dreiecke verläuft.

Anderenfalls wird zunächst der Sonderfall $a_{1..3} \approx 0$ behandelt, d. h. die Dreiecke liegen (nahezu) in der gleichen Ebene. Hierzu betrachtet man sie in dieser Ebene und erhält ein zweidimensionales Problem (Abbildung 4.4). Zuerst wird paarweise geprüft, ob sich die Seiten von e mit denen von f schneiden (Fall b). Ist dies nicht der Fall, kann trotzdem eine Kollision vorliegen, falls eines der beiden Dreiecke vollständig innerhalb des anderen liegt (Fall c). Deshalb wird weiter getestet, ob eine Ecke von e innerhalb f , oder umgekehrt, liegt. Ansonsten wird keine Kollision erkannt (Fall a).

Die unscharfe Bedingung für den koplanaren Fall macht bereits deutlich, dass an dieser Stelle numerische Schwierigkeiten auftreten können. Im Programm wird durch $|a_{1..3}| < \varepsilon$ mit einer kleinen Schranke verglichen. Tatsächlich wurden dadurch in den ersten Versuchsrechnungen doppelte oder zu Punkten entartete Schnittlinien ermittelt. Schließlich konnte mit $\varepsilon = 10^{-32}$ ein Wert gefunden werden, mit dem bei keinem der zahlreichen Tests mehr Probleme auftraten.

Sehr viel wahrscheinlicher ist, dass die Dreiecke in verschiedenen Ebenen liegen (Abbildung 4.5). Dann wird als nächster Schritt die Schnittgerade

$$\mathbf{r} = (\mathbf{n}_e \times \mathbf{n}_f) \cdot s + \mathbf{r}_0 \quad , \quad s \in \mathbb{R} \quad (4.4)$$

dieser Ebenen betrachtet. Eine Kollision tritt genau dann auf, wenn sich die Intervalle $[s_{e1}, s_{e2}]$ und $[s_{f1}, s_{f2}]$ der Geradenabschnitte innerhalb der Dreiecke überlappen. Zur Berechnung der Intervallgrenzen kann $\mathbf{r}_0 = 0$ gesetzt werden, weil die absolute Lage keinen Einfluss auf die weitere Analyse hat. Mit

$$\begin{aligned} p_{e1..3} &= (\mathbf{n}_e \times \mathbf{n}_f) \cdot \mathbf{e}_{1..3} \\ p_{f1..3} &= (\mathbf{n}_e \times \mathbf{n}_f) \cdot \mathbf{f}_{1..3} \end{aligned}$$

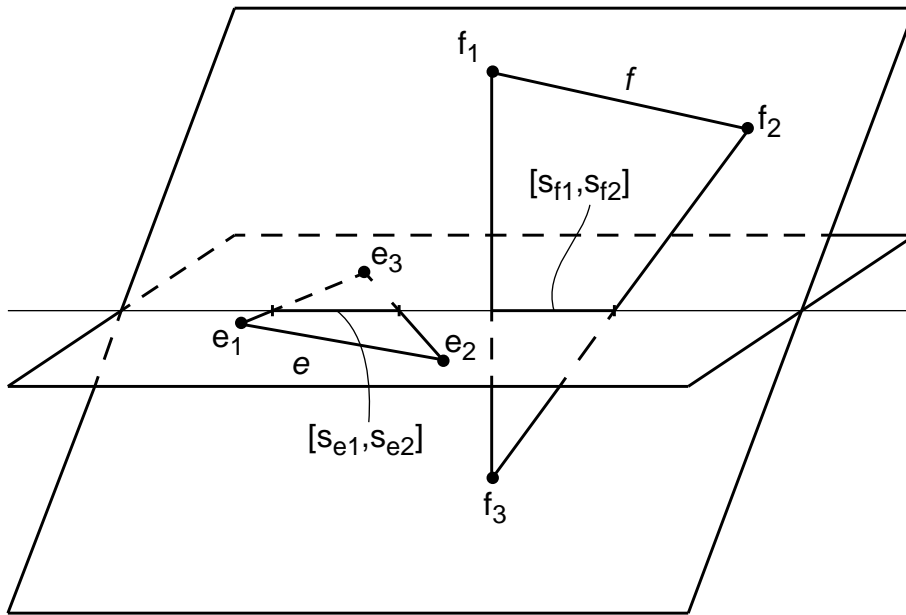


Abbildung 4.5: Schnitt zweier Dreiecke im Raum

bestimmt man die Projektion der Eckpunkte $e_{1..3}$ und $f_{1..3}$ auf die Schnittgerade (entspricht dem Abstand von r_0 zum projizierten Eckpunkt). Liegen nun (ohne Beschränkung der Allgemeinheit) e_3 auf der einen und e_1, e_2 auf der anderen Seite der Ebene von f ($a_{e3} \cdot a_{e1} \leq 0$ und $a_{e3} \cdot a_{e2} \leq 0$), können die Intervallgrenzen für e berechnet werden durch [73]:

$$s_{e1} = p_{e1} + (p_{e3} - p_{e1}) \cdot \frac{a_{e1}}{a_{e1} - a_{e3}} \quad (4.5)$$

$$s_{e2} = p_{e2} + (p_{e3} - p_{e2}) \cdot \frac{a_{e2}}{a_{e2} - a_{e3}} \quad (4.6)$$

Wendet man diese Gleichungen analog auf das Dreieck f an, kann durch einfachen Vergleich der Intervalle ermittelt werden, ob eine Kollision vorliegt. Bei Bedarf ist es dann durch Einsetzen der Grenzen der Intervallüberschneidung in die Geradengleichung (4.4) einfach möglich, die Endpunkte der Schnittlinie zu bestimmen.

Je nachdem, ob bzw. in welchem Schritt des Algorithmus die Kollision festgestellt wird, benötigt die im PCM verwendete Implementierung 76 bis 181 Gleitkommaoperationen (*FLOPs*) inklusive der erforderlichen Koordinatentransformationen.

4.1.3 Hüllvolumen–Hierarchie

Die elementare Analyse ist prinzipiell geeignet, um eine Kollision zwischen zwei Dreiecksflächen E und F zu erkennen. Dazu ist sie für alle möglichen Polygon–Kombinationen anzuwenden. Sobald ein Schnitt gefunden wird, sind Existenz und Ort der Kollision bekannt. Doch bei Flächen mit vielen Polygonen und darüber hinaus möglicherweise zahlreichen zu testenden Flächenpaaren ermöglicht selbst das schnelle Verfahren nach Möller keine effiziente Auswertung, weil die Anzahl der Polygone quadratisch in den Rechenaufwand eingeht (*all-pairs weakness*) [63].

Aus diesem Grund wurden Methoden entwickelt, durch die ein möglichst großer Teil der Polygone von der elementaren Analyse ausgeschlossen werden kann. Besonders geeignet für das PCM erscheint das hierarchische Verfahren mit sog. *diskret orientierten Polytopen* (DOPs) nach Zachmann [67, 68], welches die zuvor angeführten Anforderungen in optimaler Weise erfüllt. Im Weiteren wird es ausführlich erläutert.

Die meisten Algorithmen zur Kollisionserkennung basieren auf dem Prinzip der Hüllvolumina (*engl. Bounding Volumes, BVs*). BVs sind einfache Geometrien, die Teile der betrachteten Polygonflächen als konvexe Hülle umschließen. In einer einmaligen Vorabrechnung wird zu jeder Fläche eine Binärbaumstruktur von BVs (Abbildungen 4.6 und 4.9) bestimmt.

Das BV an der Wurzel des Baums umschließt alle Polygone der zugehörigen Fläche. An jeder Verzweigung wird die Menge der Polygone des übergeordneten BVs sukzessive in zwei räumlich zusammenhängende Teilmengen unterteilt. Die diese beiden Flächenteile umschließenden BVs werden der nächsten Ebene des BV–Baums zugeordnet. Dabei ist zu beachten, dass sich zwei benachbarte BVs in der Regel überschneiden. In der letzten Ebene umschließen die BVs jeweils nur noch ein einzelnes Polygon.

Als Geometrie der BVs werden häufig Kugeln oder Quader benutzt. Bei den Quadern lässt sich zwischen ihrer Orientierung an den Achsen des Inertialsystems (*axis aligned bounding boxes*), des Flächensystems (*6-DOPs*) oder an der Geometrie der Fläche (*oriented bounding boxes*) unterscheiden. Die hier betrachteten 6–DOPs sind den anderen Methoden bezüglich Speicherbedarf sowie Effizienz und Robustheit des Aufbaus der BV–Hierarchie überlegen und ermöglichen dennoch eine besonders leistungsfähige Kollisionserkennung [68].

Weil relative Lage und Orientierung der DOPs zum Flächenreferenzsystem konstant sind, muss die Vorabberechnung des BV–Baums für jede Fläche nur einmal durchgeführt werden, unabhängig davon, wie oft und an welcher Position sie wirksam ist und wievielen Kollisionspaarungen sie zugeordnet wird. Um einen schnellen Zugriff zu ermöglichen, wird die BV–Hierarchie anschließend unverändert im Arbeitsspeicher gehalten.

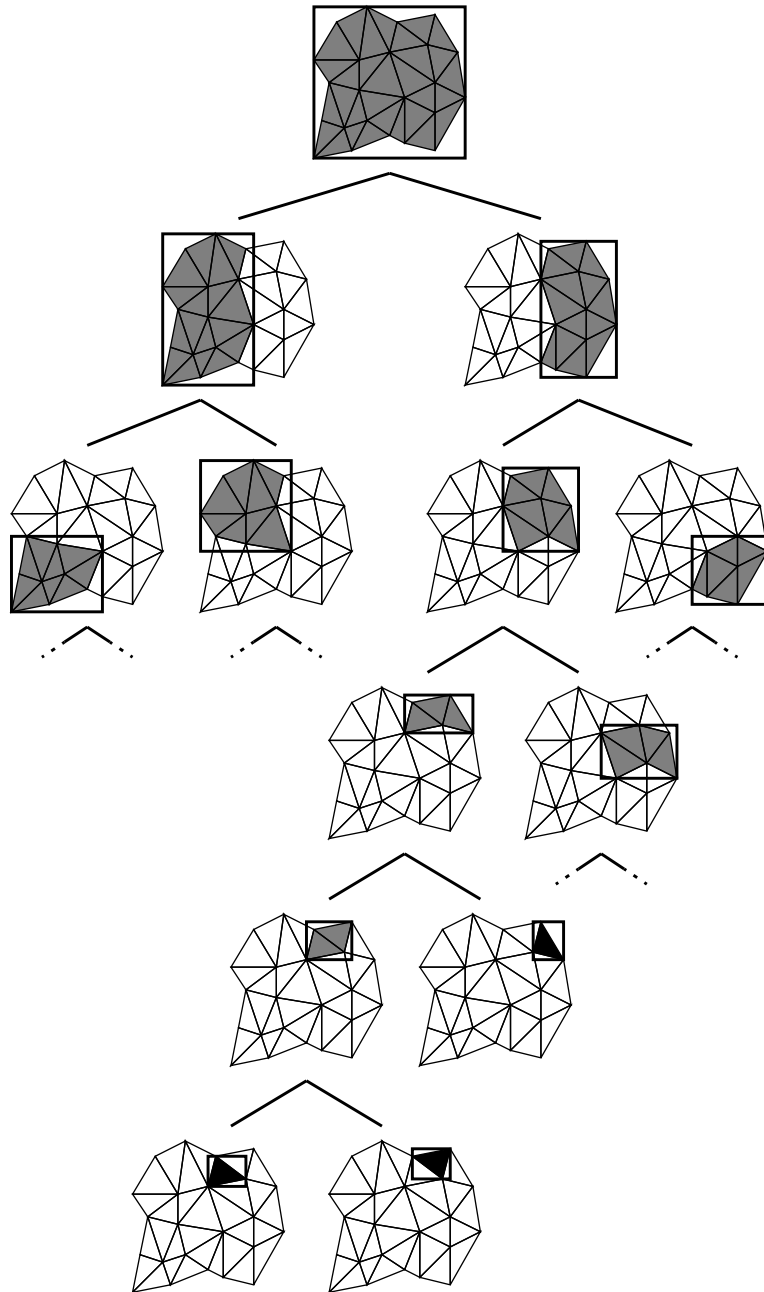


Abbildung 4.6: Ebenes Beispiel für eine Hüllvolumen-Hierarchie einer ebenen Fläche, exemplarisch für drei Blätter (schwarz) vollständig ausgeführt

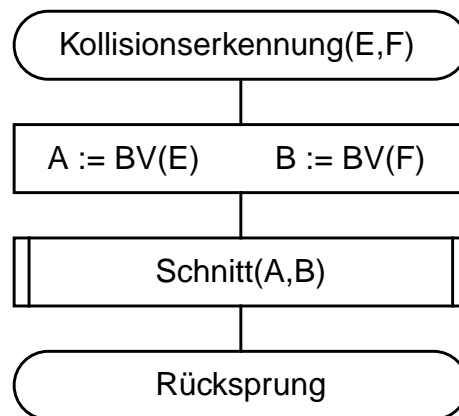


Abbildung 4.7: Hauptroutine zur Kollisionserkennung nach [67]

Als Eingangsgrößen der während der Kontaktanalyse im Zeitbereich häufig aufzurufenden Kollisionserkennung dienen Relativortsvektor und -transformation der Referenzsysteme der beiden Flächen. Zuerst werden nur die BVs betrachtet: Falls sich die BVs an der Wurzel der beiden Flächen nicht schneiden, kann eine Kollision sofort ausgeschlossen werden. Wird hingegen ein Schnitt festgestellt, kontrolliert man die BVs in der nächsten Ebene des Baums gegeneinander. Jedes BV, das kein BV der anderen Fläche schneidet, braucht nicht weiter in Betracht gezogen werden. Da sich in den meisten Fällen nur kleine Teile der Flächen schneiden, bleiben am Ende der BV-Auswertung nur wenige BVs der letzten Ebene bzw. Polygone übrig. Erst dann wird die elementare Analyse nach Abschnitt 4.1.2 für die darin liegenden Polygonpaarungen benutzt, um zu entscheiden, ob eine Kollision vorliegt.

Abbildung 4.8 zeigt den rekursiven Algorithmus, welcher für die Kollisionserkennung des PCM (Abbildung 4.7) verwendet wird. Mit $\exists a(A)$ wird die Existenz von BVs a geprüft, die einem BV A untergeordnet sind, $a_{i+1}(A)$ steht für ein weiteres BV a . Die Unterprogramme `PolyCheck(e,f)` und `BVCheck(a,b)` führen die Kollisionserkennung zweier Dreiecke e und f anhand der elementaren Analyse bzw. zweier BVs a und b mittels des im Weiteren beschriebenen Verfahrens durch. Mit `Schnitt(a,b)` ruft sich das Unterprogramm selbst auf, sodass der BV-Baum rekursiv von der Wurzel zu den Blättern abgearbeitet wird. Liefert `PolyCheck(e,f)` zwei sich schneidende Dreiecke, kann mit dieser Information abgebrochen oder bei Bedarf nach allen weiteren kollidierenden Dreiecken gesucht werden.

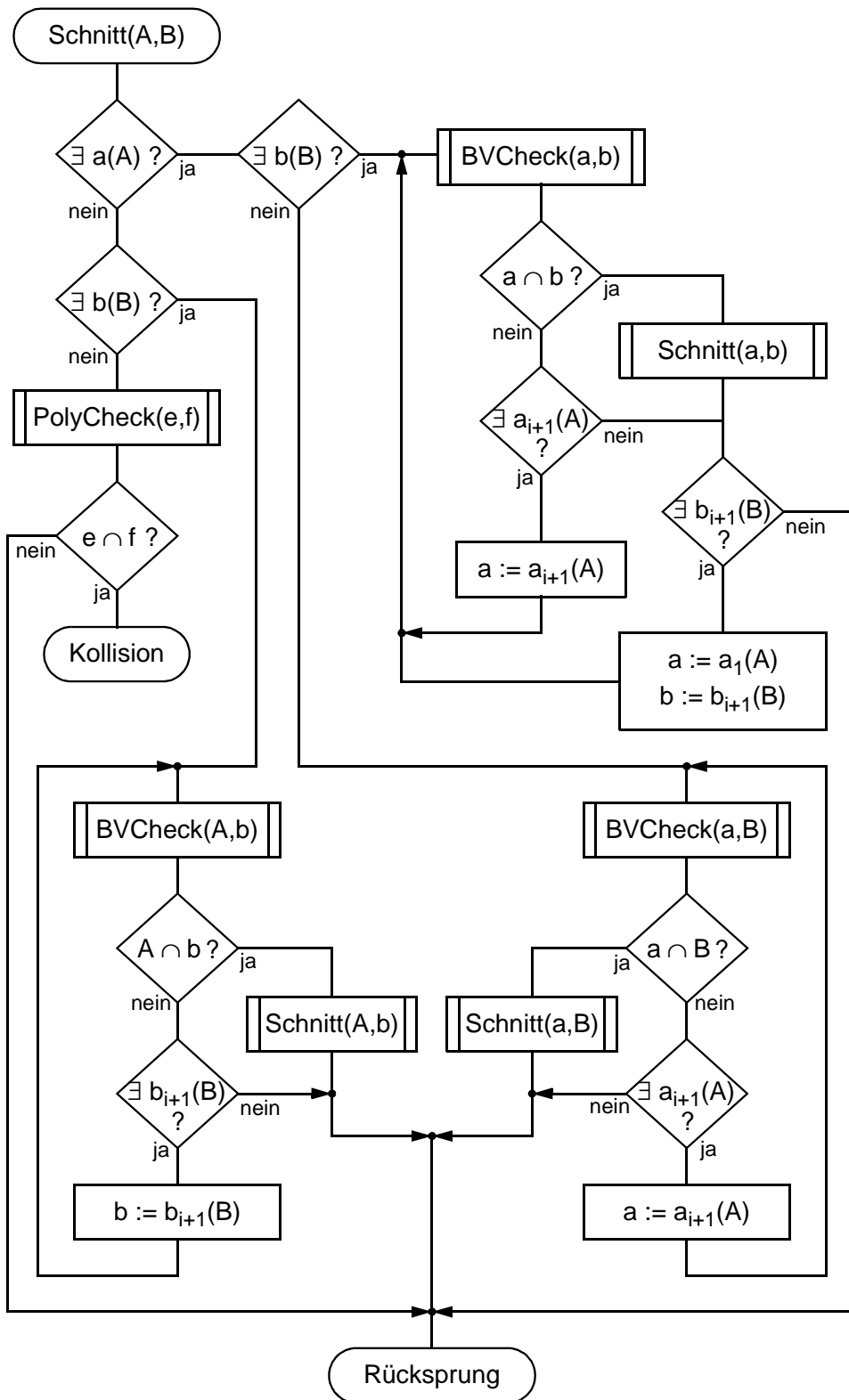


Abbildung 4.8: Rekursiver Algorithmus zur Kollisionserkennung nach [67]

Aufbau des Hüllvolumen-Baums

Die Konstruktion des Hüllvolumen-Baums nach [68] geschieht rekursiv von der Wurzel (gesamte Fläche) zu den Blättern (einzelne Polygone) hin. Alle BVs sind Quader, deren Seitenflächen normal zu den Koordinatenachsen des Flächenreferenzsystems orientiert sind. Zunächst wird bestimmt, in welcher Achsrichtung die Dreiecke des übergeordneten BV-Quaders die größte räumliche Ausdehnung besitzen. Dazu dienen durch das *Heapsort*-Verfahren nach Größe sortierte Listen der drei Koordinaten der Schwerpunkte der Dreiecke. Die umschriebenen Quader der beiden so ermittelten, am weitesten voneinander entfernten Dreiecke sind die Keimzellen der gesuchten untergeordneten BV-Quader. Nun geht man anhand der Liste abwechselnd von beiden Seiten vor und schlägt den BV-Quader des jeweiligen Polygons demjenigen BV zu, bei dem der geringere Volumenzuwachs entsteht; ist dieser auf beiden Seiten gleich, wählt man das BV mit der kleineren Polygonanzahl. Auf diese Weise entstehen zwei neue BV-Quader, die den übergeordneten BV-Quader in etwa bezüglich ihres Volumens teilen. Die Rekursion bricht ab, wenn sich nur noch ein einziges Dreieck im BV befindet.

Aus informationstechnischer Sicht ist das Endergebnis ein Binärbaum, dessen Elemente aus einer Struktur mit sechs Gleitkommazahlen und zwei Zeigern bestehen. Die reellen Variablen stellen die Lage der sechs Quaderflächen des BVs auf den Koordinatenachsen dar, und die Zeiger verweisen auf die BV-Quader der nächsten Ebene des Baums.

Abbildung 4.9 zeigt verschiedene Ebenen k des BV-Baums zur Fläche aus Abbildung 4.1, die vom PCM-Algorithmus berechnet wurden. Die schwarzen Quader stellen die Blätter des Baums dar, welche jeweils nur noch ein einzelnes Dreieck umschließen.

Kollisionserkennung zwischen Hüllvolumen

Die Analyse, ob sich zwei BVs schneiden, wird ebenso wie die elementare Analyse sehr häufig ausgeführt und ist damit zeitkritisch. Um die Effizienz zu erhöhen kann ausgenutzt werden, dass alle BV-Quader einer Fläche die gleiche Orientierung besitzen. Darüber hinaus lassen sich viele Rechenoperationen einsparen, wenn man den BV-Quader der einen Fläche durch einen umhüllenden, in den Achsrichtungen der anderen Fläche orientierten BV-Quader ersetzt. Dadurch wird zwar der Volumenüberschuß des BV erhöht, aber die Berechnung in meist überwiegendem Maße vereinfacht.

Abbildung 4.10 zeigt die Kollisionserkennung zwischen den BVs A und B zweier Flächen E und F . Der Ortsvektor \mathbf{r}_{EF} stellt die translatorische, die Transformationsmatrix A_{EF} die rotatorische Relativabweichung der Flächen-

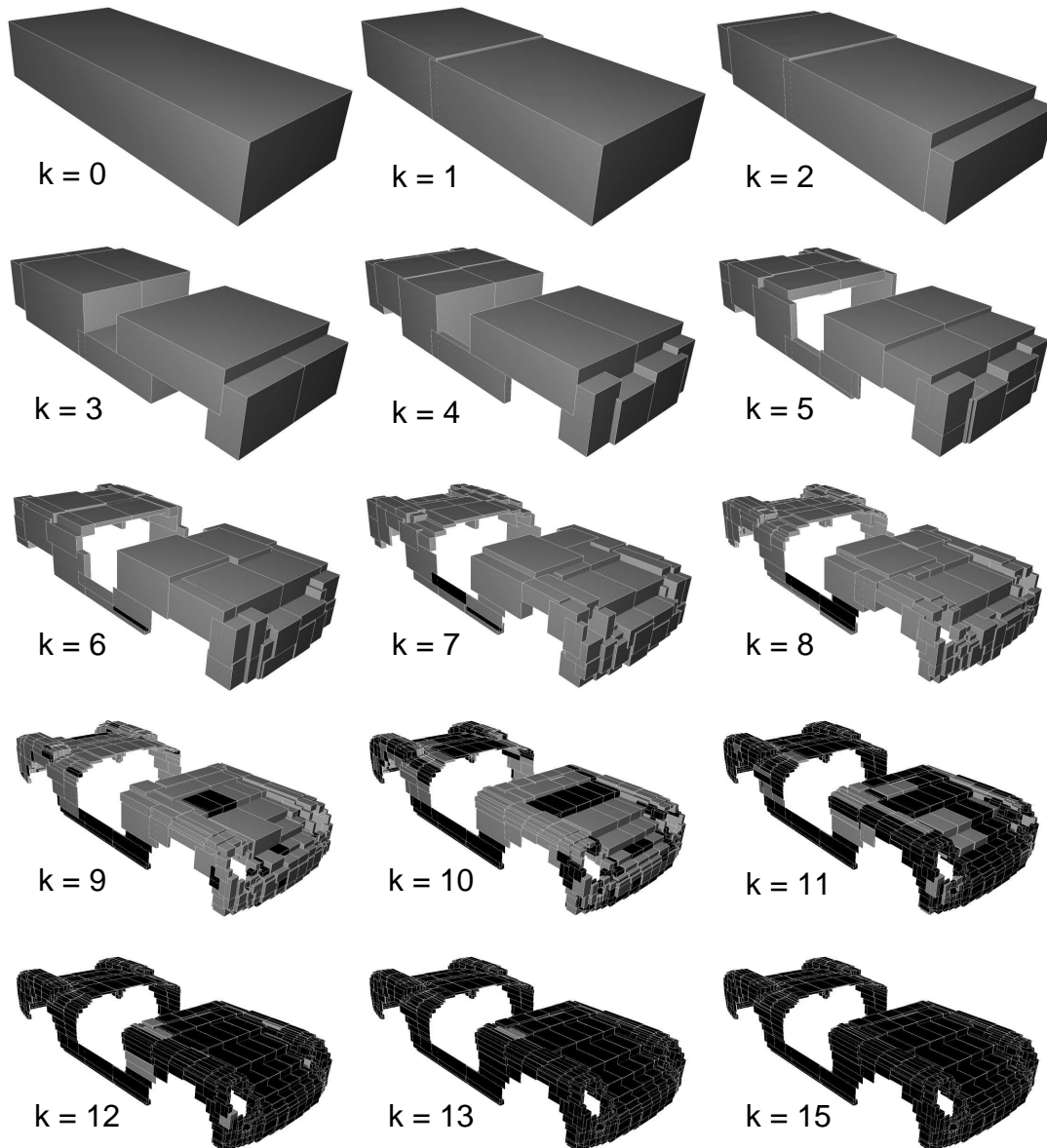


Abbildung 4.9: Hüllvolumina einer Polygonfläche für verschiedene Ebenen k der Binärbaum-Hierarchie

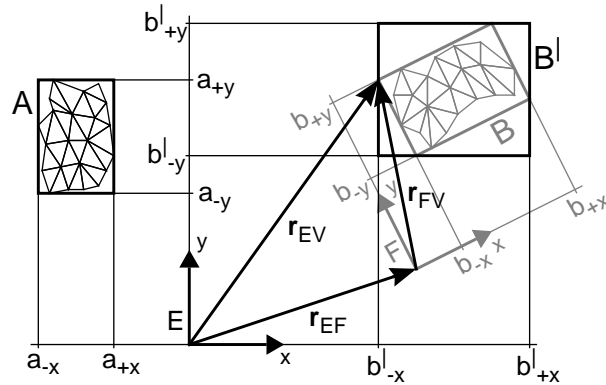


Abbildung 4.10: Ebenes Beispiel für die Kollisionserkennung zweier Hüllvolumina A und B

referenzsysteme E und F dar. b_{-x} bezeichnet beispielsweise die Lage derjenigen Seitenfläche von B , deren nach außen weisende Normale in negative x -Richtung verläuft.

Um die beiden BVs hinsichtlich Überschneidung zu analysieren, wird zuerst B durch den umschriebenen BV-Quader B' ersetzt, welcher in den Achsrichtungen von E orientiert ist. Dabei ist es grundsätzlich egal, ob die BVs von E oder F ersetzt werden. Die Ecken von B können bestimmt werden durch:

$${}_E\mathbf{r}_{EV_{1..8}} = {}_E\mathbf{r}_{EF} + A_{EF} \cdot F\mathbf{r}_{FV_{1..8}} \quad (4.7)$$

Dabei treten die Achsenabschnitte b und b' als Komponenten der Vektoren $F\mathbf{r}_{FV_{1..8}}$ bzw. ${}_E\mathbf{r}_{EV_{1..8}}$ auf.

Bei Anwendung der Gleichung (4.7) auf alle acht Eckpunkte von B würden 24 anstatt der nur sechs benötigten Koordinaten b' berechnet. Um dies zu vermeiden, kann man die Zuordnung der eingehenden Werte b zu den gesuchten b' bestimmen. Da alle BVs der Fläche F Quader mit gleicher Orientierung sind, ist diese Zuordnungstabelle stets gleich und kann für eine gegebene Anordnung A_{EF} einmalig vorab berechnet werden. Hierzu betrachtet man etwa einen Einheitswürfel ($b_{-x..z} = -1$, $b_{+x..z} = 1$) und merkt vor, für welche b Gleichung (4.7) die kleinsten und größten Werte b' liefert. So ergibt im 2D-Beispiel aus Abbildung 4.10 der Eckpunkt (b_{+x}, b_{-y}) den größten x -Wert b'_{+x} , während die y -Komponente nicht zur Darstellung von B' gebraucht wird. Im räumlichen Fall erhält man so sechs Gleichungen der Form

$$b' = r + a_1 b_x + a_2 b_y + a_3 b_z$$

und eine Kollisionsprüfung zwischen BVs benötigt nur zwölf bis 36 FLOPs.

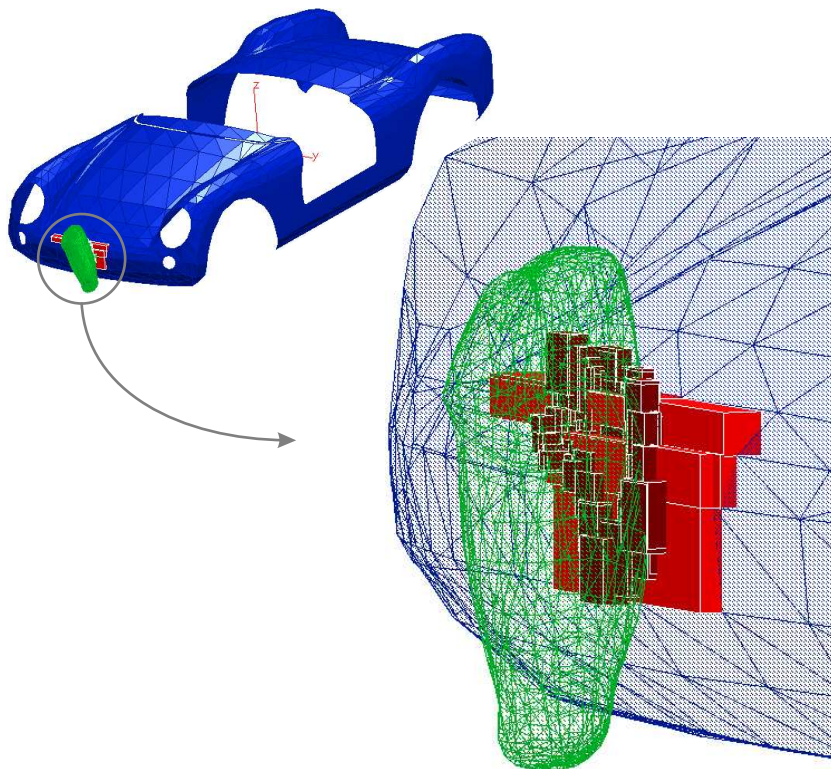


Abbildung 4.11: Hierarchische Kollisionserkennung zwischen den Oberflächen einer Autokarosserie (1218 Dreiecke) und eines Oberschenkels (1552 Dreiecke)

Effizienz

Abbildung 4.11 veranschaulicht die Funktionsweise der hierarchischen Kollisionsprüfung an einem praxisnahen Beispiel. Anstelle von 1890336 elementaren Analysen für alle möglichen Polygonpaarungen sind mit Anwendung der BV-Hierarchie nur 1537 vergleichsweise einfache Kollisionsprüfungen zwischen Hüllvolumina und 330 damit ermittelte elementare Analysen durchzuführen, die schließlich 35 Schnittlinien ergeben.

Um Funktionalität und Effizienz der Kollisionserkennung zu verifizieren, wurden Testrechnungen mit der Anordnung aus Abbildung 4.12 als Startzustand durchgeführt. Während der Simulation bewegt sich die Karosserie mit konstanter Geschwindigkeit durch die Figur hindurch. Dabei wird die Kollisionserkennung 51-mal mit unterschiedlichen Relativvektoren zwischen den Flächen aufgerufen.



Abbildung 4.12: Szenario zum Test der Kollisionserkennung; die Karosserie setzt sich aus 1218 und die Figur aus 5640 Dreiecken zusammen

CPU-Zeit	Brute Force	BV-Algorithmus
alle Schnittlinien	1:48:35,47 h	0,9 s
nur Kollisionserkennung	49:14,54 min	0,09 s

Tabelle 4.1: Rechenzeitvergleich¹ für 51 Kollisionsprüfungen zwischen den Polygonflächen aus Abbildung 4.12. Beim BV-Algorithmus wurden zur Berechnung der BV-Bäume zusätzlich einmalig 7,45 s benötigt

Der Algorithmus wurde in zwei verschiedenen Modi getestet, wobei einmal sämtliche sich schneidenden Dreiecke ermittelt wurden, während im anderen Fall abgebrochen wurde, sobald ein Schnitt gefunden war. Beide Aufgaben wurden einerseits allein mit der elementaren Analyse nach Abschnitt 4.1.2 (*Brute Force*), andererseits unter Verwendung von 6-DOP Hüllvolumen-Hierarchien gelöst. Tabelle 4.1 zeigt die ermittelten Rechenzeiten bei exakt gleichen Ergebnissen. Die dramatischen Verbesserungen um vier bis fünf Größenordnungen machen deutlich, dass eine brauchbare Implementierung des PCM erst durch den BV-Algorithmus ermöglicht wird.

¹HP Unix-Workstation, 300 MHz RISC-Prozessor, 512 MB RAM

4.2 Konstruktion der Schnittvolumina

Gemäß Abbildung 4.2 bricht der PCM-Algorithmus ab und liefert keine Kontaktkraft für den aktuellen Simulationszeitpunkt, falls die Kollisionserkennung keinen Schnitt der Oberflächen der betrachteten Kontaktpaarung feststellt. Ansonsten wird mit der Bestimmung der Schnittvolumina fortgefahren, welche die Schnittmenge der beiden unverformt angenommenen Körper darstellen. Sie werden von denjenigen „aktiven“ Dreiecken umschlossen, die ganz oder teilweise innerhalb der jeweils gegenüberliegenden Oberfläche liegen. Dazu werden als vorbereitender Schritt zunächst die Schnittpolygonzüge konstruiert.

4.2.1 Konstruktion der Schnittpolygone

Liegt Kontakt zwischen zwei Körpern vor, schneiden sich ihre unverformten Profile. Die zu beiden Oberflächen gehörigen Punkte bilden eine oder mehrere geschlossene Schnittkurven, die die Schnittvolumina von den restlichen Flächenteilen abgrenzen. Als einfaches Beispiel stelle man sich den Schnittkreis zweier Kugeln vor, deren Mittelpunkte etwas weniger als die Summe ihrer Radien voneinander entfernt liegen.

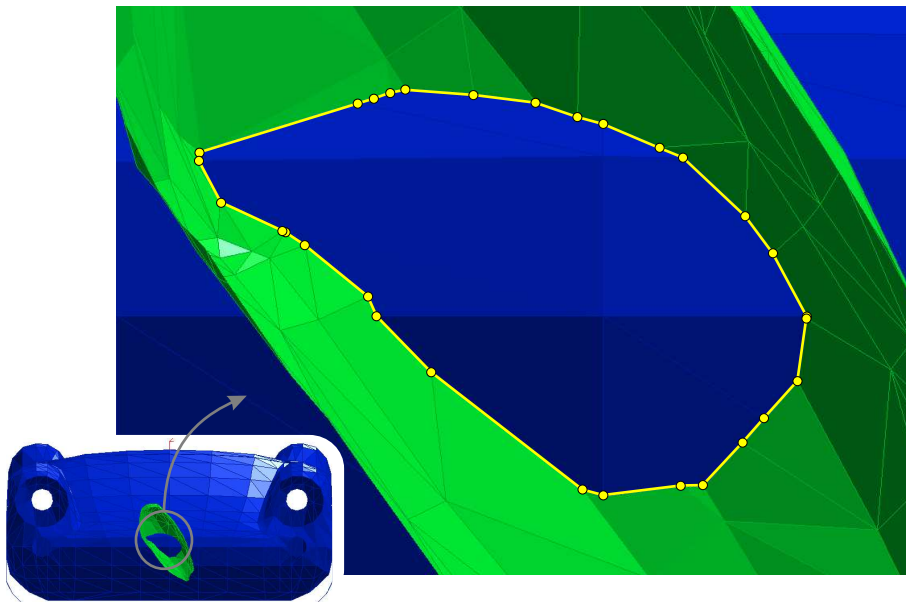


Abbildung 4.13: Schnittpolygonzug der Dreiecksflächen einer Autokarosserie und eines (zur Visualisierung geteilten) Oberschenkels

Werden die Profile durch Dreiecksflächen beschrieben, ergibt sich ein i. Allg. dreidimensionaler Schnittpolygonzug (Abbildung 4.13). Jede seiner Linien resultiert aus dem Schnitt eines Dreiecks der einen und eines weiteren der anderen Oberfläche.

Zur Kontaktanalyse wird die PCM-Kollisionserkennung stets in demjenigen Modus betrieben, der alle Schnittlinien ermittelt. Diese liegen anschließend in regelloser Reihenfolge vor. Die in einer linearen Liste (*Stack*) gespeicherten Elemente geben die kartesischen Koordinaten $\mathbf{s}_1, \mathbf{s}_2$ der beiden Endpunkte sowie die *IDs* (Kennnummern) f_E, f_F der sich schneidenden Dreiecke an.

Die Konstruktion der Schnittpolygone (Abbildung 4.14) stellt im Wesentlichen eine der räumlichen Anordnung der Schnittlinien entsprechende Ordnung der Elemente dar. Zu Beginn wird das erste Element vom Stack zum neuen Polygon p verschoben.

Bevor die weiteren Schnittlinien des Polygonzugs bestimmt werden, wird anhand des ersten Elements seine Umlaufrichtung festgelegt. Hierzu wird vorausgesetzt, dass sämtliche Dreiecke (z. B. Eckpunkte $\mathbf{v}_{1..3}$) der beiden betrachteten Flächen E und F jeweils zum Körperäußeren gerichtete Normalen

$$\mathbf{n} = \frac{(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)}{|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)|} \quad (4.8)$$

aufweisen. Damit werden die Endpunkte der Schnittlinie so indiziert, dass ihr Richtungsvektor

$$\mathbf{r}_l = \mathbf{s}_2 - \mathbf{s}_1 \quad (4.9)$$

ein Rechte-Hand-System mit den Normalenvektoren $\mathbf{n}_e, \mathbf{n}_f$ der sich schneidenden Dreiecke bildet:

$$(\mathbf{n}_e \times \mathbf{n}_f) \cdot \mathbf{r}_l > 0 \quad (4.10)$$

Anschaulich bedeutet dies, dass das Schnittvolumen stets links liegt, wenn man sich auf der Fläche E entlang des Schnittpolygonzugs bewegt (Abbildung 4.15).

Zum Aufbau des weiteren Polygonzugs wird die nächste Schnittlinie ermittelt, indem die Euklidischen Abstände zwischen den Endpunkten aller restlichen Elemente im Stack und denen des Schnittpolygons berechnet werden. Ist der kleinste gemessene Abstand kleiner als eine gegebene Schranke (bewährter Wert $d_\epsilon = 10^{-10}$), wird das entsprechende Element k vom Stack S zum Schnittpolygon p verschoben, wobei ggf. durch Vertauschung seiner Endpunkte für konsistente Laufrichtung gesorgt wird.

So wird fortgefahren, bis der Stack leer ist oder kein Element mehr gefunden wird, das nahe genug an den Enden des Schnittpolygons liegt. An-

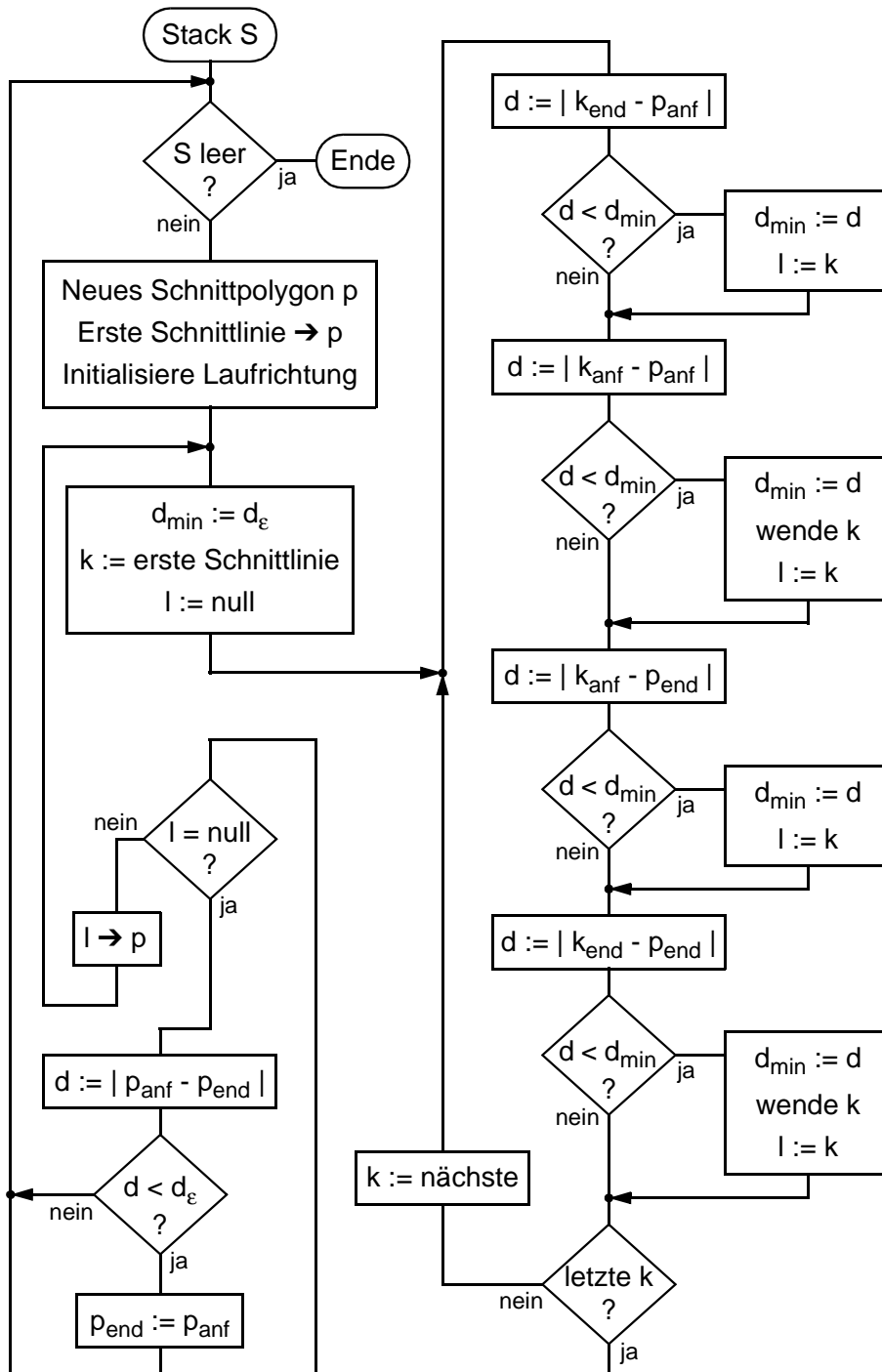


Abbildung 4.14: Algorithmus zur Konstruktion der Schnittpolygonzüge

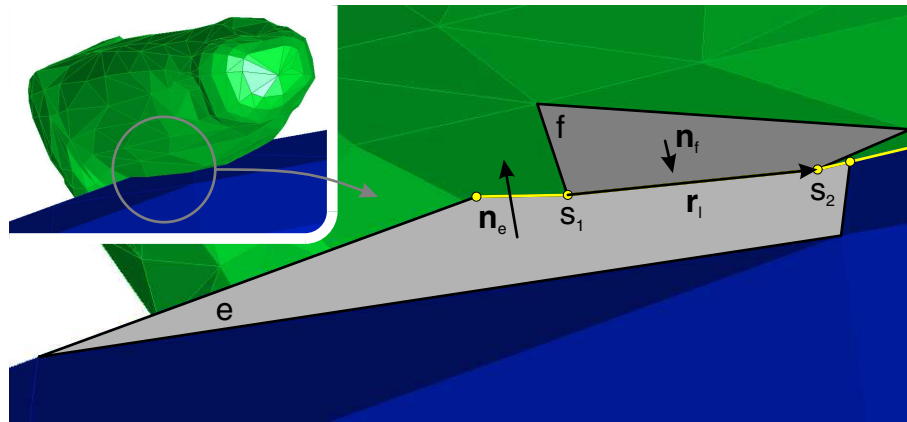


Abbildung 4.15: Bestimmung der Umlaufrichtung eines Schnittpolygonzugs

schließlich wird der Schnittpolygonzug geschlossen, falls der Abstand seiner Endpunkte kleiner als die Schranke d_ε ist; sonst wird eine Warnung ausgegeben.

Offene Schnittpolygonzüge können prinzipiell dann auftreten, wenn wenigstens eine der beiden Flächen die Körperoberfläche nur teilweise beschreibt und ihr Rand in das Schnittgebiet gerät. Diese physikalisch undefinierte Situation führt im Weiteren zu deutlichen Effizienzeinbußen (s. Abschnitt 4.2.2) und verfälschten bzw. undefinierten Kontaktkräften (s. Abschnitt 4.3).

Befinden sich nach der Konstruktion des ersten Schnittpolygonzugs weitere Elemente im Stack, werden daraus in gleicher Weise weitere Polygone aufgebaut. Mehrere Schnittpolygonzüge können auftreten, wenn sich die beiden betrachteten Körper in mehreren Gebieten berühren (Mehrfachkontakt) oder/und wenn sich infolge komplex geformter Oberflächen mehrfach berandete Kontaktgebiete ausbilden (Abbildung 4.16).

Der beschriebene Algorithmus zur Konstruktion der Schnittpolygone legt den Schwerpunkt auf Robustheit, nicht auf Effizienz. Erste Testversionen, in denen die IDs der Endpunkte benachbarter Schnittlinien ausgenutzt wurden, arbeiteten zwar etwas schneller, aber nicht immer fehlerfrei. Da auch die aufwändigere Variante nur einen geringen Anteil an der Gesamtausführungszeit einer PCM-Kontaktanalyse hat, wurde sie bevorzugt. In [72] wird ein anderer Algorithmus zur Konstruktion von Schnittpolygonzügen vorgestellt.

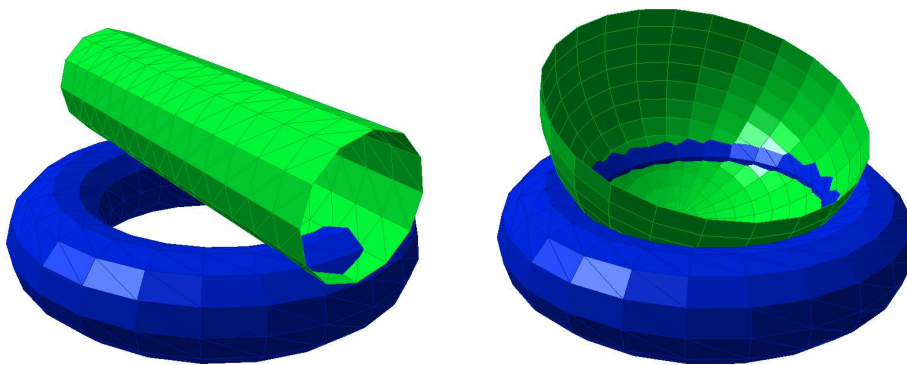


Abbildung 4.16: Beispiele für Mehrfachkontakt (links) und mehrfach (hier zweifach) berandetes Kontaktgebiet

4.2.2 Bestimmung der aktiven Flächenelemente

Unter aktiven Flächenelementen werden hier diejenigen Polygone sowie deren Eckpunkte verstanden, die ganz oder teilweise unter der unverformten Oberfläche des jeweiligen anderen Körpers liegen. Eine wichtige Grundidee des PCM ist es, bei der Kontaktanalyse nur diese i. d. R. relativ kleinen Teilflächen zu betrachten.

Bei einfachen Berührungen, die sich etwa zwischen streng konvexen Körpern ergeben, gibt es genau ein Kontaktgebiet, das von einem einzigen Schnittpolygonzug eingegrenzt wird. Auf den beiden Polygonflächen bilden sich jeweils einfach zusammenhängende Teilflächen aktiver Polygone.

Zieht man beliebig geformte Körper in Betracht, können sich auch komplexere Topologien der Berührgebiete ergeben. Die dabei möglichen Phänomene stellt Abbildung 4.16 anhand einfacher Beispiele dar: Erstens kann *Mehrfachkontakt*, d. h. gleichzeitig mehrere Berührungen der einfachen Art, auftreten. Zweitens können *mehrfach berandete Kontaktgebiete* entstehen, bei denen „Löcher“ oder „bandartige“ Geometrien zu mehreren Schnittkurven bzw. -polygonzügen führen. Es ist eine entscheidende Anforderung an ein universelles Kontaktmodell, diese Erscheinungen in beliebiger Kombination korrekt verarbeiten zu können.

Die zuvor beschriebene Konstruktion der Schnittpolygonzüge liefert bereits einen Teil der aktiven Elemente, nämlich sämtliche (im Schnitt befindlichen) *Schnittpolygone*. Durch den im Folgenden dargestellten Algorithmus werden die davon umrandeten, restlichen *inneren Polygone* ermittelt und (ggf. mehreren oder/und mehrfach berandeten) *Schnitten* zugeordnet. Anschließend liegen die Kontaktgebiete mit ihren Schnittpolygonen und aktiven Elementen beider Oberflächen als hierarchische Datenstruktur vor (Abbildung 4.17).

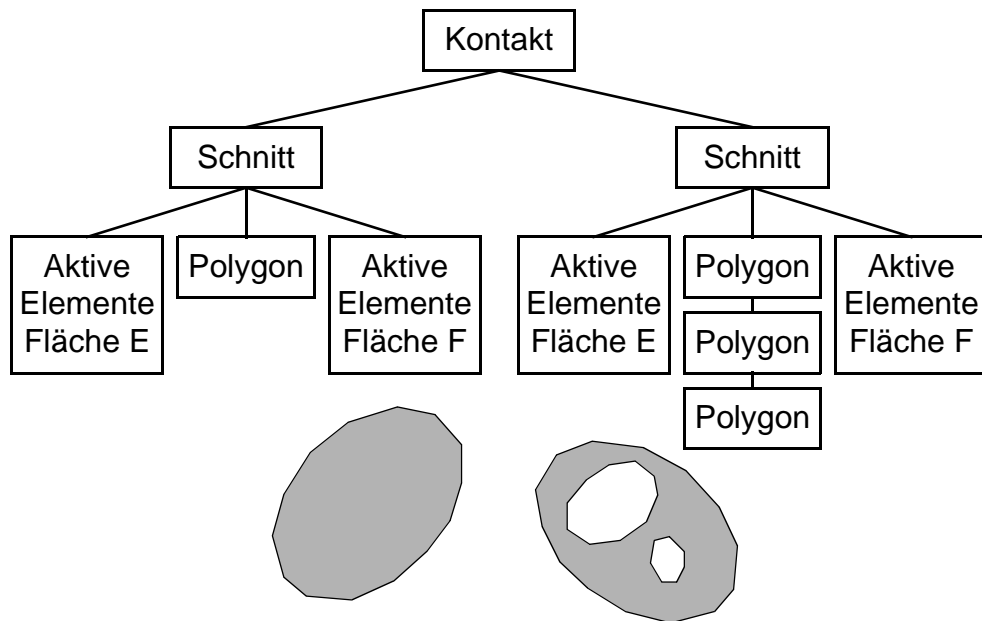


Abbildung 4.17: Datenstruktur und Topologie eines zweifachen Kontakts mit einem einfach und einem dreifach berandeten Kontaktgebiet

Das Konzept zur Bestimmung der Schnitte beruht auf einer anschaulichen Methode: Ausgehend von den Polygonen des ersten Schnittpolygonzugs werden auf den beiden Flächen jeweils sukzessive alle Nachbarpolygone bestimmt, die auf der Innenseite des Schnittgebiets liegen. Dabei wird stets kontrolliert, ob ein anderer „noch freier“ Schnittpolygonzug erreicht wird. Wenn dieser Fall eintritt, liegt ein mehrfach berandetes Kontaktgebiet vor, und der gefundene Polygonzug samt seiner aktiven Flächenelemente wird dem aktuellen Schnitt zugeordnet. Falls nach Abschluss dieser Prozedur freie Schnittpolygonzüge übrig sind, handelt es sich um einen Mehrfachkontakt; dann werden auf gleiche Weise weitere Schnitte ermittelt.

Für die Reihenfolge der Liste der Flächenelemente von Polygonflächen gibt es keine Regel („*Polygonsuppe*“). Um die inneren Elemente zu bestimmen, sind jedoch Informationen über die Nachbarschaft der Polygone der Oberflächen erforderlich. Dazu werden während des Pre-Processings einmalig für jede Fläche sog. Doubly Connected Edge Lists (DCELS) erzeugt, auf die während der Simulation zur Ermittlung der inneren Polygone zugegriffen wird.

Das Prinzip der DCELS wurde aus [66] übernommen, wo sie als besonders effiziente B-Rep (*Boundary Representation*) Datenstruktur herausgear-

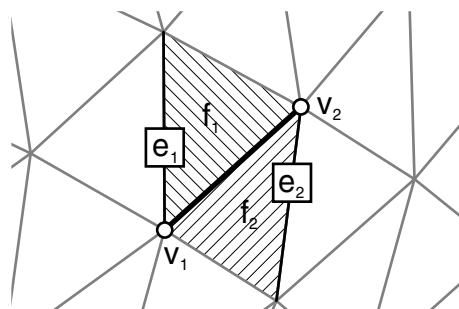


Abbildung 4.18: Verweise zu benachbarten Elementen einer Doubly Connected Edge (vom Körperäußeren betrachtet)

beitet werden. Ihre Grundelemente (DCEs) sind die Kanten der Fläche bzw. der Randlinien ihrer Polygone. Jede DCE besteht aus Verweisen auf ihre folgenden benachbarten Elemente (Abbildung 4.18): Die unten bzw. oben angrenzenden Eckpunkte v_1 und v_2 , die links bzw. rechts gelegenen Polygone f_1 und f_2 und die jeweils nächsten zugehörigen Kanten e_1 und e_2 . Bei DCEs am Rand der Fläche sind f_2 und e_2 grundsätzlich nicht belegt.

Das PCM generiert die DCEs als einfach verkettete Listen zu jedem Eckpunkt der Fläche. Auf jede DCE lässt sich somit direkt über v_1 sowie v_2 zugreifen, wobei sich ihre Orientierung bzw. die Indizierung der Eckpunkte quasi zufällig durch den Generierungsalgorithmus ergibt. In einem zusätzlichen Arbeitsschritt werden die DCEs geordnet, sodass man die in einem Eckpunkt zusammenlaufenden Kanten beim Abarbeiten der zugehörigen DCEL im mathematisch positiven Sinn durchläuft.

Der Grundalgorithmus zur Bestimmung innerer Elemente ist in Abbildung 4.19 dargestellt. Als Input ist ein Eckpunkt v anzugeben, von dem aus die Suche startet. In einer Schleife über die zugehörige DCEL werden die von ihm ausgehenden Kanten e durchlaufen. Für sie wird jeweils zunächst die Orientierung der DCE ermittelt, indem überprüft wird, ob die Ausgangsecke $e.v_1$ oder $e.v_2$ entspricht. Damit sind der benachbarte Eckpunkt w sowie die (vom Körperäußeren gesehen) links und rechts angrenzenden Polygone f_l und f_r bekannt.

Anschließend wird durch eine `srchmvip(f_r, f_l, v, w)` genannte, im Weiteren erläuterte Funktion festgestellt, ob e von einem Schnittpolygonzug geschnitten wird und w damit außerhalb des Kontaktgebiets liegt (vgl. Abbildung 4.22). Wenn dies der Fall ist, werden w und f_l verworfen und die nächste DCE der Liste verarbeitet. Sonst werden sie den aktiven Elementen hinzugefügt, sofern sie nicht bereits dazugehören. Ist w ein neuer aktiver

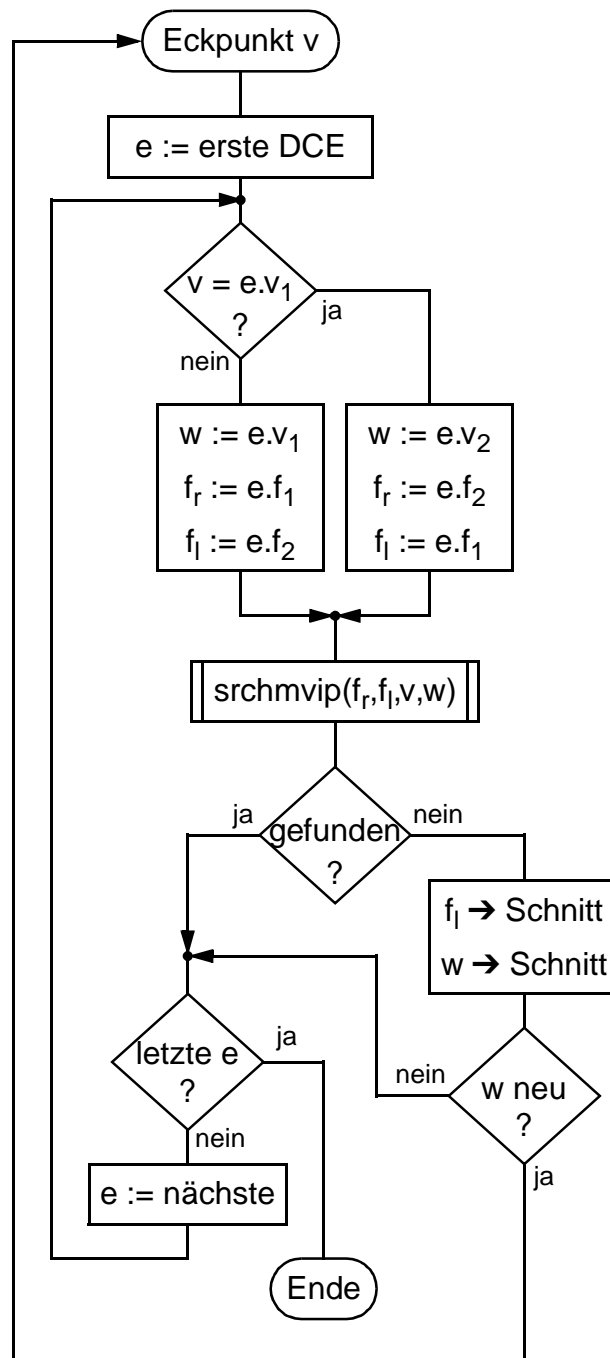


Abbildung 4.19: Grundalgorithmus zur Bestimmung innerer Elemente (hier nur Fläche E ; für F andere Zuordnung von w wegen umgekehrtem Umlaufsinn des Schnittpolygons)

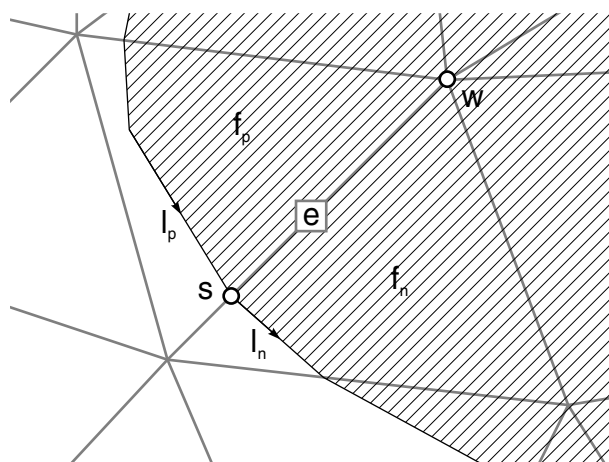


Abbildung 4.20: Geometrische Anordnung zur Ermittlung im Schnittgebiet liegender Eckpunkte w (Draufsicht Fläche E , Schnittgebiet schraffiert)

Eckpunkt, ruft sich der Algorithmus rekursiv mit w als Argument auf. Auf diese Weise breitet sich das Gebiet der bereits bestimmten aktiven Elemente innerhalb der Schnittpolygonzüge aus.

Bei komplex geformten Berührgebieten ist es nicht ausreichend, den Grundalgorithmus einmalig an einem beliebigen inneren Eckpunkt zu starten, weil „Engstellen“ dazu führen können, dass die rekursive Ausbreitung abbricht. Deshalb wird er für alle inneren Eckpunkte der von den Schnittpolygonzügen geschnittenen Polygone ausgeführt. Hierzu werden die Schnittlinien in Polygonzug-Umlaufrichtung abgearbeitet und in aufeinanderfolgenden Paaren betrachtet (Abbildungen 4.20 und 4.21).

In der Regel ändert sich zwischen den benachbarten Schnittlinien nur auf einer der beiden Flächen die ID der geschnittenen Polygone f_p und f_n (vgl. Abbildung 4.15). Dort wird über die DCEs der Eckpunkte von f_p nach der dazwischen liegenden, d. h. diese beiden Polygone trennenden Kante e gesucht. Über die in Abschnitt 4.2.1 getroffene Konvention der Umlaufrichtung der Schnittpolygonzüge liefert diese DCE eindeutig den im Schnittgebiet liegenden Endpunkt w der Kante. Findet $\text{srchmvip}(f_p, f_n, s, w)$ keinen Schnittpolygonzug zwischen dem gemeinsamen Eckpunkt s des Linienpaars und w , wird der Grundalgorithmus mit w als Argument aufgerufen.

Die erwähnte Funktion $\text{srchmvip}(f_r, f_l, p, w)$ (*search/move intersection polygons*) ist von entscheidender Bedeutung, da sie das Abbruchkriterium für die Suche nach inneren Eckpunkten liefert. Als Argumente verarbeitet sie zwei Punkte p und w , die auf der gemeinsamen Kante zweier Polygone f_r

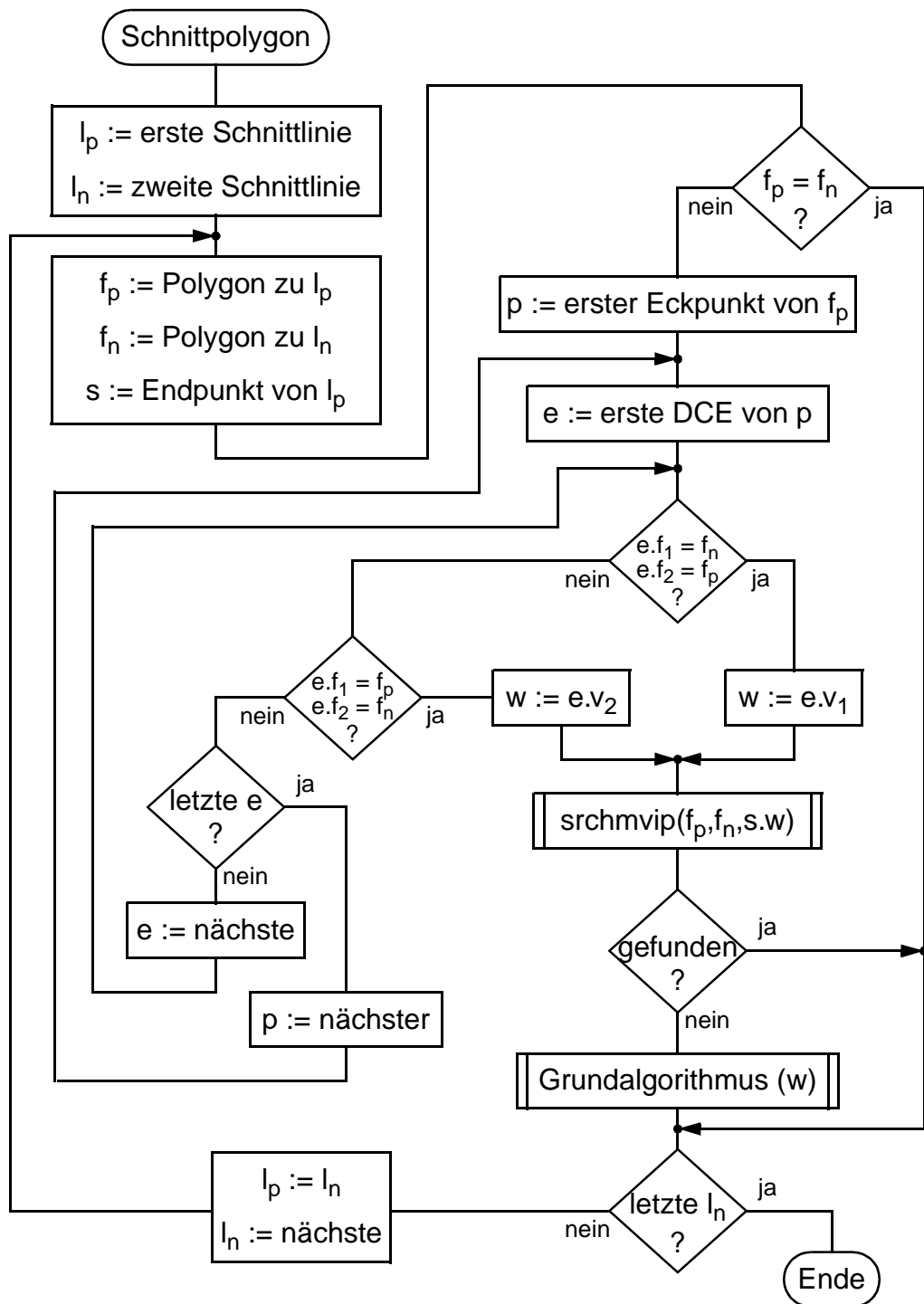


Abbildung 4.21: Aufruf des Grundalgorithmus entlang der inneren Eckpunkte eines Schnittpolygonzugs (hier nur Fläche E ; F wird parallel auf identische Weise bearbeitet)

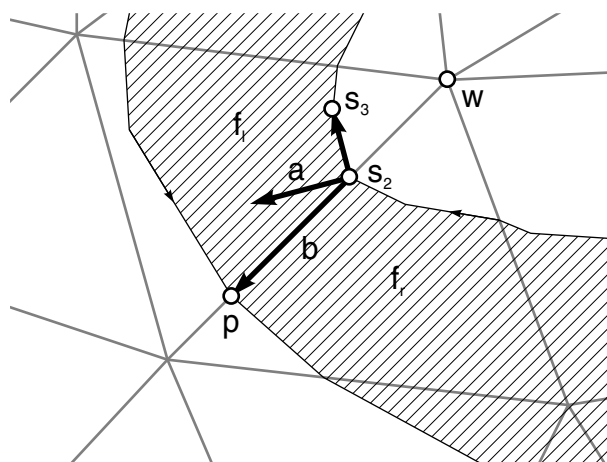


Abbildung 4.22: Geometrische Anordnung der Argumente der Funktion `srchmvip` (Draufsicht Fläche E)

und f_l liegen (Abbildung 4.22). `srchmvip` dient dazu herauszufinden, ob diese Kante zwischen p und dem Kandidaten für einen inneren Eckpunkt w von einem Schnittpolygonzug geschnitten wird, sodass w außerhalb des Schnittvolumens liegt (*search*). Ist dies der Fall, ordnet `srchmvip` den Polygonzug dem aktuellen Schnitt zu (*move*), sofern er nicht bereits dazugehört.

Bei jedem Aufruf von `srchmvip` werden alle Schnittpolygonzüge (analog der Ermittlung der Startpunkte für den Grundalgorithmus) in aufeinanderfolgenden Schnittlinienpaaren durchsucht. Nur diejenigen Paare, die von f_r nach f_l verlaufen, werden weiter betrachtet, weil sie die Kante so gerichtet durchlaufen, dass w auf ihrer äußeren Seite liegt. Unter diesen Kandidaten wird dasjenige Paar bestimmt, das zwischen p und w verläuft und in minimalem Abstand zu p schneidet (sodass der zugehörige Schnittpolygonzug das betrachtete Schnittgebiet begrenzt). Diese Kriterien werden anhand zweier Hilfsvektoren \mathbf{a} und \mathbf{b} (Abbildung 4.22) ausgewertet:

$$\mathbf{a} = \mathbf{n}_{f_l} \times (\mathbf{s}_3 - \mathbf{s}_2)$$

$$\mathbf{b} = \mathbf{p} - \mathbf{s}_2$$

Falls $\mathbf{a} \cdot \mathbf{b} < 0$ gilt, liegt der Schnittpunkt \mathbf{s}_2 nicht zwischen p und w und das Paar wird verworfen. Sonst gibt der Betrag von \mathbf{b} den gesuchten Abstand an.

Wird auf diesem Weg ein Paar (ggf. mit minimalem Abstand) gefunden, begrenzt der zugehörige Schnittpolygonzug das Schnittgebiet. In diesem Fall liefert `srchmvip` den Rückgabewert „gefunden“, sodass w von der aufrufenden Routine nicht als innerer Eckpunkt behandelt wird. Falls der Polygonzug

noch nicht zum Schnitt gehört, wurde eine neue Grenze eines mehrfach berandeten Kontaktgebiets gefunden; dann wird er dem Schnitt zugeordnet und alle von ihm geschnittenen Polygone mit ihren Eckpunkten werden zu den aktiven Elementen hinzugezählt (in Abbildung 4.21 nicht dargestellt).

Bleiben nach der skizzierten Prozedur Schnittpolygonzüge übrig, wird sie jeweils für zusätzliche Schnitte wiederholt, bis keine Polygonzüge mehr übrig sind. Jeder Schnitt entspricht schließlich einem Kontaktgebiet, das durch beliebig viele Randpolygone begrenzt sein kann.

Wie bereits im Abschnitt 4.2.1 erwähnt, können unvollständige Oberflächenbeschreibungen zu offenen Schnittpolygonzügen führen. Solche wirklichkeitsfremden Kontaktsituationen sind physikalisch undefiniert, sodass keine korrekte Auswertung möglich ist. Zwar funktioniert der Algorithmus prinzipiell auch für offene Polygonflächen, doch auf dem gegenüberliegenden Körper fehlt dann ein Teil der Eingrenzung des Kontaktgebiets, sodass sich die Menge der aktiven Elemente über die gesamte Oberfläche ausbreitet. Im günstigsten Fall arbeitet das PCM dadurch nur weniger effizient, aber i. Allg. werden falsch kombinierte Schnitte generiert oder die anschließende Bestimmung der Kontaktelemente schlägt fehl. Aus diesen Gründen gibt das PCM grundsätzlich eine Warnmeldung aus, wenn Schnittpolygonzüge nicht geschlossen werden können.

Das aufgezeigte Verfahren wurde schrittweise entwickelt, um die Anforderungen zahlreicher Testsimulationen zunehmender Komplexität zu erfüllen, wobei lediglich das Konzept der DCEL aus der Literatur übernommen wurde. Wie bei der Kollisionserkennung stand hierbei Robustheit sowie möglichst allgemeine Datenstrukturierung im Vordergrund und Effizienz erst an zweiter Stelle. Einzelheiten der Implementierung lassen sich im über [74] erhältlichen Quellcode nachvollziehen.

4.3 Bestimmung der Kontaktelemente

Nach der Konstruktion der Schnittvolumina liegen die Schnittpolygonzüge und aktiven Elemente (Polygone, die ganz oder teilweise innerhalb der gegenüberliegenden Oberfläche liegen, sowie deren Eckpunkte) in Schnitten (Datenstrukturen, die alle Objekte eines Kontaktgebiets enthalten) zusammengefasst vor. In diesem Abschnitt werden zwei verschiedene Verfahren, die *Master-Slave-Methode* und die *Variational-Interpolation-Methode* beschrieben, durch die das PCM die Berührflächen approximiert und diskretisiert, um anhand der dabei entstehenden Kontaktelemente anschließend die Kontaktkräfte zu berechnen. Die beiden Methoden unterscheiden sich zwar erheblich im Ansatz, erfüllen jedoch dieselbe Aufgabe und arbeiten mit den formal gleichen Eingangs- und Ausgangsdaten.

Das Prinzip der Kontaktelemente wird im Wesentlichen von der Halbraumapproximation (s. Abschnitt 3.4.1) übernommen: Die im Berührgebiet gemeinsame Oberfläche der Körper wird durch ebene Flächenelemente approximiert. Dabei wird näherungsweise davon ausgegangen, dass die Berührfläche so zwischen den unverformten Körperoberflächen verläuft, dass sie diese entsprechend einer Reihenschaltung der elastischen Steifigkeiten der Körper voneinander trennt. Dazu wird das statische Verhältnis

$$\lambda = \frac{u_E}{u_E + u_F} \in [0, 1] \quad (4.11)$$

der Eindringtiefen u_E , u_F der betrachteten Körper E und F in Abhängigkeit des verwendeten Kraftgesetzes der Kontaktelemente (s. Abschnitt 4.4.1) vorab festgelegt.

4.3.1 Master–Slave–Methode

Die Generierung der Kontaktelemente nach der Master–Slave–Methode ist ein sehr einfaches Verfahren. Für seine Anwendung ist vom Benutzer eine der beiden Oberflächen als *Master* festzulegen, während die andere als *Slave* behandelt wird. Dann generiert das PCM während der Simulation für jedes aktive Polygon der Master–Fläche (im Folgenden z. B. Polygon e der Fläche E) ein Kontaktelement k mit den folgenden Eigenschaften (Abbildung 4.23).

Die Fläche A_k wird derjenigen des Master–Polygons e gleichgesetzt:

$$A_k := A_e \quad (4.12)$$

Auch der Normalenvektor \mathbf{n}_k wird von e übernommen, wobei das Vorzeichen stets so gesetzt wird, dass er (wie die resultierende Kontaktkraft) zum Inneren des Körpers E weist:

$$\mathbf{n}_k := -\mathbf{n}_e \quad (4.13)$$

Als Eindringtiefe u_{nk} wird der Euklidische Abstand vom Schwerpunkt C_e des Master–Polygons zum Schnittpunkt I seiner Schwerpunkt–Normalen

$$\mathbf{r} = \mathbf{r}_{C_e} + d \mathbf{n}_e \quad (d \in \mathbb{R}) \quad (4.14)$$

mit der gegenüberliegenden Slave–Fläche F verwendet. Dazu sucht das PCM in den aktiven Elementen von F wie folgt nach dem geschnittenen Slave–Polygon f :

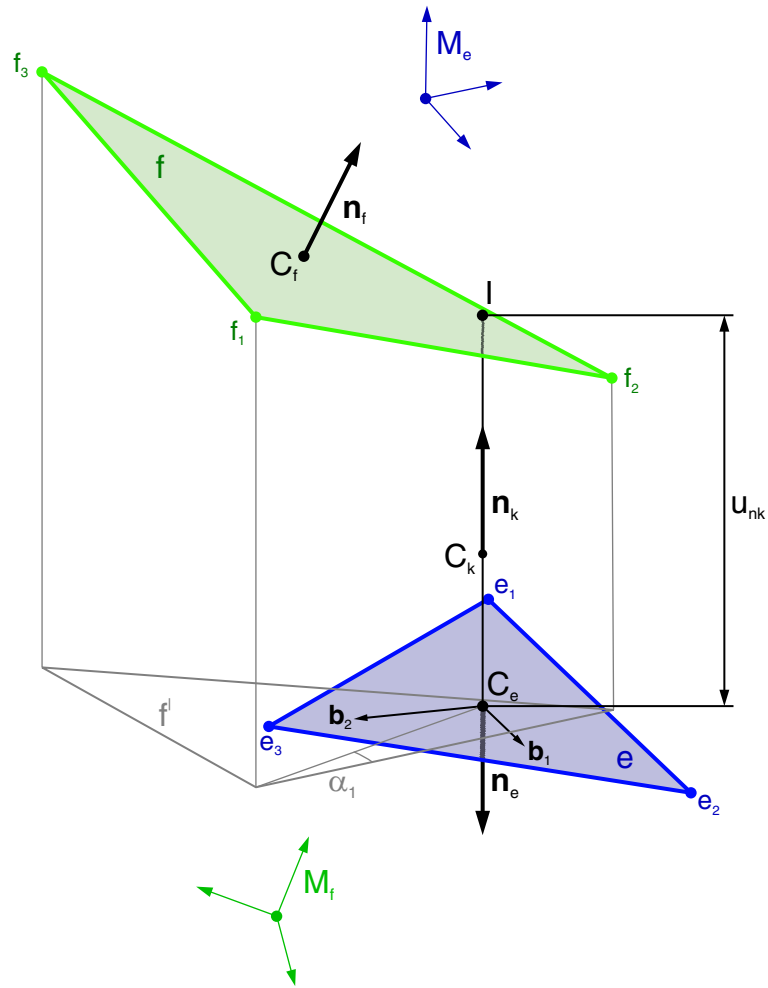


Abbildung 4.23: Geometrische Größen bei der Generierung von Kontaktelementen nach der Master-Slave-Methode; das Master-Polygon e (im Bild unten) gehört zum oben liegenden Körper E , vgl. Abbildung 4.24

Zunächst wird eine ebene Basis $(\mathbf{b}_1, \mathbf{b}_2)$ in e berechnet:

$$\begin{aligned}\mathbf{b}_1 &:= \frac{\mathbf{r}_{e_2} - \mathbf{r}_{e_1}}{|\mathbf{r}_{e_2} - \mathbf{r}_{e_1}|} \\ \mathbf{b}_2 &:= \mathbf{n}_e \times \mathbf{b}_1\end{aligned}$$

Dabei steht \mathbf{r}_{e_i} für den Ortsvektor des Eckpunkts e_i ($i \in [1, 2, 3]$) von e . Um zu entscheiden, welches Slave-Dreieck f die Gerade \mathbf{r} aus (4.14) schneidet, wird das Problem zweidimensional betrachtet. Durch die Abbildung

$$\mathbf{r}'_{C_e f_i} := [\mathbf{b}_1 \ \mathbf{b}_2]^T (\mathbf{r}_{f_i} - \mathbf{r}_{C_e}) \quad (4.15)$$

werden die Eckpunkte f_i jedes Kandidaten f in die Ebene $(\mathbf{b}_1, \mathbf{b}_2)$ von e projiziert. Damit schneidet die Normale \mathbf{r} das Dreieck f , wenn der Schwerpunkt C_e innerhalb der Projektion f' von f liegt. Dies ist genau dann erfüllt, wenn alle drei Winkel α_i von den Seitenvektoren $\mathbf{r}'_{f_i f_{i+1}} = \mathbf{r}'_{C_e f_{i+1}} - \mathbf{r}'_{C_e f_i}$ zu den umgekehrten Eckpunktvektoren $\mathbf{r}'_{f_i C_e} = -\mathbf{r}'_{C_e f_i}$ positiv sind. Somit gilt für ein von \mathbf{r} geschnittenes Slave-Dreieck f :

$$\det \begin{bmatrix} \mathbf{r}'_{f_i f_{i+1}} & \mathbf{r}'_{f_i C_e} \end{bmatrix} > 0 \quad \forall i \in [1, 2, 3]. \quad (4.16)$$

Ist f gefunden, wird die Eindringtiefe u_{nk} des Kontaktelements anhand von (4.14) und der Hesse-Normalform der Ebene von f bestimmt:

$$u_{nk} := \frac{(\mathbf{r}_{C_f} - \mathbf{r}_{C_e}) \cdot \mathbf{n}_f}{\mathbf{n}_e \cdot \mathbf{n}_f} \quad (4.17)$$

Falls die Slave-Fläche Hinterschnidungen im Kontaktgebiet aufweist, kann die Bedingung (4.16) im Prinzip von mehreren aktiven Dreiecken der Slave-Fläche erfüllt sein. Dann wird der Kandidat mit der kleinsten Eindringtiefe u_{nk} verwendet.

Schließlich ist noch die Position C_k des Kontaktelements zu bestimmen. Dieser Punkt soll näherungsweise auf der Berührfläche liegen und wird im Weiteren als Ort zur Ermittlung des Relativgeschwindigkeitsvektors \mathbf{v}_k der Oberflächen und zum Einprägen des resultierenden Kontaktkraftvektors \mathbf{F}_k herangezogen. Das angenommene Eindringtiefenverhältnis (4.11) führt zur Kontaktelementposition:

$$\mathbf{r}_{C_k} := \mathbf{r}_{C_e} + \lambda \cdot u_{nk} \cdot \mathbf{n}_k. \quad (4.18)$$

Abbildung 4.24 zeigt die nach der Master-Slave-Methode generierten Kontaktelemente eines einfachen Schnittgebiets. In der Anwendung hat sich dieses Verfahren als effiziente und robuste Lösung erwiesen. Wegen seiner unsymmetrischen Natur, bei der nur die Master-Polygone die Richtungen und Flächen der Kontaktelemente bestimmen, ist es allerdings insbesondere bei größeren Winkeldifferenzen zwischen Master- und Slave-Oberfläche mit gewissen geometrischen Ungenauigkeiten behaftet.

Da die Master-Polygonfläche weitgehend die Diskretisierung bestimmt, hat sie erheblichen Einfluss auf die Funktion des PCM: Sehr grobe Flächen lassen sich schnell auswerten, führen jedoch zu schlechterer Approximation, während übermäßig fein polygonalisierte Flächen erhebliche Effizienzeinbußen verursachen. Daher sollte der Anwender die Master-Slave-Zuordnung vor allem bei deutlichen Unterschieden der Polygongröße der beiden Flächen sorgfältig treffen.

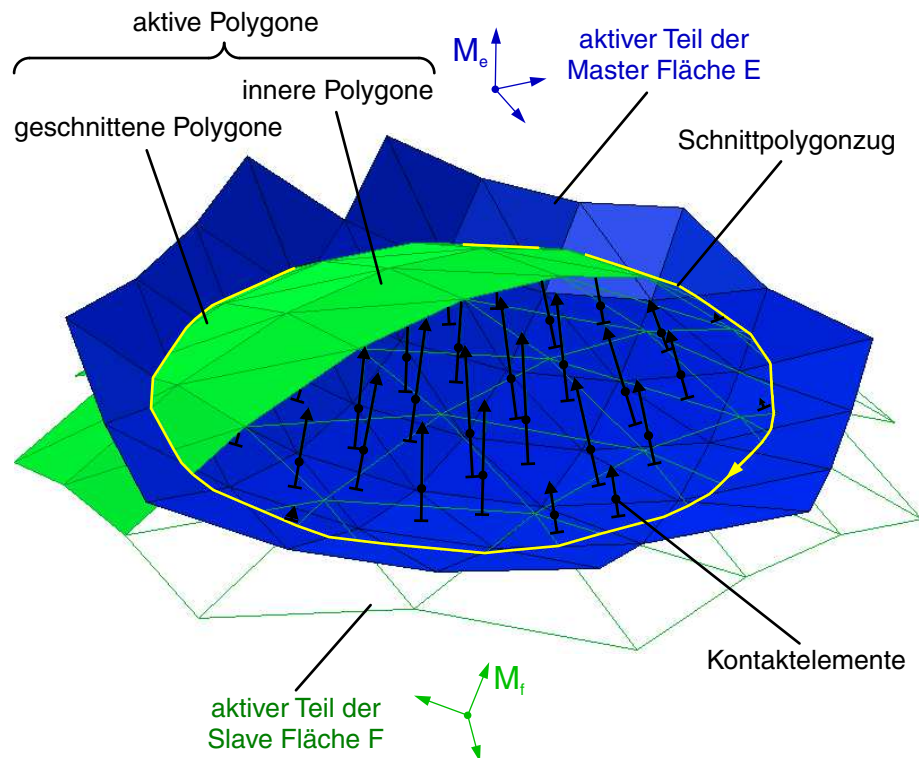


Abbildung 4.24: Kontakt der Scheitel zweier Paraboloiden mit Kontakt-elementengenerierung nach der Master-Slave-Methode

Zur Veranschaulichung kann man sich vorstellen, dass die Oberfläche des Master-Körpers alleine aus elastisch gelagerten Druckstäben in den Polygon-schwerpunkten gebildet wird, während der Slave-Körper tatsächlich seiner Polygonoberfläche entspricht. Dies lässt sich auch ausnützen, indem ein komplex geformter Körper als Master und sein einfach durch Polygone darstellbares Gegenüber (z. B. der Boden als Ebene, die aus einem einzigen großen Dreieck besteht) als Slave modelliert werden. Mit umgekehrter Zuordnung kann ein solches Modell allerdings nicht korrekt funktionieren, weil die einfache Fläche als Master zu einer zu groben Diskretisierung der Berührfläche führt; im Extremfall bewegt sich der Slave-Körper ohne Kontaktkraftwirkung zwischen den Kontaktelelementen hindurch.

Sind beide Oberflächen in etwa gleich fein polygonalisiert, kann das PCM auch im sog. *2-Pass-Modus* betrieben werden. Dann werden Kontaktelementengenerierung und -auswertung zuerst mit Fläche E und danach mit Fläche F als Master ausgeführt und der Mittelwert der beiden resultierenden Kraftwinder als Ergebnis verwendet. So erhält man bei etwa doppeltem Rechenaufwand eine verbesserte Approximation, da sich der oben erwähnte unsymmetrische Charakter des Master-Slave-Ansatzes aufhebt.

4.3.2 Variational–Interpolation–Methode

Mit der Variational–Interpolation–Methode wurde ein alternatives Verfahren zur Kontaktelementgenerierung entwickelt und implementiert. Da es in der Anwendung deutliche Schwächen zeigte, wurde es jedoch nicht weiter verfolgt. Aus diesem Grund wird die Methode im Folgenden auch nur grob skizziert; für eine detaillierte Beschreibung der verwendeten Algorithmen, der Implementierung und eines Anwendungsbeispiels wird auf [75] verwiesen.

Abbildung 4.25 zeigt an einem einfachen zweidimensionalen Beispiel die vier wesentlichen Schritte des Variational–Interpolation–Verfahrens, welche im Weiteren kurz beschrieben werden.

Interpolation der aktiven Flächen

Im ersten Schritt werden die aktiven Flächenteile interpoliert, sodass für jeden Schnitt eine implizite Funktion f_E durch die Eckpunkte auf der Fläche E und eine weitere Funktion f_F durch diejenigen der Fläche F entsteht. Dabei werden zusätzlich auf beiden Seiten die Eckpunkte der zugehörigen Schnittpolygonzüge als Stützstellen vorgegeben.

Die Ermittlung der Funktionen erfolgt durch *Variational Interpolation* [76, 77, 78, 79, 80, 81]. Dieses erst seit wenigen Jahren in der Computergrafik angewendete Verfahren erlaubt eine sehr glatte und „weiche“ Interpolation von Polygonflächen. Als Ergebnis erhält man eine Feldfunktion, welche als in impliziter Form

$$f(\mathbf{x}) = 0 \quad (4.19)$$

dargestellte Fläche durch sämtliche Eckpunkte \mathbf{c}_i der Polygone verläuft.

Aus (4.19) geht bereits ein Teil der Interpolationsbedingungen hervor: Die gesuchte Funktion f muss für jede Eckpunktposition $\mathbf{x} = \mathbf{c}_i$ verschwinden. Um eine robuste Interpolation zu erreichen, werden beim PCM zu jedem Eckpunkt zwei weitere Stützpunkte generiert, die in geringem Abstand $\delta = 0,01$ auf der gemittelten Normalen der benachbarten Polygone liegen und die Funktionswerte $f = \pm 10$ vorgeben (*normal constraints*) [76].

Hinter dem Begriff Variational Interpolation steht die Ansatzfunktion

$$f(\mathbf{x}) = \sum_i d_i \Phi(\mathbf{x} - \mathbf{c}_i) + p_0 + p_1x + p_2y + p_3z \quad (4.20)$$

mit radialen Basisfunktionen

$$\Phi(\mathbf{x}) = \mathbf{x}^2 \log(|\mathbf{x}|) \quad (4.21)$$

welche sich mittels Variationsrechnung aus einer Analogie zur minimalen Verformungsenergie einer elastischen Platte ableitet (*thin-plate interpolation*) [76]. Einsetzen der Interpolationsbedingungen an den Stützpunkten \mathbf{c}_i führt zu einem linearen, eindeutig lösbareren Gleichungssystem für die Wichtungsfaktoren d_i und die Polynomkoeffizienten p_j .

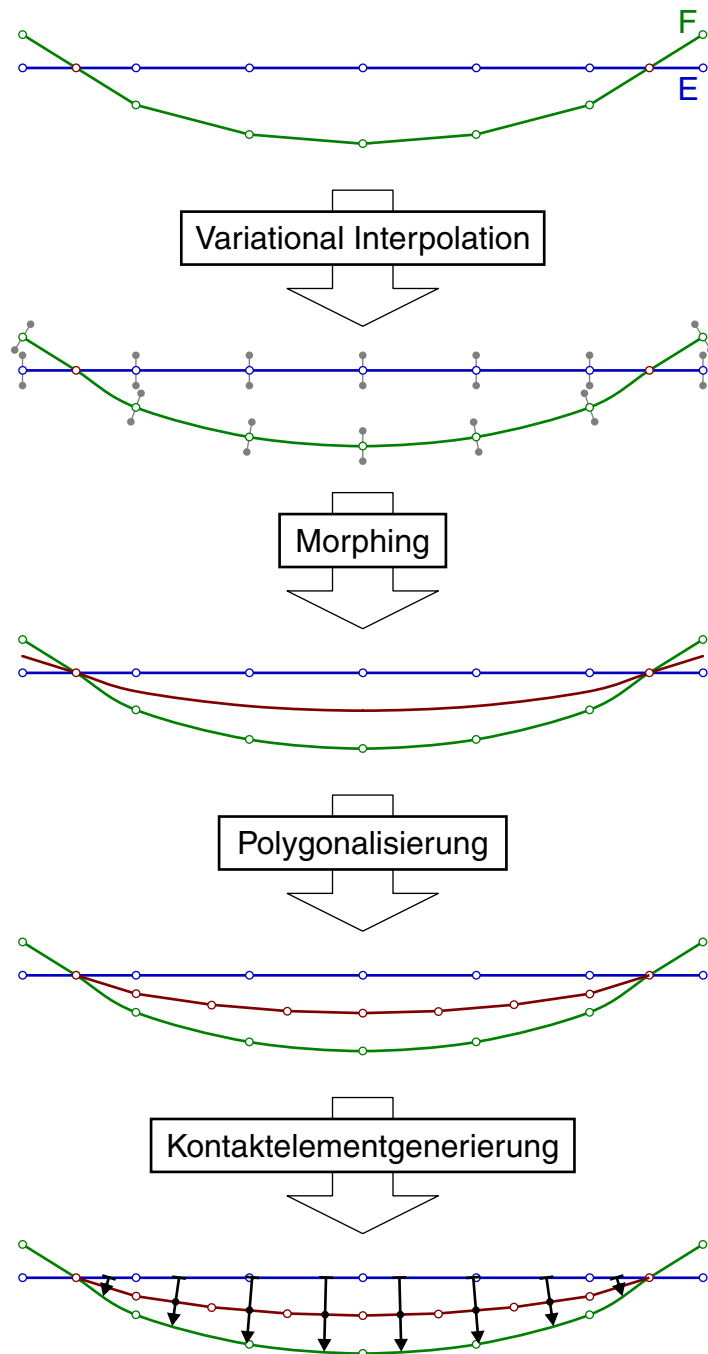


Abbildung 4.25: Die Kontaktelementgenerierung nach der Variational-Interpolation-Methode erfolgt in vier Schritten

Überblendung der Berührfläche

Die Überführung der aktiven Flächenteile in implizite Darstellung ist ein vorbereitender Schritt zur Überblendung der Berührfläche. Dabei wird in Anlehnung an ein *Morphing*-Verfahren aus der Computergrafik [82] ein stufenloser Übergang zwischen zwei impliziten Flächen mittels Linearkombination erzielt. Die grundlegende Idee ist, diese Methode zur Verwandlung von einer zu einer anderen Oberfläche auf die aktiven Flächenteile der gefundenen Schnitte anzuwenden:

$$f_B(\mathbf{x}) := \lambda f_E(\mathbf{x}) + (1 - \lambda) f_F(\mathbf{x}) \quad (4.22)$$

So erhält man mit λ nach (4.11) eine Approximation f_B der Berührfläche.

Triangulierung der Berührfläche

Um das Kontaktproblem der numerischen Behandlung zugänglich zu machen, wird die in impliziter Darstellung vorliegende Berührfläche trianguliert, d. h. in eine aus Dreiecken bestehende Polygonfläche überführt. Zur Lösung dieser Aufgabe existieren zahlreiche Algorithmen, die sich in erster Linie in *volumenbasierte* [83, 84] und *flächenbasierte* [85, 86, 87, 88, 89] Ansätze unterteilen lassen. Für die Variational-Interpolation-Methode wurde das flächenbasierte *Marching Triangles* Verfahren nach Hartmann [88] implementiert, weil es eine gleichmäßige Diskretisierung bei geringem Rechenaufwand verspricht.

Bei der Triangulierung werden die gefundenen Schnitte sequenziell abgearbeitet. Zunächst werden die zugehörigen Schnittpolygonzüge in initiale *Frontpolygone* überführt, deren Eckpunkte in etwa im Abstand der für die Triangulierung angestrebten Seitenlänge δ_t auf den Schnittlinien liegen. Wie ihr Name andeutet, trennen die Frontpolygone im Weiteren den bereits triangulierten Teil der Schnittfläche von den noch zu bearbeitenden Abschnitten und schnüren diese nach und nach ein.

Die Grundoperation des Algorithmus wird sukzessive immer am Frontpolygoneckpunkt P_0 mit dem größten *Frontwinkel* ω zwischen allen benachbarten Linien ausgeführt (Abbildung 4.26). Zuerst werden die zugehörigen Eckpunkte in die Tangentialebene, die durch den Gradienten der Feldfunktion gegeben ist, projiziert. Dann wird der dazwischen liegende Winkelbereich so oft unterteilt, dass sich eine möglichst geringe Abweichung vom angestrebten Triangulierungswinkel von 60° ergibt. Ausgehend von im Abstand δ_t vom Ausgangspunkt auf diesen Polarlinien liegenden Punkten q_i ergeben sich die neuen Eckpunkte P_i der Polygonfläche durch Rückprojektion von der Tangentialebene auf die implizite Fläche.

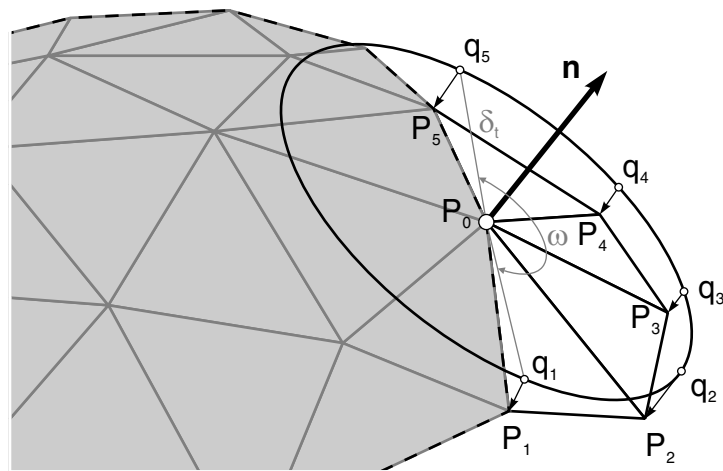


Abbildung 4.26: Generierung neuer Dreiecke am Eckpunkt P_0 mit dem größten Frontwinkel ω des gestrichelt dargestellten Frontpolygons

Vor jeder Grundoperation ermittelt der Algorithmus außerdem die paarweisen Abstände zwischen den Eckpunkten der Frontpolygone und schließt bei Unterschreitung der Referenzlänge δ_t die Lücke, sodass ein Frontpolygon geteilt werden kann oder zwei Frontpolygone zu einem einzigen verschmelzen können [75, 88]. Die Triangulierung ist abgeschlossen, wenn sämtliche Frontpolygone bearbeitet wurden. Abbildung 4.27 zeigt die Triangulierungen einer Berührfläche für zwei verschiedene Referenzlängen.

Erzeugung der Kontaktelemente

Liegt die Berührfläche in triangulierter Form vor, lassen sich sehr einfach Kontaktelemente generieren. Jedes Dreieck ergibt ein Element mit identischem Flächeninhalt A_k und Normalenvektor \mathbf{n}_k sowie dem Ort seines Schwerpunkts als Position C_k . Zur Bestimmung der Eindringtiefe u_{nk} lässt sich prinzipiell der entlang des Normalenvektors gemessene Abstand zwischen den unverformten Oberflächen in polygonaler oder impliziter Form heranziehen. Für das PCM wurde eine iterative Berechnung der Schnittpunkte der Normalen mit den interpolierten Flächen implementiert.

Die Generierung der Kontaktelemente mittels Variational Interpolation birgt einige prinzipielle Vorteile gegenüber dem Master–Slave–Verfahren. Insbesondere kann die Auflösung der Diskretisierung der Berührfläche anhand des Parameters δ_t völlig unabhängig von den gegebenen Polygonoberflächen gewählt werden. Außerdem entstehen keine Näherungsfehler durch unsymmetrische Anordnung und Flächenbestimmung der Kontaktelemente. Doch

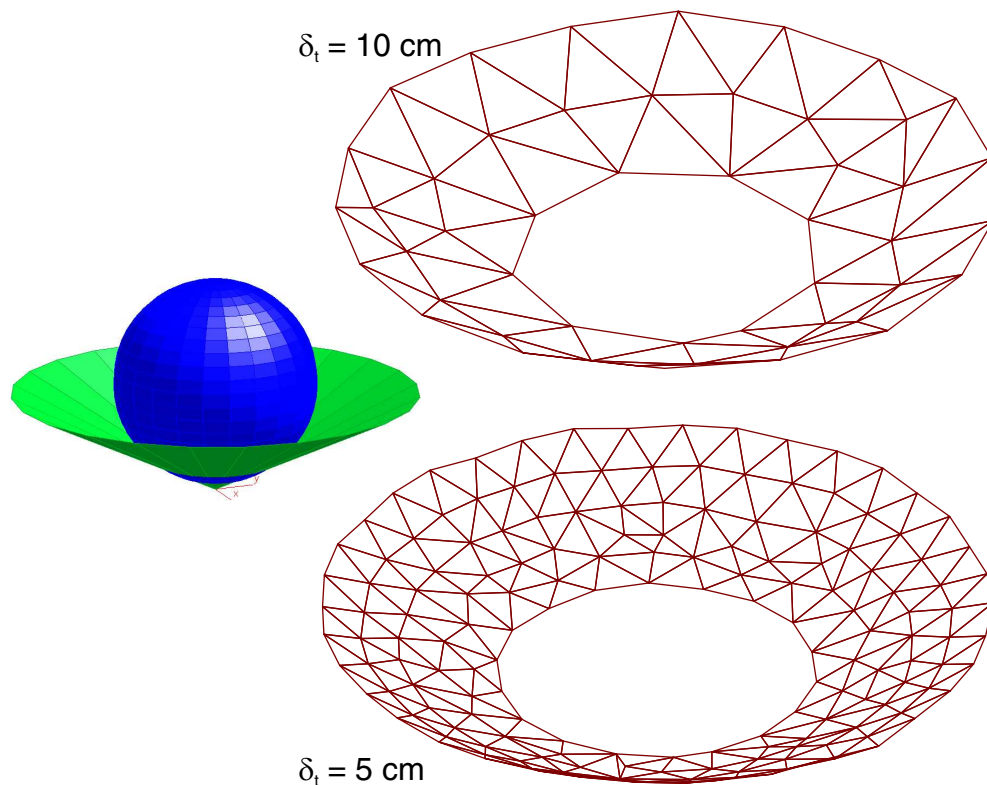


Abbildung 4.27: Triangulierung der doppelt berandeten Berührfläche zwischen einer Kugel und einem Kegel für zwei unterschiedliche Referenzlängen δ_t

nachdem Variational Interpolation zunächst als favorisierte Lösung entwickelt worden war, wurde nach den ersten praxisnahen Vergleichssimulationen die Master–Slave–Methode bevorzugt.

Als entscheidender Nachteil des Verfahrens ist seine Untauglichkeit für MKS–Simulationen unter Verwendung impliziter Integratoren zu nennen (s. Abschnitt 2.3). Während sich die Master–Slave–Methode ähnlich problemlos wie eindimensionale Krafelemente (s. Abschnitt 3.5.1) erwies, führte Variational Interpolation zu extrem kleinen Integrationsschrittweiten im Mikrosekundenbereich.

Für dieses Verhalten gibt es mehrere Ursachen. Erstens können kleinste Veränderungen der relativen Lage der Körper eine ganz unterschiedliche Triangulierung des Kontaktgebiets nach sich ziehen, sodass sich verhältnismäßig große Abweichungen im resultierenden Kontaktkraftvektor ergeben. Zweitens verändert sich während dynamischer Kontaktvorgänge ständig die

Untermenge der aktiven Polygone der Oberflächen. Dabei bewirkt jede Zu- oder Abschaltung eines Polygons sprunghaft neue bzw. wegfallende Stützpunkte der Interpolation, sodass die resultierende Berührfläche unmittelbar eine neue Gestalt annimmt. Und schließlich erscheint die Verwendung von Newton-Iterationen zur Projektion neuer Eckpunkte und zur Bestimmung der Kontaktelementeindringtiefen problematisch, weil die durch das PCM berechnete Kontaktkraft in die Funktionsauswertung des übergeordneten Newton-Verfahrens des Integrators eingeht.

Numerisch betrachtet bleibt der als Differenz zwischen berechnetem und tatsächlichem Kontaktkraftvektor definierte Modellfehler

$$\Delta_F(\mathbf{p}) := \mathbf{F}_{PCM}(\mathbf{p}) - \mathbf{F}_{real}(\mathbf{p}) \quad (4.23)$$

bei der Variational-Interpolation-Methode zwar beschränkt, hängt aber nicht glatt von den Lagekoordinaten \mathbf{p} des Systems ab. Somit führt die numerische Differentiation zur Bestimmung der Jacobimatrix der Systemfunktion (s. Abschnitt 2.3) häufig zu unbrauchbaren Ergebnissen und folglich erheblich beeinträchtigter Korrektoiteration der Zeitintegration.

Neben den numerischen Schwierigkeiten im Zusammenspiel mit impliziten Integratoren erwies sich die Variational-Interpolation-Methode auch als zu rechenaufwändig. Selbst bei moderater Vorgabe der Dreiecksgröße für die Triangulierung führte das Verfahren mitunter zu Rechenzeiten im Sekundenbereich für eine einzige Funktionsauswertung. Damit sind keine Simulationen mit MKS-typisch hoher Effizienz mehr möglich.

Wegen der genannten Probleme erwies sich die Übertragung der Variational-Interpolation von der Computergrafik auf die Kontaktkraftberechnung in MKS als kaum praktikabel. In den Anwendungen (s. Kapitel 5) wird deshalb das erheblich vorteilhaftere Master-Slave-Verfahren bevorzugt.

4.4 Bestimmung der Kontaktkraft

Die Berechnung der Kontaktkraft orientiert sich im Wesentlichen am Rand-schichtmodell (s. Abschnitt 3.4.2), kombiniert mit Ideen der Halbraumapproximation (s. Abschnitt 3.4.1) und von Kontakt-Kraftelementen (s. Abschnitt 3.5.1).

Die zuvor beschriebene Diskretisierung der Berührfläche in Kontaktelemente ähnelt im geometrischen Prinzip der Halbraumapproximation. Allerdings bringen die Verfahren des PCM den Vorteil mit sich, dass keine Berührebene benötigt wird, sondern beliebig geformte Kontaktgebiete näherungsweise entlang der tatsächlichen Berührfläche unterteilt werden.

Hinsichtlich der mathematischen Modellbildung orientiert sich das PCM dagegen am einfacheren Randschichtmodell, bei dem von starren Körpern ausgegangen wird, welche mit linear-elastischen Randschichten überzogen sind. Dies führt in Kombination mit der Diskretisierung der Berührfläche zu voneinander unabhängigen Kontaktelementen, deren Kraftgesetze weitgehend denen eindimensionaler Kontakt-Kraftelemente entsprechen.

Neben der für das Randschichtmodell typischen Entkopplung der Kontaktelemente wird auch die Rückwirkung von tangentialen zu normalen Kontaktspannungen vernachlässigt, d. h. es wird wie bei einfacheren Kontaktmodellen üblich von quasiidentisch elastischen Körpern ($\kappa = 0$ nach (3.14)) ausgegangen.

4.4.1 Bestimmung der Normalkraft

Das PCM implementiert eine erweiterte Version des klassischen Randschichtmodells nach (3.21). Dabei werden beide Starrkörper E und F als von linear-elastischen Randschichten überzogen betrachtet, die durch ihre elastischen Module K_E und K_F nach (3.22) und ihre Dicken b_E und b_F bestimmt werden. Aus diesen Größen berechnet das PCM entsprechend einer Reihenschaltung linearer Druckfedern eine kombinierte Randschichtsteifigkeit

$$c_l := \frac{c_{l_E} \cdot c_{l_F}}{c_{l_E} + c_{l_F}} \quad (4.24)$$

mit

$$\begin{aligned} c_{l_E} &= \frac{K_E}{b_E} = \frac{1 - \nu_E}{(1 + \nu_E)(1 - 2\nu_E)} \cdot \frac{E_E}{b_E} \\ c_{l_F} &= \frac{K_F}{b_F} = \frac{1 - \nu_F}{(1 + \nu_F)(1 - 2\nu_F)} \cdot \frac{E_F}{b_F} \end{aligned} \quad (4.25)$$

sowie das statische Verhältnis (4.11) der elastischen Eindringtiefen

$$\lambda := \frac{c_{l_F}}{c_{l_E} + c_{l_F}} \quad , \quad (4.26)$$

welches in (4.18) bzw. (4.22) bereits zur Bestimmung der Kontaktelemente herangezogen wurde. Insbesondere kann hierbei einer der beiden Körper starr modelliert werden, sodass c_l der Randschichtsteifigkeit des anderen Körpers entspricht und λ den Grenzwert 0 (Körper E starr) bzw. 1 (Körper F starr) annimmt.

Mit (3.21) ergibt sich für ein Kontaktelement unter Annahme konstanter Druckspannung über seine gesamte Fläche A_k

$$F_{ck} := c_l \cdot A_k \cdot u_{nk} \quad (4.27)$$

als elastische Normalkraftkomponente bei der Eindringtiefe u_{nk} . Dieses Kraftgesetz entspricht einem linear-elastischen Kontakt-Kraftelement (3.25) mit der Federsteifigkeit $c_k = c_l \cdot A_k$. Die weiteren Kraftkomponenten wurden im umgekehrten Sinne der Analogie in Anlehnung an die eindimensionalen Kraftelemente entwickelt. So berücksichtigt das PCM die Materialdämpfung der Randschichten anhand einer flächenbezogenen Dämpfungskonstante d_l , sodass sich als dämpfende Normalkraftkomponente

$$F_{dk} := \begin{cases} d_l \cdot A_k \cdot v_{nk} & : u_{nk} \geq u_d \\ d_l \cdot A_k \cdot v_{nk} \cdot \frac{u_{nk}}{u_d} & : u_{nk} < u_d \end{cases} \quad (4.28)$$

ergibt. Dabei steht

$$v_{nk} = \mathbf{n}_k \cdot \mathbf{v}_k \quad (4.29)$$

für den in Normalenrichtung projizierten Anteil der Relativgeschwindigkeit

$$\mathbf{v}_k = \mathbf{v}_{M_e M_f} + \omega_{M_e M_f} \times \mathbf{r}_{M_f C_k} \quad (4.30)$$

der beiden Körper an der Position C_k des Kontaktelements. Mit dem Parameter u_d kann außerdem eine Eindringtiefe vorgegeben werden, bis zu welcher die Dämpfungskraft linear eingeblendet wird. Auf diese Weise lassen sich unrealistische Kraftsprünge bei schnellen Kontaktvorgängen vermeiden.

Zur Berechnung der gesamten Normalkraft F_{nk} wird ein weiterer empirischer Ansatz verwendet, um beim Lösen des Kontakts Zugkräfte auszuschließen:

$$F_{nk} := \begin{cases} F_{ck} + F_{dk} & : F_{ck} + F_{dk} > 0 \\ 0 & : F_{ck} + F_{dk} \leq 0 \end{cases} \quad (4.31)$$

4.4.2 Bestimmung der Tangentialkraft

Die tangentiale Reibungskraft wird in Abhängigkeit der tangentialen Relativgeschwindigkeit

$$\mathbf{v}_{tk} = \mathbf{v}_k - v_{nk} \cdot \mathbf{n}_k \quad (4.32)$$

$$v_{tk} = |\mathbf{v}_{tk}| \quad (4.33)$$

und der Normalkraft F_{nk} des Kontaktelements bestimmt:

$$F_{tk} := \begin{cases} \mu \cdot F_{nk} & : v_{tk} \geq v_\varepsilon \\ \mu \cdot F_{nk} \cdot \frac{v_{tk}}{v_\varepsilon} \left(2 - \frac{v_{tk}}{v_\varepsilon} \right) & : v_{tk} < v_\varepsilon \end{cases} \quad (4.34)$$

Damit wird die in Abbildung 3.15 dargestellte Approximation des Coulombschen Reibgesetzes mit quadratischer Regularisierung implementiert: Fällt die Gleitgeschwindigkeit unter eine gegebene kleine Grenze v_ε , wird die Reibungskraft ausgeblendet, um den mengenwertigen Haftzustand zu vermeiden.

Der gesamte Kraftvektor \mathbf{F}_k des Kontaktelements ergibt sich durch Projektion der Normalkomponente F_{nk} in seine Normalenrichtung \mathbf{n}_k und der tangentialen Komponente F_{tk} in Richtung des tangentialen Relativgeschwindigkeitsvektors \mathbf{v}_{tk} :

$$\mathbf{F}_k = F_{nk} \cdot \mathbf{n}_k + F_{tk} \cdot \frac{\mathbf{v}_{tk}}{v_{tk}} \quad (4.35)$$

Entsprechend der in MKS-Programmen üblichen Konvention berechnet das PCM die im Ursprung des Koppelmarkers M_f auf den Körper E wirkende Kontaktkraft. Somit ergibt sich der resultierende Momentenvektor eines Kontaktelements zu (Abbildung 4.23):

$$\mathbf{M}_k = \mathbf{r}_{M_f C_k} \times \mathbf{F}_k \quad (4.36)$$

Schließlich werden die Kräfte und Momente aller Kontaktelemente zum resultierenden Gesamtkraftwinder aufaddiert:

$$\mathbf{F}_E^{M_f} = \sum_k \mathbf{F}_k \quad (4.37)$$

$$\mathbf{M}_E = \sum_k \mathbf{M}_k \quad (4.38)$$

Die vorgestellte Bestimmung der Kontaktkraft ist strukturell sehr einfach: Es müssen weder dynamische oder diskrete Zustände verwaltet werden, noch sind Zwangsbedingungen oder iterative Schritte erforderlich. Stattdessen lässt sich für einen gegebenen relativkinematischen Zustand der Körper auf Lage- und Geschwindigkeitsebene direkt und eindeutig der resultierende Kontaktkraftwinder berechnen. Das PCM stellt somit sehr niedrige Anforderungen an das MKS-Programm und arbeitet bei zweckmäßiger Diskretisierung verhältnismäßig effizient gegenüber vergleichbaren Kontaktmodellen.

Allerdings eignen sich die in den Abschnitten 4.2 und 4.3 entwickelten geometrischen Verfahren zur Generierung diskretisierter Berührflächen prinzipiell auch für komplexere Kontaktmodelle. So ließe sich das Kraftgesetz der Kontaktelemente für entsprechende Anwendungen durch andere rheologische Modelle ersetzen, um beispielsweise fließendes oder plastisches Materialverhalten zu simulieren. Zur verbesserten Darstellung der Reibung könnte das im Abschnitt 3.5.1 erwähnte Bürstenmodell implementiert werden, welches Haftreibung realitätsnah abbildet. Schließlich böte sich für sehr hohe Genauigkeitsanforderungen die Halbraumapproximation (s. Abschnitt 3.4.1) an, welche die tatsächliche Gestalt der Berührfläche berücksichtigt und partielle Gleitzustände auflösen kann. Da die meisten dieser Methoden Kontaktelemente mit internen Zustandsvariablen erfordern, sind sie allerdings nur mit der Master-Slave-Methode vereinbar, weil nur hier eine feste Zuordnung zwischen Zuständen und Kontaktelementen anhand der Polygon-IDs möglich ist.

Kapitel 5

Simulationsergebnisse

In diesem Kapitel werden mit dem PCM berechnete Simulationsergebnisse vorgestellt und diskutiert. Sie stammen aus umfangreichen Testsimulationen, die die Entwicklung des PCM im fortgeschrittenen Stadium begleiteten und auch beeinflussten. Sämtliche Beispiele sind akademischer Natur, d. h. es werden keine konkreten Probleme aus der Praxis gelöst, sondern selbst definierte Szenarien simuliert, die zielgerichtete Analysen der Kontaktmodellierung ermöglichen. Somit können sie Plausibilität, Robustheit und Effizienz des Verfahrens demonstrieren, nicht aber zur Verifikation anhand gemessener Daten dienen.

Alle Simulationen wurden mit dem kommerziellen MKS-Programm SIMPACK [3] in der Version 8.600 berechnet, wobei das PCM als gewöhnliches User-Kraftelement eingebunden wurde. Die Modelldaten, der PCM-Quellcode (s. Anhang A) sowie die Interface-Routinen zu SIMPACK sind über die PCM-Webseite [74] erhältlich.

5.1 Plausibilität

Die ersten Simulationen wurden mit einem sehr einfachen Modell durchgeführt, welches statischen, dämpfungs- und reibungsfreien Kontakt zwischen einem Rad und dem Boden beschreibt (Abbildung 5.1).

Die Oberfläche des Bodens bilden acht inertial feste, koplanare Dreiecke. Die aus 416 Dreiecken bestehende, das Rad beschreibende Polygonfläche wurde einer freien Internet-Bibliothek [90] entnommen. Das Rad ist 201 mm breit und hat einen Durchmesser von 571 mm.

Für die Kontaktkraftberechnung mit dem PCM wurden folgende Parameter vorgegeben: Am Boden (Fläche F) wird von einer $b_F = 1,0$ mm dicken Randschicht ausgegangen, die einen Elastizitätsmodul von $E_F = 5,0 \cdot$

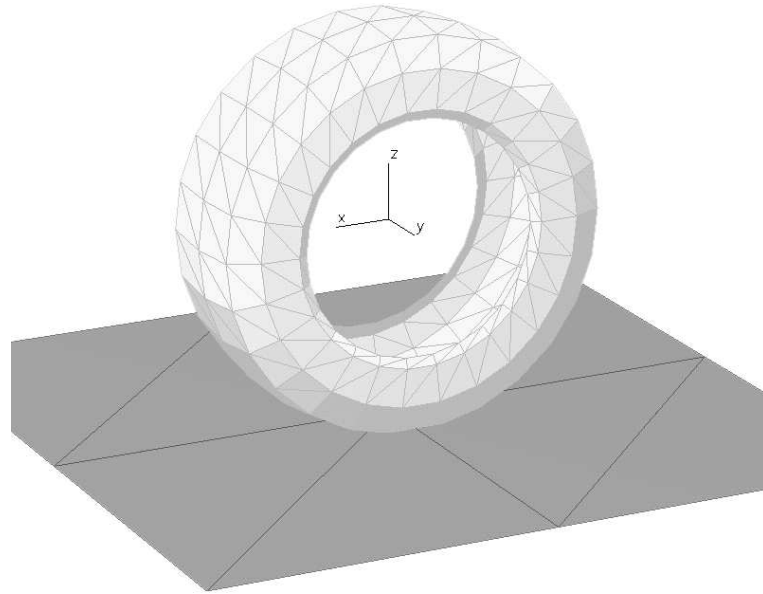


Abbildung 5.1: Modell zur Demonstration der statischen Eigenschaften des PCM: Kontakt zwischen Rad und ebenem Boden

10^6 N/m^2 und eine Querkontraktionszahl von $\nu_F = 0,2$ aufweist, während das Rad (Fläche E) mit $b_E = 1,0 \text{ cm}$, $E_E = 3,0 \cdot 10^6 \text{ N/m}^2$ und $\nu_E = 0,4$ durch eine höhere Nachgiebigkeit gekennzeichnet ist. So ergibt sich nach (4.24) und (4.26) eine kombinierte Randschichtsteifigkeit von $c_l = 5,76 \cdot 10^8 \text{ N/m}^3$ und ein elastisches Eindringtiefenverhältnis von $\lambda = 0,896$.

Zur statischen Analyse dieses Kontaktproblems wurden 500 Auswertungen durchgeführt, zwischen denen der Radmittelpunkt in $0,04\text{-mm}$ -Schritten nach unten versetzt wurde. Da das System keinen mechanischen Freiheitsgrad besitzt und weder Dämpfung noch Reibung berücksichtigt werden, erhält man so ein vollelastisches Kraftgesetz zwischen der Annäherung der Körperreferenzsysteme und der vertikalen Druckkraft.

Zunächst sollen jedoch einige geometrische Zwischenergebnisse betrachtet werden. Im oberen Diagramm der Abbildung 5.2 wird die Anzahl der durch die Kollisionserkennung ermittelten Schnittlinien zwischen den beiden Oberflächen dargestellt. Die Stufenform ergibt sich aus der hohen Regelmäßigkeit der Anordnung: In bestimmten Positionen tauchen neue Polygonreihen des Rads in den Boden ein.

Das mittlere Diagramm zeigt die Anzahl der aktiven Kontaktelemente. Die Master–Slave–Methode liefert (mit der Radoberfläche als Master) die rote/dunkelgraue Kurve. Auch hier resultiert ein stufenweise ansteigender

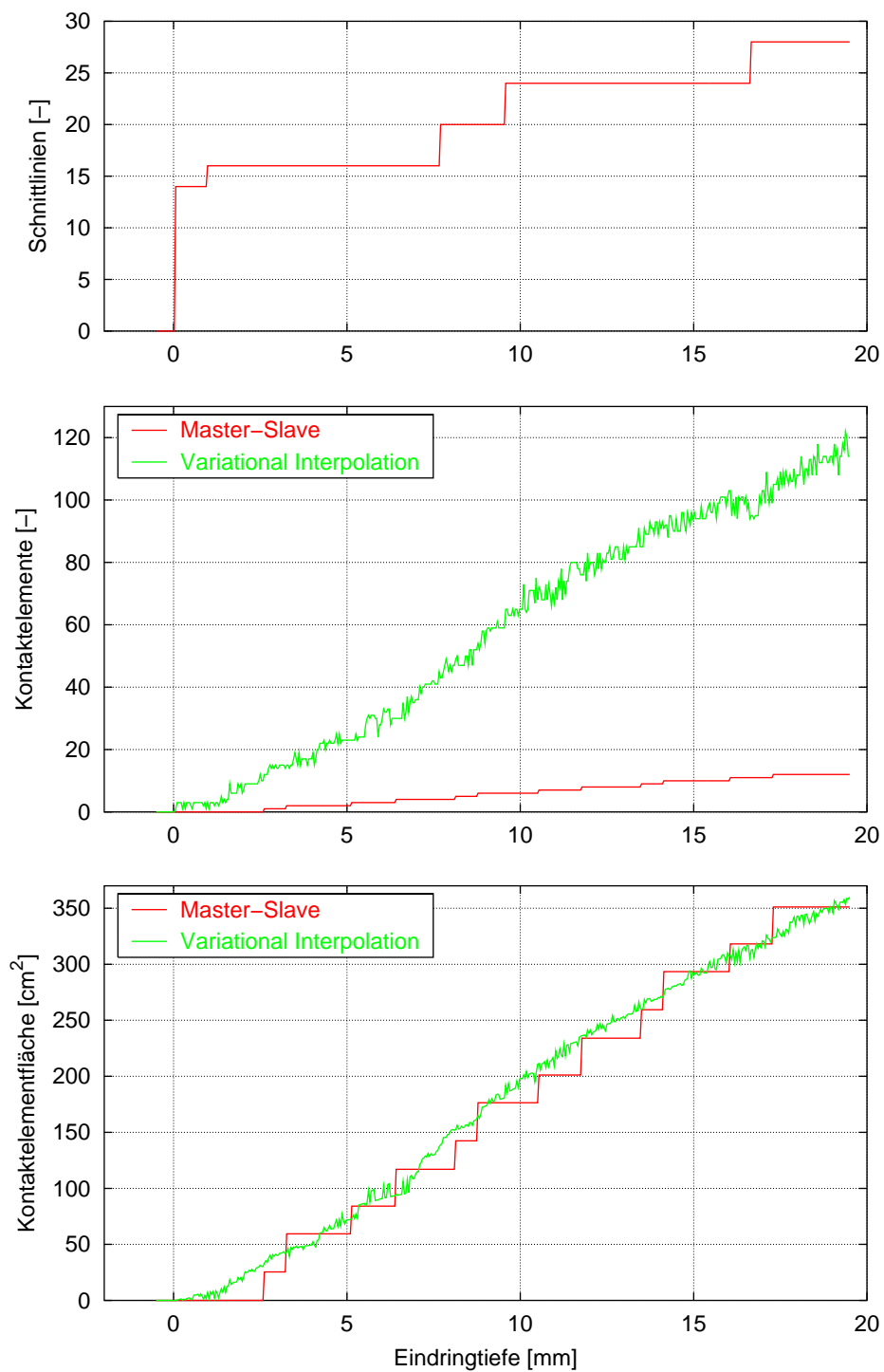


Abbildung 5.2: Anzahl der Schnittlinien, der Kontaktelemente und gesamte Kontaktelementfläche in Abhängigkeit der Eindringtiefe

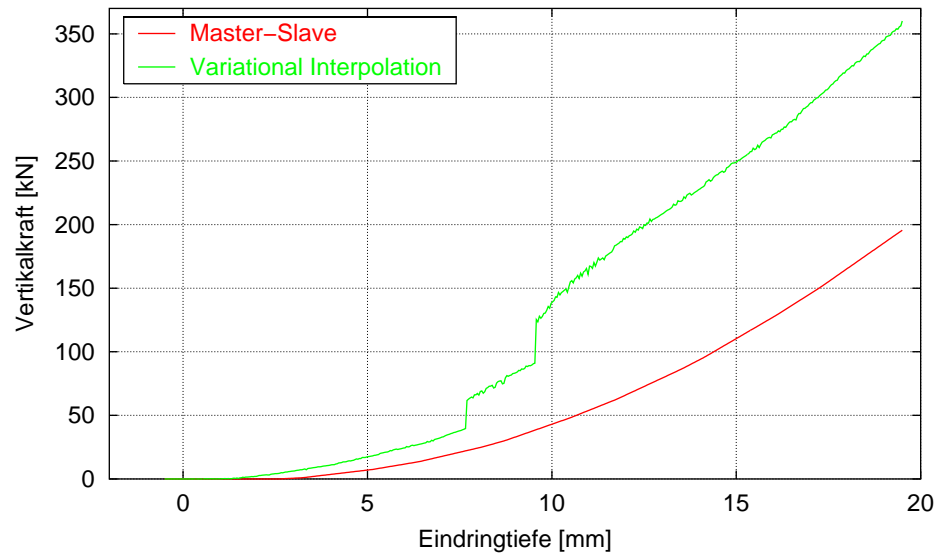


Abbildung 5.3: Vertikalkraft in Abhängigkeit der Eindringtiefe

Verlauf, da während der Absenkung immer mehr Dreiecksschwerpunkte unter die Bodenfläche gelangen. Deutlich instabiler verhält sich dagegen die grün/hellgrau dargestellte Variational-Interpolation-Methode, hier bei einer Referenzlänge der Triangulierung von $\delta_t = 30$ mm, die außerdem zu einer wesentlich feineren Diskretisierung führt. Das Rauschen lässt sich auf die in Abschnitt 4.3.2 erläuterte Sensitivität des Triangulierungsverfahrens gegenüber kleinsten Änderungen der Anfangsbedingungen zurückführen.

Für die gesamte Fläche aller Kontaktelemente (unteres Diagramm) ergibt sich ein annähernd linearer Zusammenhang. Dabei stimmen Master-Slave- und Variational-Interpolation-Methode unter Beibehaltung ihrer Charakteristika gut überein.

Abbildung 5.3 zeigt den Verlauf der vertikalen Druckkraft über der Eindringtiefe. Beide Verfahren zur Kontaktelementgenerierung liefern progressive Kraftgesetze, wobei allerdings die Variational-Interpolation-Methode deutlich höhere Werte ergibt und zudem überraschenderweise zwei ausgeprägte Sprünge aufweist.

Um diese Unstetigkeiten im Detail zu analysieren, wurden die beiden an den zweiten Sprung angrenzenden Zustände mit 9,54 mm und 9,58 mm Eindringtiefe untersucht. Abbildung 5.4 zeigt die zugehörigen approximierten Berührflächen. Die Ursache des plötzlichen Anstiegs der Kontaktkraft liegt offenbar in einer erheblichen Veränderung des Ergebnisses der Variational Interpolation, wenn die Ecken der Dreiecke am linken Rand der Lauffläche

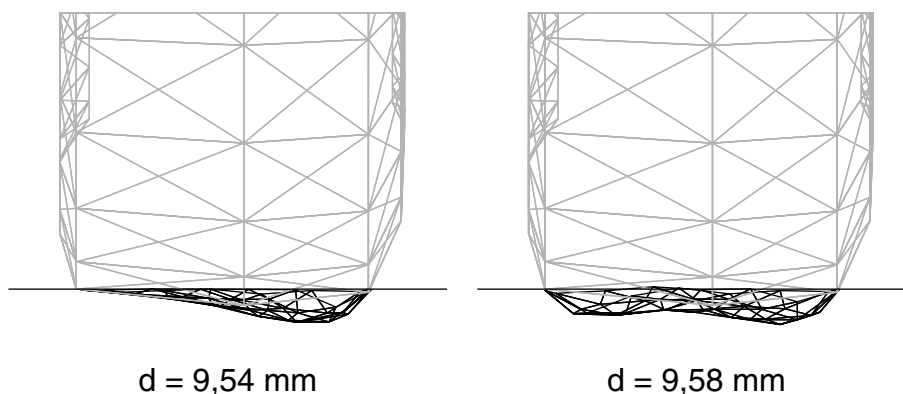


Abbildung 5.4: Überschwingen der durch Variational Interpolation approximierten Berührfläche

ins Schnittgebiet gelangen. Da die Polygonfläche des Rads auf der rechten Seite einen etwas größeren Durchmesser aufweist, entsteht zwischen 7,66 mm und 7,70 mm Eindringtiefe auf gleiche Weise der erste Sprung.

Wie im Abschnitt 4.3.2 gezeigt wurde, dienen die Eckpunkte der aktiven Flächenteile als Stützpunkte der Variational Interpolation. Sobald ein Dreieck auf der Außenseite des Rads aktiv wird, kommt daher ein steil nach oben führender Punkt hinzu. In Abbildung 5.4 ist erkennbar, dass die Berührfläche dadurch sogar deutlich aus dem Schnittvolumen heraus schwingt, was zu hohe Kräfte der Kontaktelemente zur Folge hat.

Die mit der Master–Slave–Methode ermittelte Weg–Kraft–Kurve verläuft dagegen ausgesprochen glatt und streng monoton, obwohl hier eine wesentlich gröbere Diskretisierung eingeht. Alle folgenden Ergebnisse wurden mit dem Master–Slave–Verfahren berechnet.

Die gezeigte Anordnung von Rad und Boden war Grundlage weiterer Simulationen, um die Implementierung des PCM prinzipiell zu testen. So wurde die Energieerhaltung beim wiederholten dämpfungsfreien Kontakt im Wechselspiel mit der Schwerkraft nachgewiesen und Fallversuche mit Dämpfung und Reibung simuliert, die ebenfalls plausible Ergebnisse lieferten.

5.2 Vergleich zum Hertzschen Modell

Im Rahmen dieser Arbeit wurde keine Verifikation des PCM anhand von Versuchsergebnissen durchgeführt. Um dennoch die Korrektheit des Verfahrens und seiner Implementierung nachzuweisen, wird im Folgenden ein Vergleich von numerischen und theoretischen Ergebnissen aufgezeigt. In Anlehnung

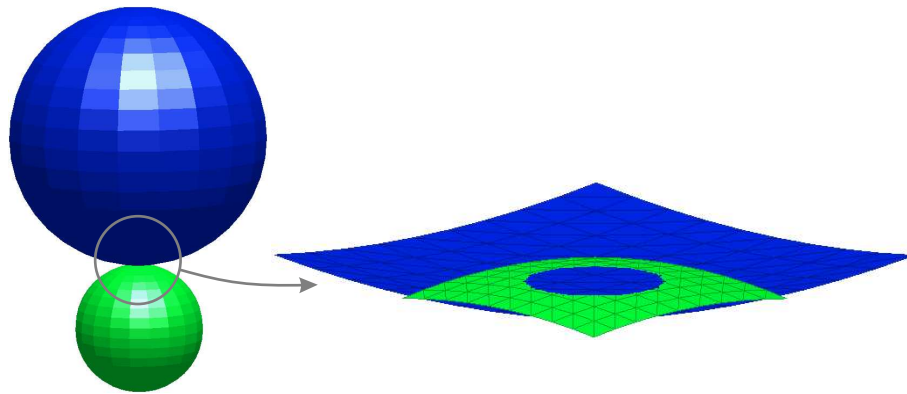


Abbildung 5.5: Modell zur Verifikation des PCM: Kontakt zweier Metallkugeln

an [17] dienen dabei das Randschichtmodell (s. Abschnitt 3.4.2) sowie das Hertzsche Modell als Referenz.

Als Experiment wird der statische, reibungsfreie Kontakt zweier Kugeln betrachtet (Abbildung 5.5). Die erste Kugel habe einen Radius von $R_1 = 20$ cm und weise mit $E_1 = 105 \cdot 10^9$ N/m² und $\nu_1 = 0,25$ die elastischen Materialeigenschaften von Grauguss auf. Das Gegenüber sei durch den halben Radius $R_2 = 10$ cm sowie $E_2 = 210 \cdot 10^9$ N/m² und $\nu_2 = 0,3$ für Stahl gekennzeichnet.

Für den Hertzschen Ansatz müssen die Oberflächen im Zentrum des Kontaktgebiets durch rotationssymmetrische Paraboloiden zweiter Ordnung approximiert werden. Daraus ergibt sich mit dem effektiven Radius

$$R = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}} \quad (5.1)$$

die Gap-Funktion

$$g(r) = \frac{1}{2R}r^2 - d + u_1(r) + u_2(r) \geq 0 \quad (5.2)$$

mit der Eindringtiefe (der unverformten Profile) d der Kugeln und ihren lokalen Verformungen u_1 und u_2 im radialen Abstand r von der gemeinsamen Rotationsachse.

Das Hertzsche Modell [53] liefert einen von den Materialkenngrößen unabhängigen Berührkreisradius

$$\rho_H = \sqrt{Rd} \quad (5.3)$$

mit der Fläche

$$A_H = \pi R d \quad (5.4)$$

und eine elliptische Druckverteilung

$$p(r) = \frac{2E^*d}{\pi\rho_H} \sqrt{1 - \left(\frac{r}{\rho_H}\right)^2} \quad (5.5)$$

mit dem kombinierten Elastizitätsmodul

$$E^* = \frac{1}{\frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}} . \quad (5.6)$$

Als Kraftgesetz ergibt sich

$$F_H(d) = \frac{4}{3} E^* \sqrt{R} \cdot d^{\frac{3}{2}} \quad (5.7)$$

mit dem Exponenten 1,5.

Um der Referenz möglichst nahe zu kommen, wird bei der folgenden Betrachtung des Problems mit dem Randschichtmodell von den gleichen parabolischen Oberflächen ausgegangen. Die Überschneidung der unverformten Profile führt zum Berührkreisradius

$$\rho_R = \sqrt{2Rd} , \quad (5.8)$$

welcher im Vergleich zu (5.4) einer doppelt so großen Berührfläche

$$A_R = 2\pi R d \quad (5.9)$$

entspricht. Dieser große Unterschied ist allein auf die Vernachlässigung elastischer Verformung infolge von Scherspannungen zurückzuführen (vgl. Abbildung 3.5) und macht bereits deutlich, dass das Randschichtmodell nur bedingt mit dem Hertzsch Ansatz vergleichbar ist.

Für die Normalspannung im Berührgebiet liefert das Randschichtmodell die parabolische Verteilung

$$p(r) = c_l d \left(1 - \left(\frac{r}{\rho_R}\right)^2 \right) , \quad (5.10)$$

welche zum quadratischen Kraftgesetz

$$F_R(d) = \pi c_l R \cdot d^2 \quad (5.11)$$

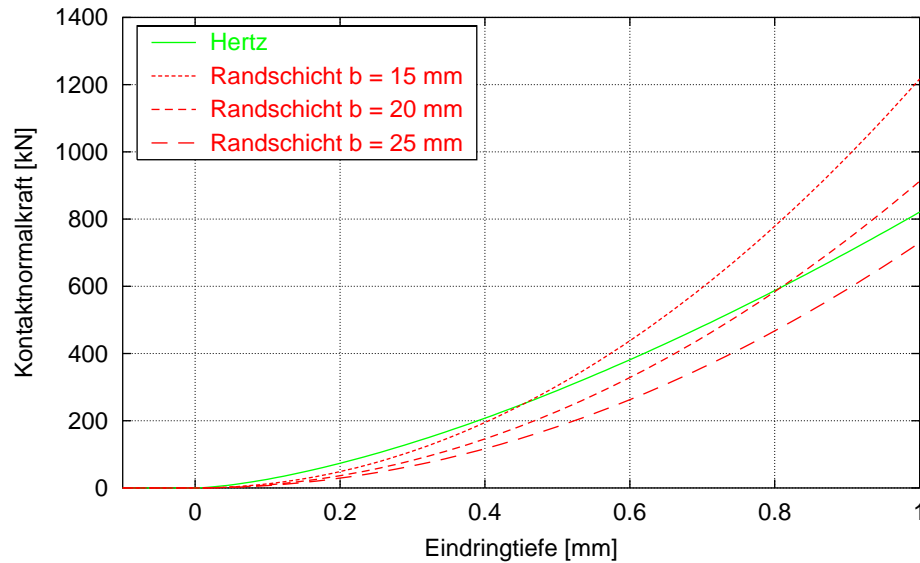


Abbildung 5.6: Vergleich der Kraftgesetze nach dem Hertzischen Modell und dem Randschichtmodell

zwischen Eindringtiefe und Normalkraft führt. Hierbei gehen in die kombinierte Randschichtsteifigkeit c_l nach (4.24) neben den Elastizitätsmodulen und Querkontraktionszahlen auch die Randschichtdicken ein. Folglich lässt sich die Kontaktsteifigkeit mit diesen zusätzlichen Parametern beliebig einstellen: Dünnere Randschichten führen zu höheren Werten.

Hier wirkt sich das vom Hertzischen Ansatz wesentlich verschiedene physikalische Modell der elastischen Randschichten auf ansonsten starren Körpern aus. Dennoch wird das Randschichtmodell in der Praxis erfolgreich als Approximation für den Kontakt zwischen Körpern angewendet, die eigentlich der Hertzischen Vorstellung von homogenem, linear-elastischem Material entsprechen, beispielsweise für den Rad-Schiene-Kontakt [17]. Dazu wird c_l nicht nach (4.24) berechnet, sondern durch Versuche oder Analogien bestimmt.

Wegen der unterschiedlichen Exponenten in (5.7) und (5.11) lässt sich allerdings keine Randschichtsteifigkeit c_l finden, für die das Randschichtmodell das gleiche Kraftgesetz liefert wie das Hertzische Modell. Um einen passenden Wert für den Kontakt der beiden Kugeln zu ermitteln, wurden deshalb die Kennlinien der Kontaktkraft für Eindringtiefen bis 1,0 mm verglichen. In Abbildung 5.6 ist ersichtlich, dass Randschichtdicken von 20 mm (auf beiden Kugeln) eine recht gute Näherung ergeben. Daher wurde dieser Wert für die numerische Simulation verwendet.

Zur Anwendung des PCM wurden die Kugeln im Berührungsbereich durch Polygonflächen approximiert, deren Stützpunkte auf einem äquidistanten

Gitter liegen (Abbildung 5.5). In Kontaktnormalenrichtung erfüllen sie die (als erster Term in (5.2) auftretende) Paraboloidgleichung, welche auch bei den theoretischen Untersuchungen Verwendung fand.

Um den Einfluss der Oberflächendiskretisierung zu veranschaulichen, wurde die Simulation mit drei verschiedenen feinen Gittern durchgeführt: 5×5 steht mit 25 Stützpunkten (32 Dreiecke) für eine sehr grobe Teilung, 10×10 entspricht 100 und 20×20 400 Punkten (162 bzw. 722 Dreiecke). Da jeweils Flächen gleicher Auflösung aufeinander trafen, fiel die Wahl des Verfahrens zur Kontaktelementgenerierung auf die Master–Slave–Methode im 2–Pass–Modus, sodass sich Ungenauigkeiten infolge unsymmetrischer Anordnung weitgehend aufheben (s. Abschnitt 4.3.1).

Wie im Abschnitt 5.1 wurden viele (hier 220) einzelne statische Simulationen ausgeführt, bei denen die vertikale Position der oberen Kugel in gleichen Schritten variiert wurde, sodass sich Eindringtiefen im Bereich $d = -0,1 \dots 1,0$ mm ergaben. Abbildung 5.7 zeigt die Gesamtfläche der Kontaktelemente sowie die Kontaktnormalkraft dieser Rechnungen im Vergleich zu den theoretischen Ergebnissen des Hertzschens Modells und des Randschichtmodells.

Für das Hertzsche Modell und das Randschichtmodell zeigt das obere Diagramm die Kontaktfläche nach (5.4) bzw. (5.9). Wie zu erwarten konvergiert die numerische Simulation mit feinerer Diskretisierung gegen das theoretische Ergebnis des Randschichtmodells. Mit den gröber geteilten Flächen erhält man einen deutlich gestuften Funktionsverlauf, der fast ständig unterhalb der Referenzlösung liegt. Diese Charakteristik wird dadurch verursacht, dass Polygone erst dann zur Kontaktfläche gezählt werden, wenn ihr Schwerpunkt im Kontaktgebiet liegt, sodass sie einen Beitrag zur Kontaktkraft leisten.

Ein ähnliches Verhalten zeigt die Kontaktnormalkraft, wobei bei der Diskretisierung 5×5 Knicke statt Stufen zu beobachten sind. Auch hier bewirken Diskretisierungsfehler stets niedrigere Werte, da die Annahme konstanter Druckspannung bei der Berechnung der Normalkraft eines Kontaktelements nach (4.27) signifikant verletzt wird.

Insgesamt verdeutlicht dieses Beispiel, dass das PCM bei geeigneter Diskretisierung gegen das Randschichtmodell konvergiert. Im Vergleich zum in der Technik häufig angewendeten Hertzschens Modell ergibt sich allerdings ein wesentlicher Nachteil: Die Annahme starrer, mit einer dünnen, elastischen Randschicht überzogener Körper entspricht nur selten den realen Gegebenheiten, sodass die Randschichtsteifigkeit für eine brauchbare Approximation nicht direkt über bekannte Geometrie- und Materialparameter bestimmt werden kann. Andererseits birgt das PCM auch gegenüber dem klassischen Randschichtmodell den großen Vorteil, dass das Kontaktgebiet beliebig komplex geformt und berandet sein darf. Unrealistische Ergebnisse sind lediglich bei scharfen Kanten und Spitzen zu erwarten.

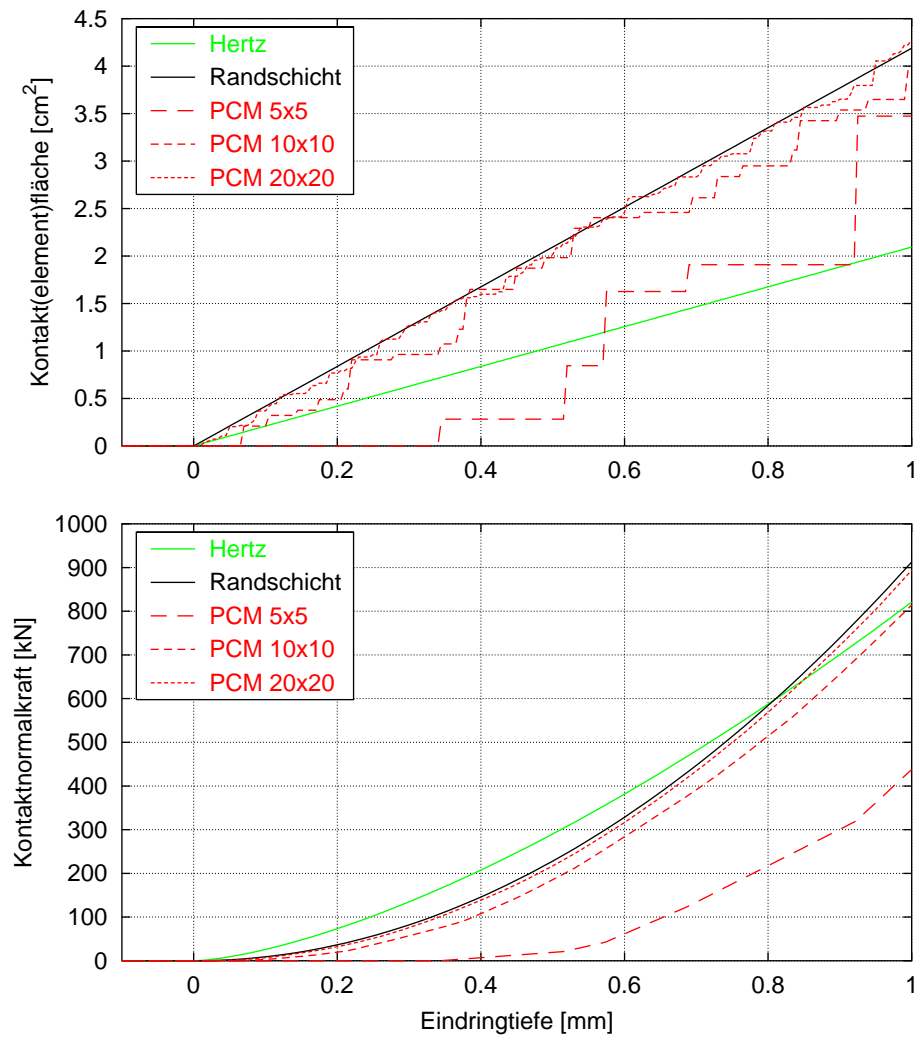


Abbildung 5.7: Berührfläche und Kontaktkraft der Kugeln nach dem Hertzchen Modell, dem Randschichtmodell sowie der numerischen PCM-Simulation mit verschiedenen feinen Diskretisierungen

5.3 Bouncing Bubbles

Die in diesem Abschnitt beschriebene Simulation dient als erstes Beispiel für die Simulation im Zeitbereich. Dabei soll die Leistungsfähigkeit des PCM demonstriert werden. Außerdem wird ausführlich auf für die Anwendung bedeutsame Zusammenhänge eingegangen.

Modellbeschreibung

Abbildung 5.8 zeigt das *Bouncing Bubbles* benannte Modell im Ausgangszustand: Drei identische rotationssymmetrische Körper in Ruhe werden der Erdgravitation mit $g = -9,81 \text{ m/s}^2$ ausgesetzt. Ihr Fall wird durch einen inertial festen „Trichter“ in Form einer Pyramide mit quadratischem Grundriss abgefangen. Im Weiteren sorgen die Kontaktkräfte der Bubbles untereinander sowie zum Trichter für ein dynamisches Wechselspiel, welches allmählich zum Erliegen kommt, da ihre potentielle Anfangsenergie durch Kontaktdämpfung und -reibung dissipiert (Animation s. [74]).

Ein Quadrat mit 2 m Seitenlänge bildet die obere Öffnung des 50 cm hohen Trichters. Die zugehörige Polygonfläche besteht aus nur vier Dreiecken. Die durch jeweils 950 Dreiecke dargestellten Bubbles sind 40 cm lang und haben ein hantelförmiges Profil, dessen größte Durchmesser von 20 cm

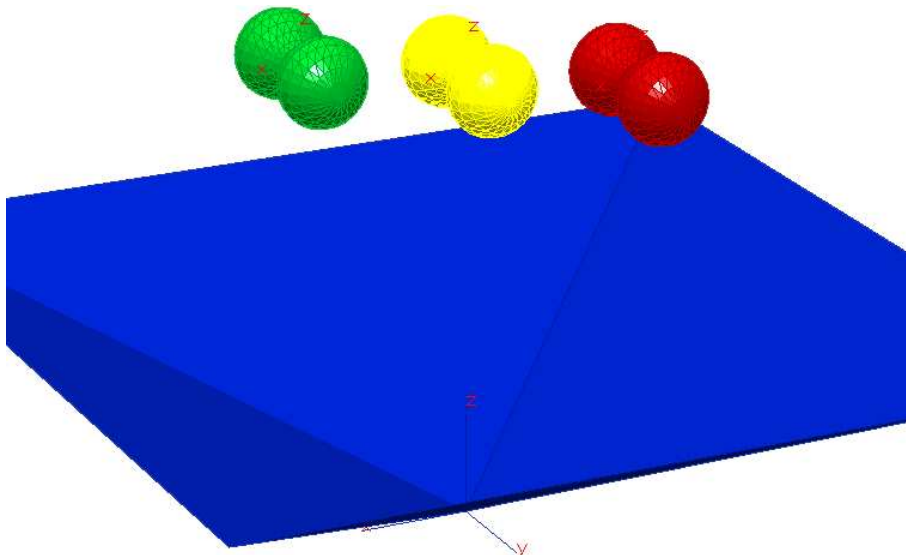


Abbildung 5.8: Anwendungsbeispiel für die Simulation mit dem PCM im Zeitbereich: Bouncing Bubbles

sich in der Mitte auf 10 cm verjüngen. Diese Form wurde gewählt, um konkave Oberflächenbereiche einzubeziehen und mehrfache sowie schmiegende Berührungen hervorzurufen, welche mit eindimensionalen Kontaktmodellen nicht analysiert werden können (s. Abschnitt 3.5.3).

Mit einer Masse von 2 kg und Trägheitsmomenten von $0,01 \text{ kgm}^2$ um die Längsachse und $0,02 \text{ kgm}^2$ um die Querachsen wird eine relativ kleine Dichte für die Bubbles angenommen. Zu Beginn der Simulation befindet sich ihr Mittel- und Schwerpunkt 100 cm über der Spitze des Trichters. Die beiden äußeren Bubbles 2 (rot/dunkelgrau) und 3 (grün/hellgrau) sind außerdem um -30 cm bzw. 30 cm in Querrichtung sowie 20 cm bzw. -20 cm in Längsrichtung verschoben (Abbildung 5.8).

Bei den Bouncing Bubbles treten insgesamt sechs Kontaktpaarungen auf, drei zwischen den Bubbles und dem Trichter sowie die drei weiteren möglichen Kombinationen der Bubbles untereinander. Dabei wird für die Bubbles eine 1 cm dicke elastische Randschicht mit $E = 5500 \text{ N/m}^2$ und $\nu = 0,4$ vorgegeben.

Der Trichter wird starr angenommen, sodass als Randschichtsteifigkeit c_l für die Kontakte zu den Bubbles direkt der Wert der Bubbles eingeht. Weiter sind diese Kontaktpaarungen durch die flächenbezogene Dämpfung $d_l = 500 \text{ Ns/m}^3$ sowie die Gleitreibzahl $\mu = 0,25$ bestimmt. Da nur in den Schwerpunkten der als Master modellierten Polygonflächen Kontaktelemente wirken (s. Abschnitt 4.3.1), können hier nur die Bubbles als Master definiert werden; anderenfalls würden sie wegen der extrem groben Diskretisierung des Trichters zwischen seinen Kontaktelementen hindurch fallen.

Für die Kontakte zwischen den Bubbles ergibt sich die kombinierte Randschichtsteifigkeit nach (4.24), wobei die elastischen Parameter ihrer Randschicht auf beiden Seiten eingehen. Hier wird mit der Dämpfung $d_l = 1000 \text{ Ns/m}^3$ und ebenfalls $\mu = 0,25$ gearbeitet. Da die beteiligten Flächen gleich fein diskretisiert sind, kommt das 2-Pass-Verfahren zum Einsatz, welches Approximationsfehler durch den unsymmetrischen Master-Slave-Ansatz weitgehend ausgleichen kann (s. Abschnitt 4.3.1).

Der in (4.28) eingehende Verfahrensparameter zur Einblendung der Dämpfungskraft wird für alle Kontakte mit $u_d = 1 \text{ mm}$ angesetzt. Ebenso wird überall $v_\varepsilon = 2,0 \text{ mm/s}$ als Grenzggeschwindigkeit zur Regularisierung des Reibgesetzes nach (4.34) benutzt. Ähnlich wie Parameter zur Konfiguration numerischer Lösungsverfahren sind u_d und v_ε aus Sicht des Anwenders schwierig zu handhabende Größen, weil sie im realen System nicht in dieser Form vorkommen. Deshalb kann hier nur auf Erfahrungswerte oder Verifikation durch Messungen zurückgegriffen werden.

Ergebnisse

Alle im Folgenden dargestellten Ergebnisse wurden mit dem im Abschnitt 2.3 beschriebenen Integrationsverfahren DASSL, welches in SIMPACK auf die Gear–Gupta–Leimkuhler–Formulierung [41] der MKS–Bewegungsgleichungen angewendet wird, berechnet. Als absolute und relative Toleranzen wurden $atol = rtol = 2,0 \cdot 10^{-4}$ eingestellt. Damit die Schrittweiten während des freien Falls am Anfang nicht zu groß werden, wurden sie auf $h_{max} = 10$ ms beschränkt; sonst kann die numerische Integration beim Aufprall der Bubbles ineffizient und sogar instabil verlaufen.

Bevor auf die Ergebnisse im Einzelnen eingegangen wird, sei auf die hohe Sensitivität des Systems hingewiesen. Ein solches Experiment, prinzipiell verwandt mit der Ermittlung der Lottozahlen, wird in der Realität stets zu unterschiedlichen Bewegungsabläufen führen, weil sich kleinste Störungen wie Luftströmungen, Temperaturschwankungen oder mikroskopische Unterschiede im Anfangszustand mit jedem der zahlreichen hochdynamischen Kontaktvorgänge stärker auswirken. Diese chaotische Charakteristik spiegelt sich in der Simulation wieder: Die Verwendung unterschiedlicher Integrationsverfahren oder -parameter sowie selbst minimale Abweichungen in den physikalischen Systemgrößen führen zu anfangs meist kaum merklichen, im Verlauf der Bewegung jedoch immer signifikanteren Unterschieden. Somit ergaben sich am Ende stets verschiedene Ruhelagen der Bubbles.

Trotz dieser hohen Empfindlichkeit erlaubt die Bouncing Bubbles Simulation schon alleine durch Betrachtung der Animation eine Beurteilung des PCM, denn die menschliche Bewegungsvorstellung vermag unrealistische Ergebnisse derartiger Experimente erstaunlich gut zu erkennen. So erschien das Systemverhalten bei den ersten Versuchen augenscheinlich fehlerhaft, weil Massen und Trägheitsmomente der Bubbles in unrealistischem Verhältnis zueinander standen. Auch anfängliche Implementierungsfehler in der Berechnung der Dämpfungs- und Reibungskräfte waren deutlich in der Animation erkennbar.

Abbildung 5.9 zeigt die Anzahl der Schnitte (s. Abschnitt 4.2.2) aller sechs Kontaktpaarungen über der Zeit. Anhand dieser Diagramme lässt sich das Geschehen während der ersten Sekunde recht gut nachvollziehen: Zuerst prallen die beiden äußeren Bubbles 2 und 3, wenig später das mittlere mit jeweils zwei Berührgebieten auf den Trichter. Nach kurzem freien Fall sind Kontakte der äußeren mit dem mittleren Bubble sowie zwischen 2 und 3 zu erkennen. Noch bevor 1 wieder auf dem Trichter auftritt, berühren sich 2 und 3 ein weiteres Mal, kurzzeitig sogar zweifach.

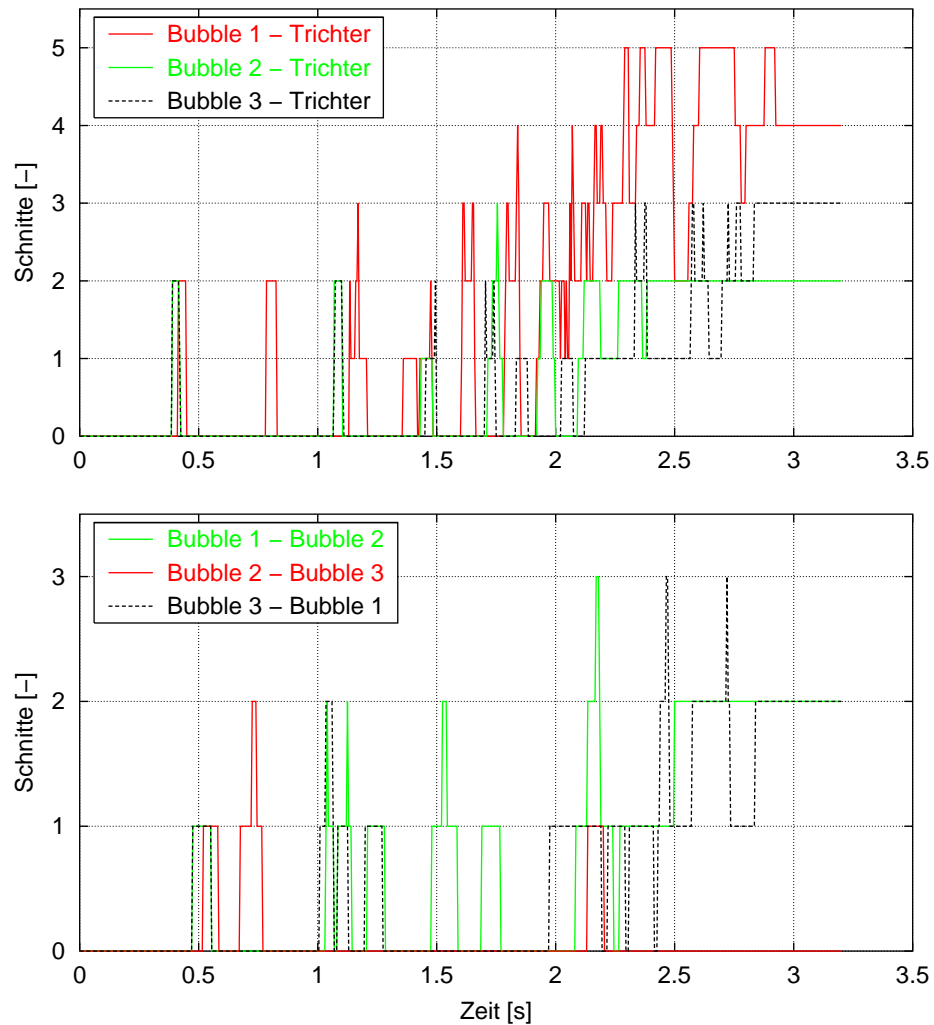


Abbildung 5.9: Anzahl der Schnitte zwischen Bubbles und Trichter (oben) sowie der Bubbles untereinander

Diese ersten acht Kontakte mit insgesamt 13 Berührungen laufen, wie theoretisch zu erwarten, symmetrisch zur mittleren vertikalen Achse des Trichters ab. Doch im Weiteren stellen sich numerisch bedingte, bald dominierende Abweichungen ein.

In der Endlage sind vier Schnitte zwischen dem Trichter und Bubble 1 zu beobachten (Abbildung 5.10). Dabei liegt es mit einem Ende wie eine Kugel in der Mitte des Trichters (zwei Schnitte) und mit dem anderen an einer seiner Kanten auf. Auch bei den Kontakten der Bubbles untereinander kommen

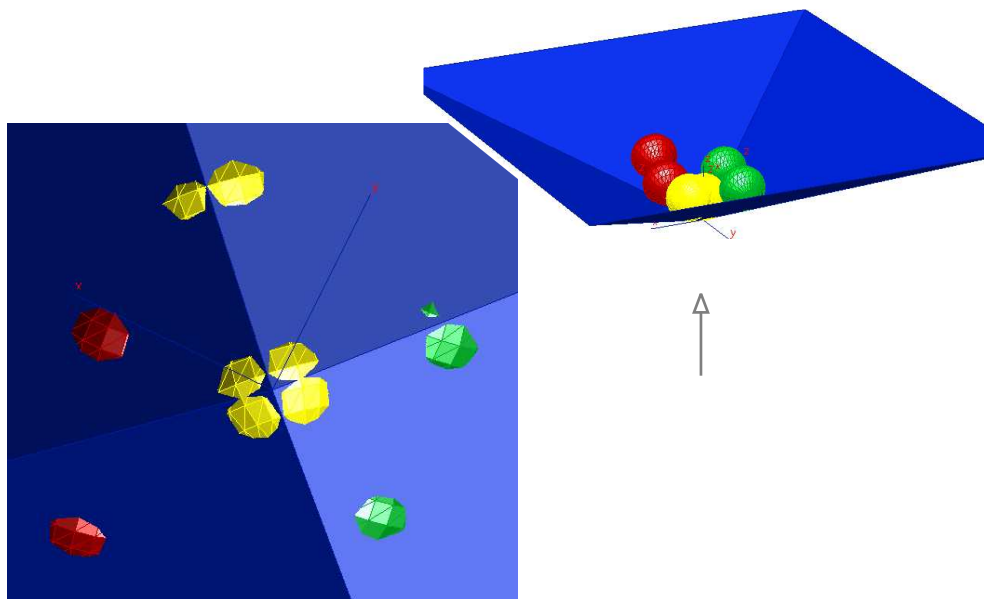


Abbildung 5.10: Endlage der Bouncing Bubbles Simulation: Die Schnitte mit dem Trichter sind von unten gesehen gut erkennbar

bis zu drei Schnitte gleichzeitig vor. Hier wird deutlich, dass dieser Vorgang die Kontaktpunktbestimmung des klassischen eindimensionalen Ansatzes (s. Abschnitt 3.5.3) bei Weitem überfordern würde.

In Abbildung 5.11 wird die Anzahl der BV-Kollisionsprüfungen dargestellt. Zwischen den Bubbles ist ein eindeutiger Zusammenhang mit den Schnitten erkennbar. Dagegen ergibt sich bei den Kontakten zum Trichter überwiegend ein Sättigungswert von 7585, was der Gesamtzahl aller Prüfungen zwischen den sieben BVs des Trichters und den 1899 BVs eines Bubbles entspricht. Sehr grob diskretisierte Flächen können den Rechenaufwand also erheblich erhöhen, weil die zugehörige BV-Hierarchie entsprechend flach wird und ihre Aufgabe, möglichst viele Polygonpaarungen von der elementaren Analyse auszuschließen, nur noch ungenügend erfüllt. Auf diese Problematik wird am Ende des Abschnitts näher eingegangen.

Abbildung 5.12 zeigt die Anzahl der elementaren Analysen, d. h. Schnitttests von Dreiecken bei der Kollisionserkennung der jeweiligen Flächenpaarung. Wie zu erwarten ähneln die Verläufe denen der BV-Tests (Abbildung 5.11). Die absoluten Zahlen sind jedoch deutlich kleiner, weil die elementare Analyse nur für die sich schneidenden Blätter der BV-Hierarchie durchgeführt wird (s. Abschnitt 4.1.3).

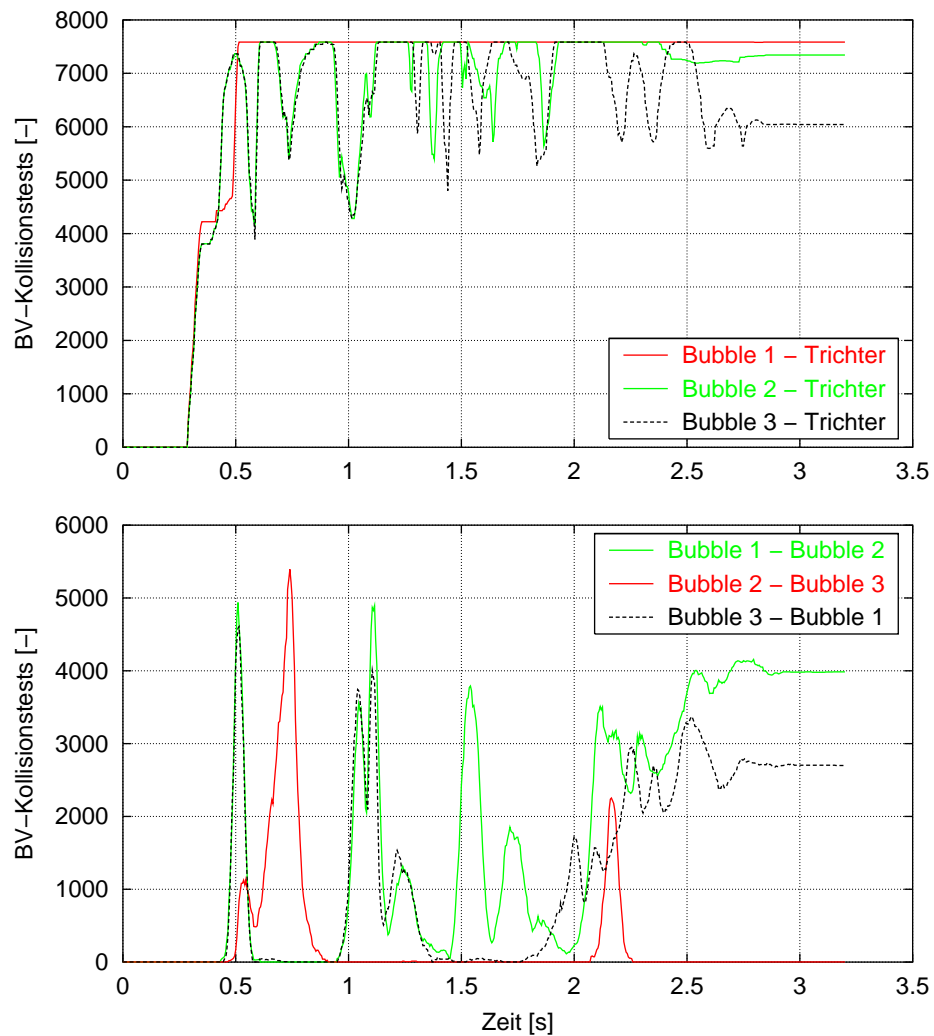


Abbildung 5.11: Anzahl der Kollisionstests zwischen den Hüllvolumina der Bouncing Bubbles Kontakte

Ebenfalls qualitativ ähnlich, aber quantitativ noch einmal erheblich kleiner, ist die Anzahl der gefundenen Schnittlinien zwischen den Bubbles (Abbildung 5.13 unten). Bei den Kontakten zwischen den Bubbles und dem Trichter (oben) wird hingegen deutlich, dass die flache BV-Hierarchie des Trichters bei fast jeder Auswertung viele tausend vergleichsweise rechenaufwändige, ergebnislose Schnittpunkte verursacht, denn nur ein kleiner Teil der in Abbildung 5.11 angegebenen BV-Kollisionsprüfungen liefert schneidende Polygonpaare.

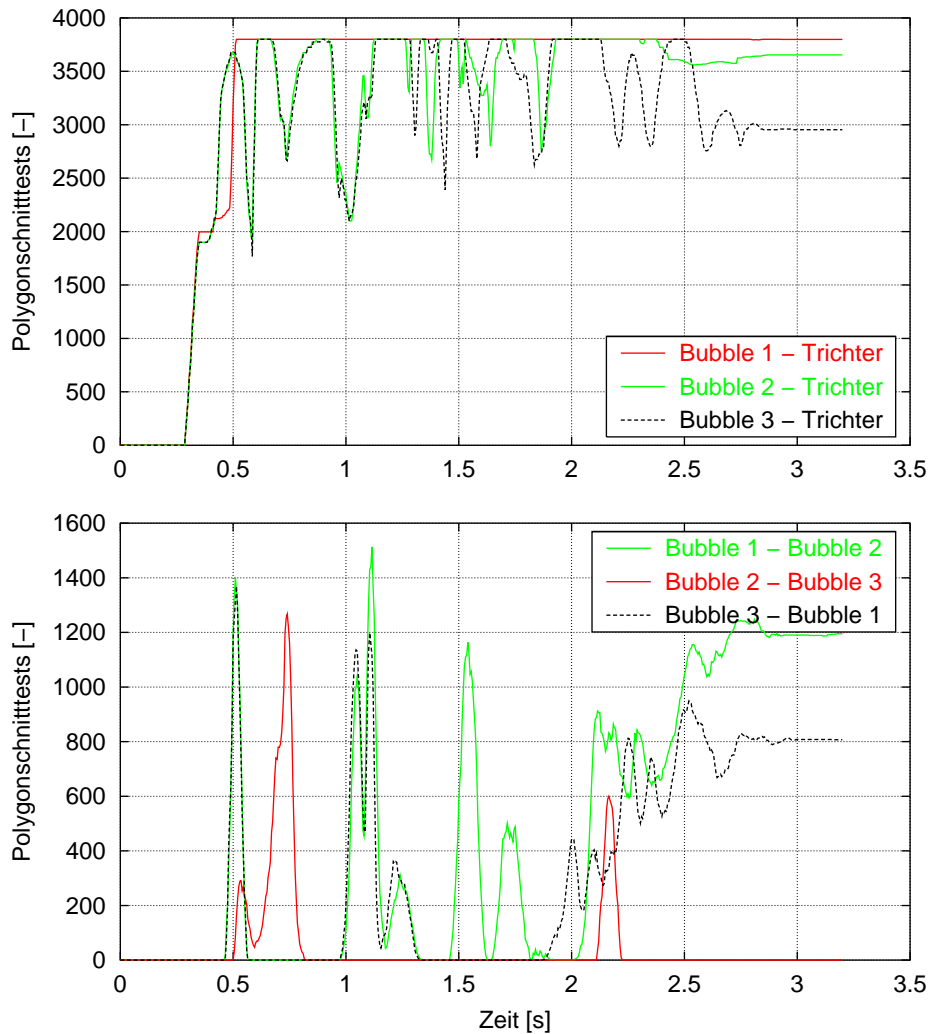


Abbildung 5.12: Anzahl der Schnitttests zwischen Dreiecken der Bouncing Bubbles Kontaktflächenpaarungen

Die Anzahl der Schnittpolygonzüge wird hier nicht dargestellt, weil sie mit der Anzahl der Schnitte (Abbildung 5.9) nahezu identisch ist. Nur an einigen wenigen Stellen sind mehr Schnittpolygone als Schnitte bei Berührungen zwischen den Bubbles zu beobachten, was auf mehrfach berandete Kontaktgebiete hinweist. Diese können bei schmiegender Berührung vorkommen, wenn etwa infolge der Polygonrepräsentation der Oberflächen ein einzelner Eckpunkt aus dem Schnittvolumen herausragt.

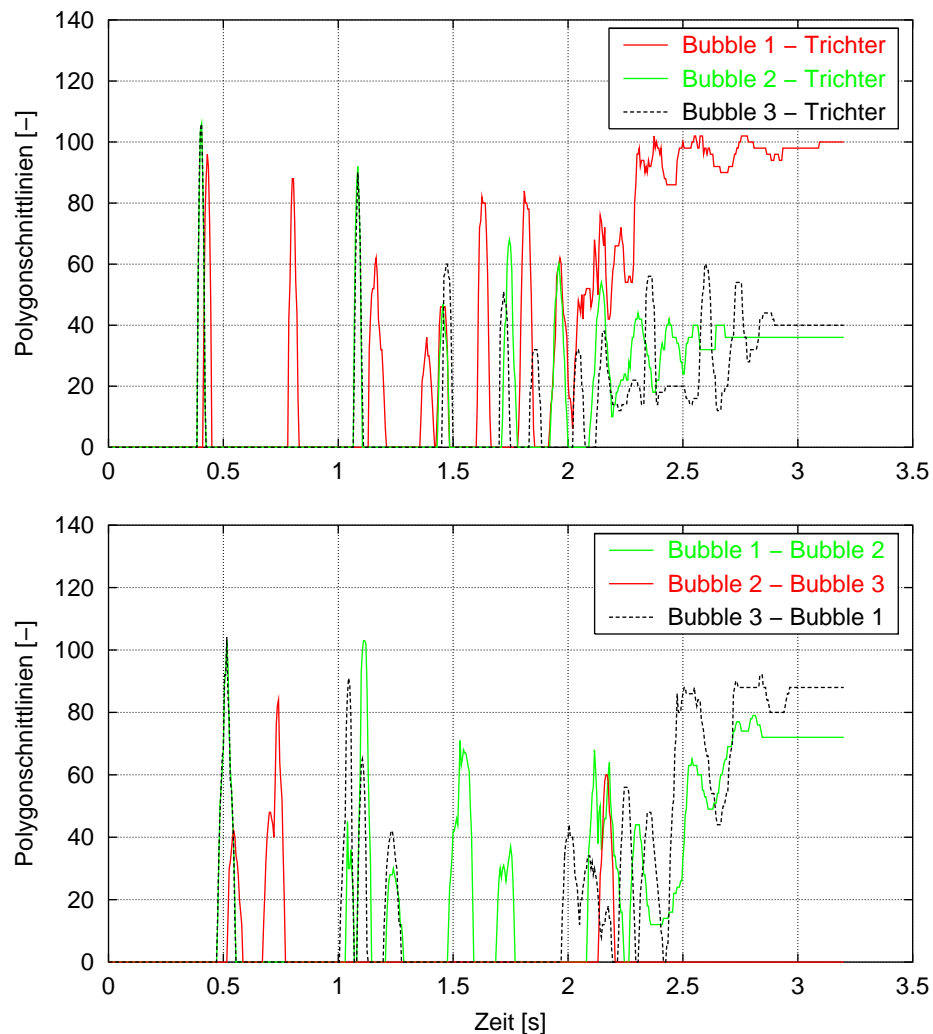


Abbildung 5.13: Anzahl der gefundenen Schnittlinien zwischen Dreiecken der Bouncing Bubbles Kontaktflächenpaarungen

Abbildung 5.14 zeigt die Anzahl der wirkenden Kontaktelemente. Selbst bei den großen Eindringtiefen während der ersten Kontakte zwischen den Bubbles und dem Trichter sind maximal 171 Elemente aktiv. In der gleichen Größenordnung liegt die Anzahl der Elemente zwischen den Bubbles, wobei hier einerseits eine annähernde Verdoppelung durch den 2-Pass-Modus enthalten ist, andererseits aber geringere Aufprallgeschwindigkeiten auftreten.

Da die Bubbles recht gleichmäßig diskretisiert sind, ähneln die in Abbildung 5.15 dargestellten Diagramme der Berührflächen denen der Anzahl der

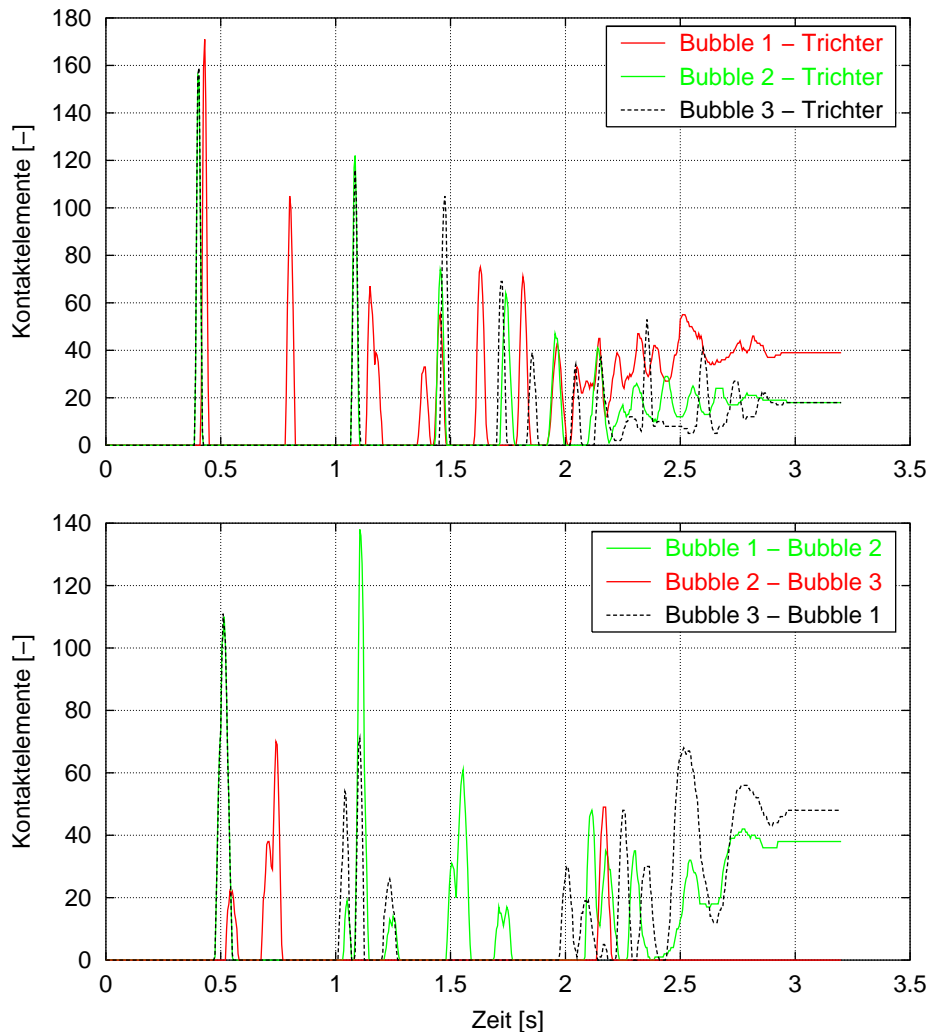


Abbildung 5.14: Anzahl der wirksamen Kontaktelemente bei der Bouncing Bubbles Simulation

Kontaktelemente. Interessant ist, dass hier bereits bei den ersten Kontakten zwischen den Bubbles leicht unsymmetrische Verläufe zu beobachten sind.

Wie im Abschnitt 4.4 gezeigt wurde, ergibt sich die Kontaktkraft des PCM als Summe aller wirksamen Kontaktelemente. Es fällt also eine sich i. Allg. ständig ändernde, ggf. sehr große Anzahl kinematischer Eingangs- und dynamischer Ausgangsgrößen an, deren vollständige Erfassung weder sinnvoll noch mit erträglichem Aufwand implementierbar ist. Um dennoch mehr Informationen als nur die resultierenden Kraft- und Momentenvektoren einer Kontaktpaarung bereitzustellen, liefert das PCM einige gewichtete

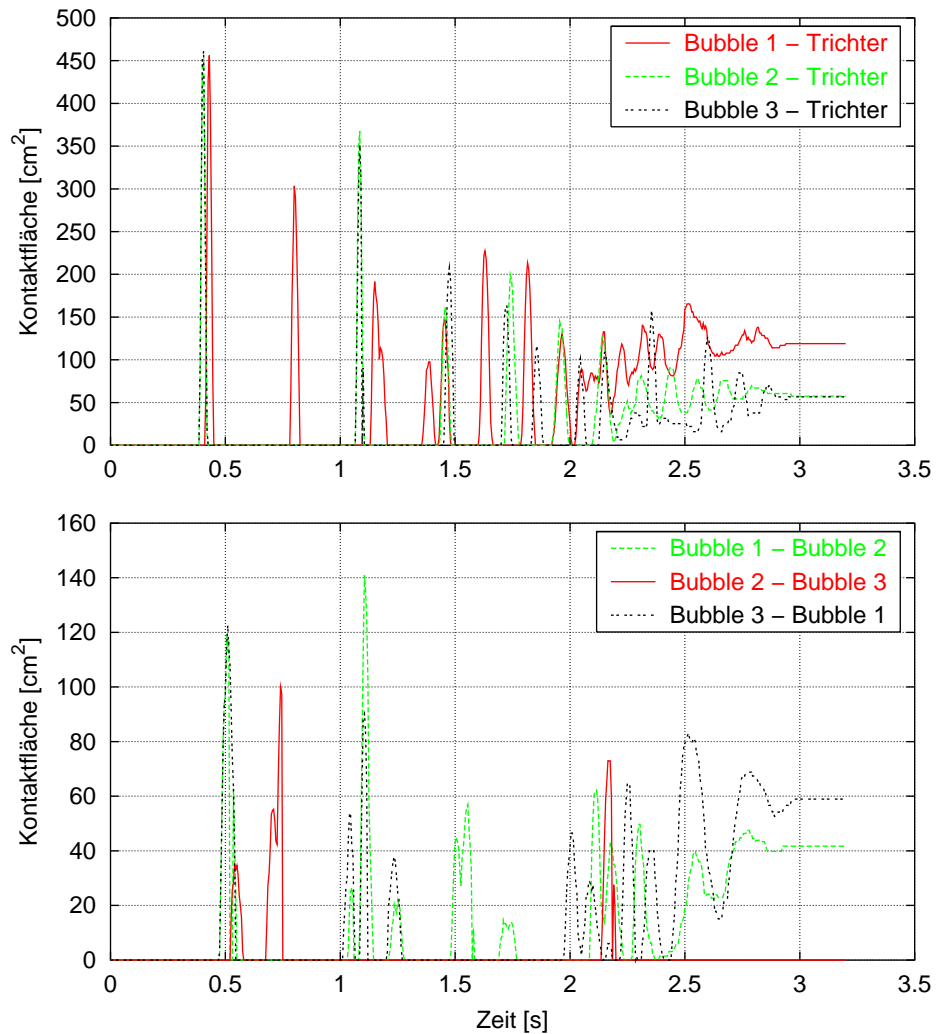


Abbildung 5.15: Im PCM entspricht die Kontaktfläche der Summe der Flächen der wirksamen Kontaktelemente

Werte sowie extremale Größen wie etwa die maximale Normalspannung aller Kontaktelemente.

So zeigt Abbildung 5.16 die anhand der Kontaktelementflächen gewichtete Eindringtiefe

$$\bar{u}_n = \frac{\sum_k u_{nk} A_k}{\sum_k A_k} , \quad (5.12)$$

welche in vielen Fällen ein brauchbares Maß für die gesamte Überschneidung der Kontaktflächen darstellt. Als weitere zusätzliche Ergebnisse gibt das PCM die maximale Eindringtiefe, die minimale, gewichtete und maximale

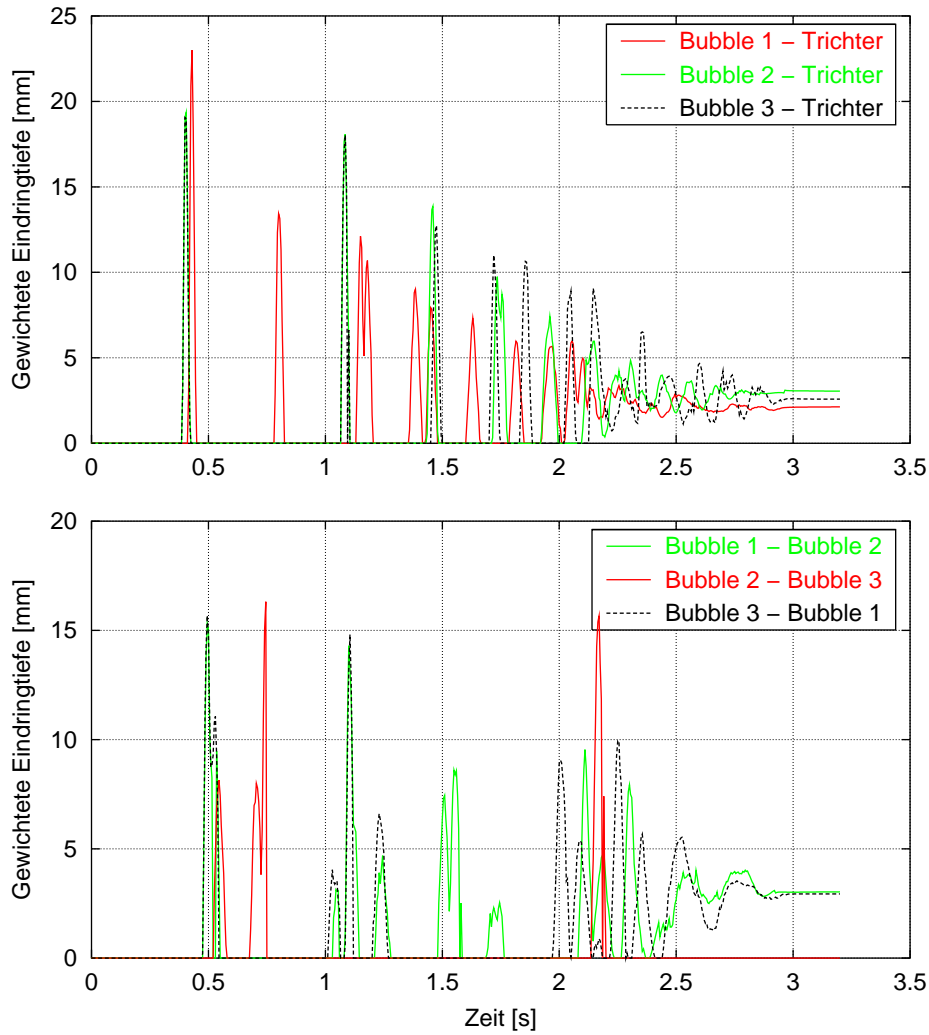


Abbildung 5.16: Gewichtete Eindringtiefe der Bouncing Bubbles nach (5.12)

Normal- und Tangentialgeschwindigkeit und die gewichtete und maximale Normal- und Tangentialspannung aus. Außerdem wird z. B. zur Visualisierung durch

$$\bar{\mathbf{r}}_{MeC} = \frac{\sum_k |\mathbf{F}_k| \mathbf{r}_{MeC_k}}{\sum_k |\mathbf{F}_k|} \quad (5.13)$$

sowie

$$\bar{\mathbf{n}} = \frac{\sum_k |\mathbf{F}_k| \mathbf{n}_k}{\sum_k |\mathbf{F}_k|} \quad (5.14)$$

eine durch den Kraftbetrag gewichtete Position sowie Wirkrichtung der gesamten Kontaktkraft berechnet.

Numerik und Effizienz

Die jeweils sechs Freiheitsgrade der Bubbles führen zu 36 Zustandsgrößen erster Ordnung. Da im Modell weder kinematische Schleifen oder sonstige algebraische Nebenbedingungen, noch extrem große Steifigkeiten bzw. kleine Massen enthalten sind, eignet sich prinzipiell jedes Integrationsverfahren zur Lösung der Bewegungsgleichungen.

Bei der Bouncing Bubbles Simulation kann der Rechenaufwand im Integrator und im MKS-Formalismus gegenüber der vergleichsweise aufwändigen Kontaktkraftberechnung durch das PCM vernachlässigt werden. Da die Kontaktsteifigkeiten und -dämpfungen niedrig sind und nicht zu einem steifen System führen, ist bestmögliche Effizienz für Verfahren zu erwarten, die ohne Jacobimatrixauswertungen (welche zahlreiche Funktionsaufrufe erfordern) auskommen (s. Abschnitt 2.3).

Tatsächlich erwies sich der Integrator LSODE [92] unter Verwendung der Adams-Bashforth-Moulton Mehrschrittmethode (variable Schrittweite und Ordnung) als schnellstes der in SIMPACK zur Verfügung stehenden Verfahren. Dies gilt jedoch nur für den dynamischen Anteil der Simulation bis ca. 3,0 s; im weiteren Verlauf der Rechnung brach die Schrittweite dramatisch ein.

Als Ursache für dieses Problem konnte das regularisierte Reibkraftgesetz (4.34) ausgemacht werden: Sobald die Bubbles in der Spitze des Trichters liegen bleiben, fallen die Tangentialgeschwindigkeiten der Kontakte infolge der Reibkräfte unter die Grenzgeschwindigkeit v_ε . Dann wirkt im Regularisierungsbereich faktisch ein Dämpferelement mit sehr hoher Dämpfung und es stellt sich eine Kriechbewegung ein. In diesem Zustand stellen sich entsprechend große (negative, reelle) Eigenwerte ein, sodass ein steifes System vorliegt, für das Integrationsmethoden ohne fehlerkontrollierte Korrekturiteration nicht geeignet sind. Mit Verringerung von v_ε verschärft sich dieses Problem dramatisch.

Während LSODE sowie explizite Verfahren am Ende der Simulation sehr ineffizient arbeiten, berechnen Methoden, die für steife Systeme geeignet sind, diesen Bereich besonders schnell. Obwohl alle Integratoren wegen der hohen Sensitivität des Systems unterschiedliche Ergebnisse liefern, dominiert dieser Effekt bereits bei Erhöhung der Simulationszeit von 3,0 auf 3,2 Sekunden grundsätzlich die erforderliche Rechenzeit.

Abbildung 5.17 zeigt Schrittweite und Ordnung der Simulationsrechnung mit DASSL. In den Freiflugphasen, aber auch während des Kriechbereichs, arbeitet das Verfahren mit der größten zugelassenen Schrittweite von 10 ms. Ansonsten bleiben die Schrittweiten stets größer als 0,1 ms, und die Ordnung liegt überwiegend zwischen zwei und vier. Hier wird deutlich, dass das PCM

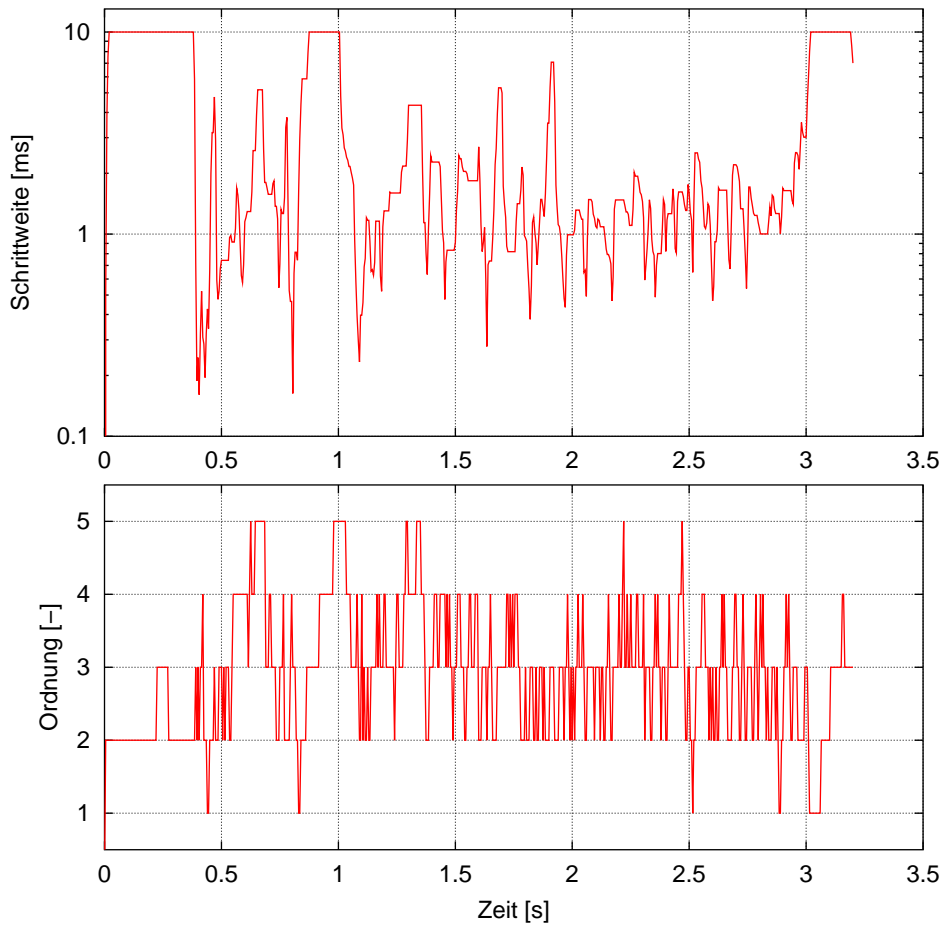


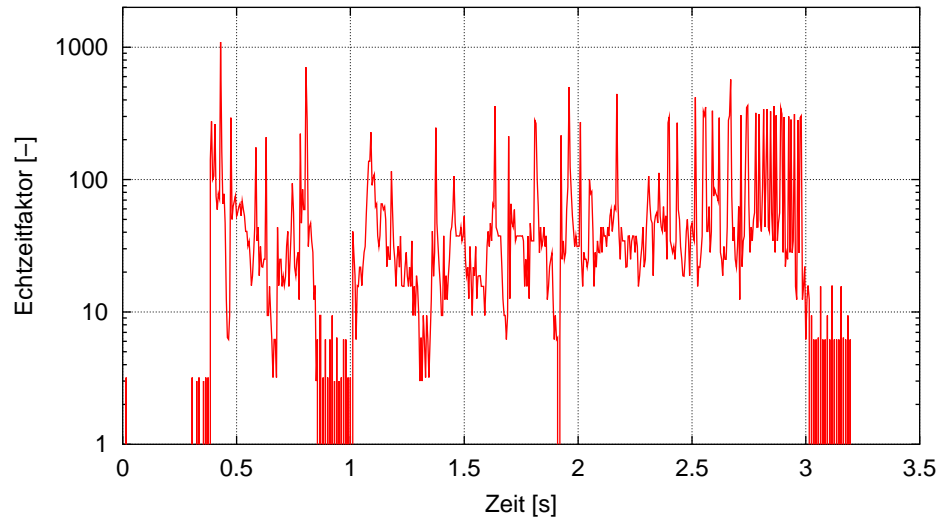
Abbildung 5.17: Schrittweite und Ordnung des Integrators DASSL bei der Bouncing Bubbles Simulation

numerisch ausgesprochen unproblematisch zu handhaben ist, denn komplexe oder gar inkonsistente Systeme erfordern erfahrungsgemäß erheblich kleinere Schritte.

Ein interessantes Maß für die Effizienz von Simulationen im Zeitbereich ist der Echtzeitfaktor τ , welcher als Quotient

$$\tau = \frac{\Delta T_{\text{Rechenzeit}}}{\Delta T_{\text{Echtzeit}}} \quad (5.15)$$

der für die Simulation benötigte Rechenzeit $\Delta T_{\text{Rechenzeit}}$ und der dabei berechneten Simulationszeit $\Delta T_{\text{Echtzeit}}$ berechnet wird. Angewendet auf die Ausgabeintervalle ergibt sich für die Bouncing Bubbles Simulation der in

Abbildung 5.18: Echtzeitfaktor¹ der Bouncing Bubbles Simulation

Simulationszeit	3,2 s
Mittlere Schrittweite	1,33 ms
Funktionsauswertungen	6099
Jacobimatrixauswertungen	67
CPU-Zeit ¹	166,1 s
Mittlerer Echtzeitfaktor ¹	51,9

Tabelle 5.1: Integrationsstatistik des Integrators DASSL bei der Bouncing Bubbles Simulation

Abbildung 5.18 dargestellte Verlauf. Die sehr hohen Spitzen bis in den Bereich von 1000 sind typisch für einen BDF-Integrator, der sich daher nicht für Echtzeitanwendungen eignet. Natürlich hängt τ auch stark von der verwendeten Hardware-Architektur ab.

In Tabelle 5.1 werden die wichtigsten statistischen Größen der gesamten Simulation aufgelistet. Angesichts der stark nichtlinearen Kinematik und des hochdynamischen, durch zahlreiche Kontaktvorgänge geprägten Bewegungsablaufs erscheint der numerische Aufwand akzeptabel.

¹PC, Win 2k, Xeon 2400 MHz Prozessor, 1024 MB RAM

Zusammenhang zwischen Modellierung und Effizienz

Wie bereits zuvor erwähnt, hängt die Effizienz des PCM signifikant von der Diskretisierung der verwendeten Polygonflächen ab. Dabei wird der Rechenaufwand nicht notwendigerweise kleiner, wenn weniger Polygone zu verarbeiten sind. Die Bouncing Bubbles Simulation kann in diesem Zusammenhang als Beispiel dienen, da die Trichterfläche durch nur vier große Dreiecke beschrieben wird.

Die aus nur sieben Elementen bestehende zugehörige BV-Hierarchie ist besonders flach: Das Wurzelement umschließt den gesamten Trichter, die in der übernächsten Ebene liegenden Blätter jeweils eines der Dreiecke. Somit umschließen die kleinsten Hüllquader wenigstens die Hälfte des Trichters. Sie können jedoch noch erheblich größer werden, wenn sich das Bubble der betrachteten Kontaktpaarung gegenüber dem Initialzustand verdreht. Denn wie in Abschnitt 4.1.3 (Abbildung 4.10) gezeigt wurde, werden die BVs der mit der Kennung F behandelten Fläche (hier Trichter) zur Vermeidung rechenaufwändiger Koordinatentransformationen durch umhüllende Quader ersetzt, deren Achsrichtungen durch die Fläche E (hier Bubble) bestimmt werden.

Im Fall der Bouncing Bubbles Simulation wirkt sich dieser Effekt besonders ungünstig aus: Sobald sich die Bubbles in der Spitze des Trichters befinden, müssen meist alle möglichen 7585 BV-Kollisionsprüfungen und 3800 elementaren Analysen ausgeführt werden (Abbildungen 5.11 und 5.12). Der BV-Algorithmus kann dann also keinen einzigen der aufwändigen Polygonschnitttests vermeiden, sondern erhöht vielmehr die Anzahl der Rechenoperationen erheblich.

Dieses Problem lässt sich durch geschickte Modellierung deutlich entschärfen. So ist es möglich, den Trichter als Fläche E und die Bubbles als F zu definieren, sodass die großen BVs des Trichters durch den BV-Algorithmus nicht weiter vergrößert werden. Damit wird die BV-Hierarchie wirksam und die Effizienz verbessert sich. Hierbei ist jedoch zu beachten, dass bei dieser Anordnung die Anzahl der erforderlichen Vektortransformationen von den fein diskretisierten Bubbles zum Bezugssystem E erheblich ansteigt und den Rechenzeitgewinn teilweise aufhebt.

Weiter besteht die Möglichkeit, die vier Dreiecke des Trichters zu unterteilen, sodass eine geometrisch äquivalente, feiner diskretisierte Fläche entsteht. So kann man erreichen, dass der Trichter über eine tiefere BV-Hierarchie verfügt, die den Rechenaufwand der Kollisionserkennung wesentlich reduzieren kann. Natürlich stellt sich auch hier ein gegenläufiger Effekt ein, wenn die Anzahl der zu verarbeitenden Polygone zu groß wird.

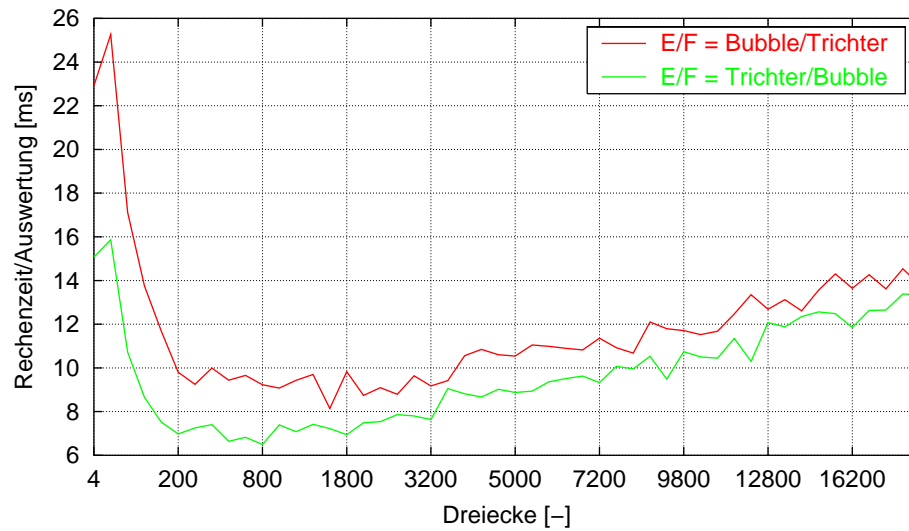


Abbildung 5.19: Mittlere Rechenzeit für die Funktionsauswertung der Bouncing Bubbles Simulation für verschiedene Diskretisierungen des Trichters und unterschiedliche Zuordnung als Fläche E und F

Um den Einfluss dieser Modellierungsdetails zu untersuchen, wurde die Bouncing Bubbles Simulation mit 50 verschiedenen Diskretisierungen des Trichters, jeweils für die beiden möglichen Zuordnungen von Trichter und Bubbles als Flächen E und F , berechnet. Obwohl die dabei erzielten Ergebnisse theoretisch identisch sein müssten, erhält man aufgrund der extrem hohen Sensitivität des Systems stets unterschiedliche Bewegungsabläufe. Um dennoch die Effizienz der Simulationen vergleichen zu können, wurden die benötigten Rechenzeiten auf die Anzahl der Funktionsaufrufe bezogen. Außerdem wurden nur die beiden ersten Sekunden des Vorgangs simuliert, während derer sich die Bewegungstrajektorien noch nicht allzu deutlich voneinander unterscheiden.

Abbildung 5.19 zeigt die durchschnittliche Rechenzeit je Funktionsaufruf dieser Simulationen. Die beschriebenen Einflüsse sind gut erkennbar. Bei grober Diskretisierung des Trichters reduziert sich der Rechenaufwand um etwa ein Drittel, wenn die Flächenzuordnung umgekehrt zum Referenzmodell definiert wird.

Noch signifikanter ist der Einfluss der Diskretisierung. Mit feinerer Darstellung des Trichters steigt die Effizienz rasch an, um sich im Bereich von ca. 1000 Dreiecken mehr als zu verdoppeln. Gleichzeitig nimmt die relative Auswirkung der Flächenzuordnung deutlich ab. Bei weiterer Verfeinerung vergrößert sich der Aufwand nur relativ langsam.

Somit lassen sich zwei Faustregeln zur Anwendung des PCM ableiten. Erstens sollte die Diskretisierung der Flächen möglichst so gewählt werden, dass die Polygone der Kontaktpaarung etwa gleich groß sind. Zweitens empfiehlt es sich, die Fläche mit den größeren BVs als Körper E zu definieren.

5.4 Fußgänger–Pkw–Unfall

Als weiteres Simulationsbeispiel wird ein Unfall eines Fußgängers mit einem Pkw betrachtet. Diese Anwendung wurde gewählt, weil sie extreme Anforderungen an die Kontaktmodellierung stellt: Hier treten zahlreiche Kontakte zwischen komplex geformten Oberflächen auf, die den Bewegungsablauf wesentlich bestimmen. Eindimensionale Modelle sind dafür praktisch nicht anwendbar, weil typischerweise Kontaktsituationen auftreten, für die die Bestimmung extremer Kontaktpunkte (s. Abschnitt 3.5.3) fehlschlägt [10].

Modellbeschreibung

Bevor genauer auf die Modellierung eingegangen wird, sei darauf hingewiesen, dass auch dieses Simulationsbeispiel rein zu Testzwecken dienen soll. Die eingehenden Parameter wurden möglichst plausibel gewählt, jedoch nicht durch reale Experimente oder Messungen verifiziert. Unter dieser Prämisse sind auch die Ergebnisse zu betrachten.

Die beiden Teilsysteme „Auto“ und „Mann“ wurden zunächst voneinander unabhängig modelliert und anschließend als Substrukturen in das MKS–Modell zur Unfallsimulation eingebunden. Im Gesamtmodell wurden schließlich die Kontakte und der Anfangszustand definiert.

Das Auto wird durch einen einzigen Körper dargestellt, der rheonom mit konstanter Geschwindigkeit von 30 km/h in Fahrtrichtung geführt wird. Damit werden weder Aktionen des Fahrers noch dynamische Rückwirkungen des Zusammenstoßes auf die Fahrzeugbewegung berücksichtigt. Die für die Kontaktberechnung durch das PCM verwendete Oberfläche des Pkw stellt einen Camaro Baujahr 1983 durch 1240 Dreiecke dar und stammt aus einer freien Internetbibliothek [90].

Abbildung 5.20 zeigt das MKS–Modell des 180 cm großen und 75,2 kg schweren Manns. Er wird durch 15 Starrkörper repräsentiert, die ausgehend vom Oberkörper durch Gelenke mit Drehfreiheitsgraden in Baumstruktur verbunden sind.

Tabelle 5.2 stellt die Gelenkfreiheitsgrade sowie die relativen Positionen der Gelenkkoppelpunkte zusammen. Um physiologisch unmögliche Bewegungen zu vermeiden, sind sämtliche Drehfreiheitsgrade auf bestimmte Winkel-

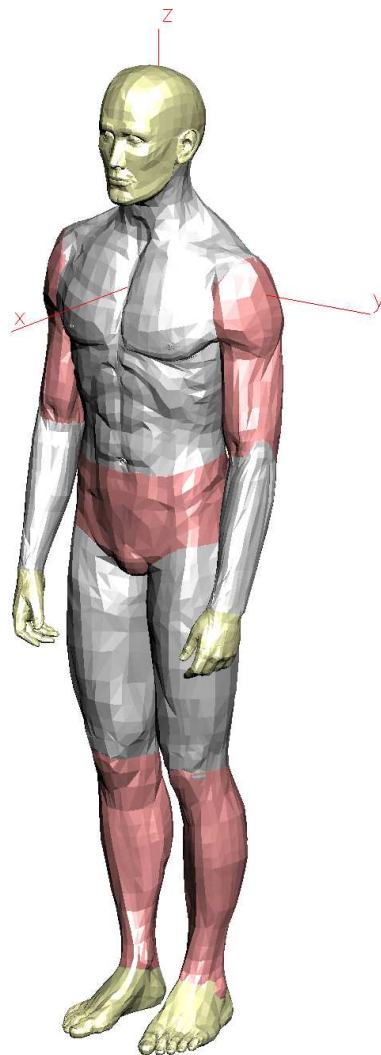


Abbildung 5.20: MKS-Modell des Manns der Fußgänger-Pkw-Unfall Simulation

bereiche beschränkt. Jenseits dieser Grenzen prägen Kraftelemente elastische Rückstellmomente ein, wobei mit einer linearen Steifigkeit von 980 Nm/rad gearbeitet wird.

Die Massen, Trägheitsmomente und Schwerpunktpositionen der Körperteile zeigt Tabelle 5.3. Sie wurden von einem zweidimensionalen, 14-gliedrigem Menschmodell abgeleitet, welches nach den Formeln von Dempster [93] parametrisiert worden war.

Gelenk	Freiheitsgrade		Position		
	Achse	Bereich [rad]	x [mm]	y [mm]	z [mm]
Oberkörper Kopf	1. α	-0,9..0,9	20	0	200
	2. β	-1,0..0,9			
	3. γ	-1,3..1,3			
Oberkörper Oberarm	1. α	0,0..1,5	0	190	0
	2. β	-2,6..1,0			
Oberkörper Unterleib	1. α	-0,6..0,6	20	0	-330
	2. β	-1,0..0,4			
	3. γ	-0,6..0,6			
Unterleib Oberschenkel	1. α	-0,2..0,7	-40	80	-150
	2. β	-2,0..0,9			
Oberarm Unterarm	1. β	-2,7..0,0	0	0	-300
Unterarm Hand	1. α	-1,3..0,5	0	-15	-281
Oberschenkel Unterschenkel	1. β	-0,1..2,5	10	0	-430
Unterschenkel Fuß	1. β	-0,7..1,0	-40	0	-425

Tabelle 5.2: Gelenke des Manns der Fußgänger-Pkw-Unfall Simulation: Freiheitsgrade und relative Koppelkoordinaten (bezüglich des körpfernäheren Koppelsystems) für die linke Körperhälfte

Durch eine umfangreiche Internetrecherche konnten auch für den Mann geeignete Polygonoberflächen gefunden werden [91]. Allerdings waren insbesondere die Flächen für Kopf, Hände und Füße extrem fein diskretisiert, was bei den ersten Tests zu relativ langen Rechenzeiten führte. Mit dem Shareware-Programm *AC3D* [62] lässt sich die Polygonalisierung in beliebigem Maße vergrößern. So konnte ein guter Kompromiss zwischen Effizienz und Detailtreue erzielt werden. In Tabelle 5.4 wird die Anzahl der Dreiecke der originalen und der reduzierten, für die Kontaktanalyse verwendeten Flächen gegenübergestellt.

Neben den 15 Kontaktpaarungen zwischen sämtlichen Körperteilen und dem Fahrzeug enthält das Modell zwölf weitere zwischen Körperteilen, die sich berühren können: Unterschenkel links – Unterschenkel rechts, Oberkörper – Unterarm links, Oberkörper – Unterarm rechts, Hand links – Ober-

Körperteil	m [kg]	J_x [kgm ²]	J_y [kgm ²]	J_z [kgm ²]	x_S [mm]	y_S [mm]	z_S [mm]
Kopf	6,0	0,025	0,028	0,018	20	0	90
Oberkörper	25,0	0,52	0,41	0,27	0	0	-140
Unterleib	12,5	0,13	0,08	0,13	-30	0	-110
Oberarm	2,1	0,0125	0,0125	0,0025	0	0	-150
Unterarm	1,2	0,0087	0,0087	0,0009	0	0	-130
Hand	0,45	0,0003	0,0004	0,0001	0	0	-50
Oberschenkel	7,5	0,148	0,148	0,024	0	0	-200
Unterschenkel	3,5	0,041	0,041	0,004	0	0	-180
Fuß	1,1	0,005	0,01	0,01	50	0	-35

Tabelle 5.3: Trägheitsdaten des Manns der Fußgänger-Pkw-Unfall Simulation: Masse, Trägheitsmomente (bezüglich des Schwerpunkts) und Schwerpunktposition (bezüglich des körpernäheren Gelenkkoppelsystems) für die linke Körperteile

schenkel links, Hand rechts – Oberschenkel rechts, Hand links – Oberschenkel rechts, Hand rechts – Oberschenkel links, Oberschenkel links – Unterarm links, Oberschenkel rechts – Unterarm rechts, Unterleib – Unterarm links, Kopf – Unterarm links und Kopf – Unterarm rechts.

Aus verschiedenen Gründen erwies sich die Bestimmung realitätsnaher Parameter für die Kontakte als besonders problematisch. Wegen großer Streuungen und schwieriger experimenteller Ermittlung lassen sich für biomechanische Systeme grundsätzlich nur grob angenäherte Materialkonstanten angeben. Weiter verhalten sich sowohl der menschliche Körper als auch Pkw-Karosserien bei Kontaktvorgängen ausgesprochen komplex. Die dem PCM zugrunde liegende Annahme linear-elastischer Randschichten kann daher nur eine sehr grobe Näherung der realen Vorgänge liefern. Darüber hinaus ist auch bei dieser Simulation fraglich, welche Werte für die essentiellen Parameter Randschichtdicke b in (4.24) und flächenbezogene Dämpfung d_l in (4.28) anzunehmen sind.

Um dennoch Werte zu bekommen, die zumindest in der richtigen Größenordnung liegen, wurde iterativ Übereinstimmung zwischen einem Gedankenexperiment und der Simulation hergestellt. Als Szenario wurde der Aufprall eines Oberschenkels auf der Motorhaube nach freiem Fall aus einem Meter Höhe betrachtet. Für diesen Vorgang wurde von einer Deformation im Bereich eines Zentimeters und eines einzigen Rückpralls ausgegangen.

Körperteil	Dreiecke	
	original	reduziert
Kopf	4270	1704
Oberkörper	4522	2710
Unterleib	2568	1538
Oberarm	1548	1236
Unterarm	1480	1182
Hand	3868	1542
Oberschenkel	1552	1240
Unterschenkel	1522	1216
Fuß	2094	834
insgesamt	35488	20452

Tabelle 5.4: Diskretisierung der Polygonoberflächen des Manns der Fußgänger-Pkw-Unfall Simulation

Das gewünschte Verhalten ergibt sich, wenn man für die Randschichten der beiden Körper einen Elastizitätsmodul von $E = 6,0 \cdot 10^5 \text{ N/m}^2$, eine Querkontraktionszahl von $\nu = 0,4$ und eine Dicke von $b = 10 \text{ mm}$ vorgibt. Dies entspricht der kombinierten Randschichtsteifigkeit $c_l = 6,429 \cdot 10^7 \text{ N/m}^3$ bei dem elastischen Eindringungsverhältnis $\lambda = 0,5$. Als zugehörige flächenbezogene Dämpfung wurde $d_l = 1,0 \cdot 10^5 \text{ Ns/m}^3$ ermittelt. Diese Werte werden für alle 15 Kontakte zwischen den Körperteilen und der Karosserie verwendet. Den verbleibenden zwölf Kontakten zwischen Körperteilen liegt ein Elastizitätsmodul von $E = 2,0 \cdot 10^5 \text{ N/m}^2$ und eine flächenbezogene Dämpfung von $d_l = 2,0 \cdot 10^5 \text{ Ns/m}^3$ bei ansonsten identischer Parametrierung zugrunde, was zur kombinierten Randschichtsteifigkeit $c_l = 2,143 \cdot 10^7 \text{ N/m}^3$ führt.

Ferner wurde für alle 27 Kontaktpaarungen die Eindringtiefe $u_d = 1 \text{ mm}$ zur Einblendung der Dämpfung nach (4.28) sowie die Reibzahl $\mu = 0,4$ und die Grenzgleitgeschwindigkeit $v_\varepsilon = 0,2 \text{ mm/s}$ zur Bestimmung der Reibkraft nach (4.34) vorgegeben.

Während für die Kontaktelementgenerierung zwischen Körperteilen die 2-Pass-Methode zur Anwendung kommt, fungieren bei den Kontakten zum Fahrzeug die feiner diskretisierten Polygonflächen des Manns als Master; dabei werden die zugehörigen Referenzsysteme als Bezugssysteme E für die Vektordarstellung benützt.

Abgesehen von den Krafterelementen zur Darstellung der Gelenkanschlüsse verfügt das Mensch-Modell über keinerlei eigene eingeprägte Kräfte oder Momente. Da willkürliche Bewegungen bei derartigen Unfällen bekanntlich

nur marginalen Einfluss auf die durch große Trägheitswirkungen bestimmte Dynamik haben, erscheint diese Vereinfachung akzeptabel. Um auch vor dem Zusammenstoß auf Muskel- und Bodenkräfte verzichten zu können, wurde der Mann im Anfangszustand nur wenige Millimeter vor dem Fahrzeug positioniert, sodass er nach kürzester Zeit freien Falls vom Fahrzeug erfasst wird.

Ergebnisse

Auch die in diesem Abschnitt gezeigten Ergebnisse wurden mit dem Integrationsverfahren DASSL berechnet, wobei $atol = rtol = 1,0 \cdot 10^{-3}$ als absolute und relative Toleranzen eingestellt war.

Die zahlreichen Körper, Kraftelemente und Kontakte der Fußgänger-Pkw-Unfall Simulation resultieren in einer nahezu unüberschaubaren Menge an Ergebnissen. Da viele geschätzte Parameter eingehen und keine vergleichbaren Meßergebnisse zur Verifikation vorliegen, werden an dieser Stelle nur einige wenige Resultate dargestellt.

Abbildung 5.21 zeigt einige Bilder des simulierten Bewegungsablaufs. Unter [74] lässt sich diese Animation aus verschiedenen Ansichten betrachten. Der Mann wird an den Beinen erfasst (0,01 bis 0,09 s), rollt über die Motorhaube ab (bis 0,23 s), wird im freien Fall auf das Dach geschleudert (bis 0,84 s) und bleibt dort nach insgesamt 1,5 Sekunden auf dem Rücken liegen.

Eine wesentliche Motivation zur Entwicklung des PCM waren erhebliche Probleme bei der Simulation von Fußgänger-Pkw-Unfällen mit eindimensionalen Kontaktmodellen und der damit verbundenen Kontaktpunkt-kinematik [10]. Das PCM warf dagegen beim hier vorgestellten Modell in zahlreichen Simulationsrechnungen mit unterschiedlichen Anfangszuständen, Parametrierungen und Integrationsverfahren und -konfigurationen keinerlei algorithmische oder numerische Probleme auf, sondern rechnete stets robust und lieferte plausible Ergebnisse.

Durch die Diagramme in Abbildung 5.22 wird die Komplexität der Kontaktanalyse verdeutlicht. Die Kurven stellen jeweils die Summe der entsprechenden Werte aus allen 27 Kontaktpaarungen dar.

Das obere Diagramm gibt die Anzahl der Kollisionstests zwischen Bounding Volumen pro Funktionsauswertung an. Hier wird deutlich, dass das Prinzip der Kollisionserkennung auf Basis von BV-Hierarchien in diesem Fall wie gewünscht funktioniert, denn der Rechenaufwand korreliert gut mit der Kontaktsituation. Erst wenn der Mann am Ende mit dem ganzen Körper auf dem Auto liegt, steigt die Anzahl der BV-Kollisionstests erheblich auf etwa 170000 an. Wird die Simulation nach 1,5 s fortgesetzt, verursacht daher die Kollisionserkennung den größten Anteil des Rechenaufwands.

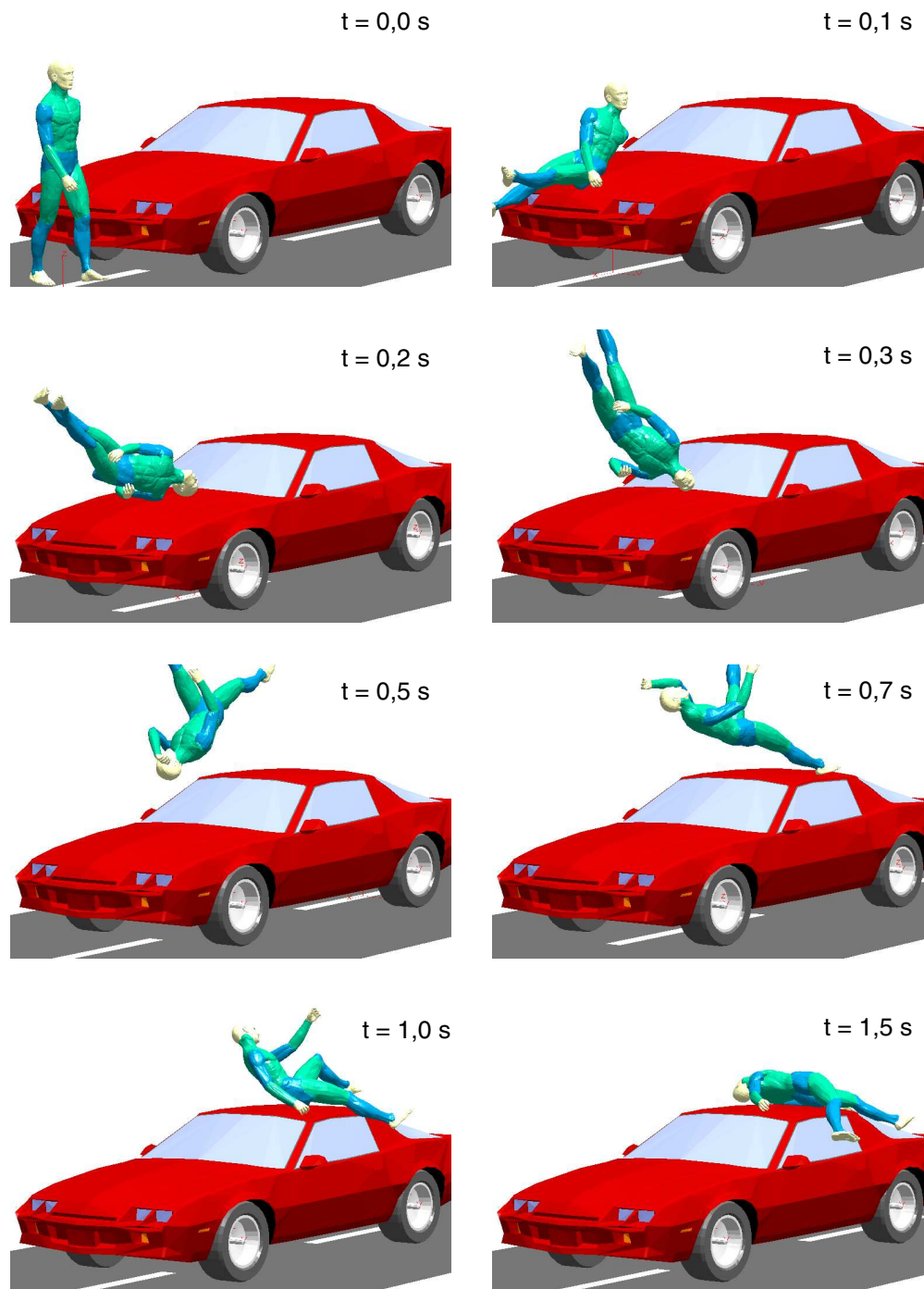


Abbildung 5.21: Animation der Fußgänger-Pkw-Unfall Simulation, Fahr-
geschwindigkeit 30 km/h

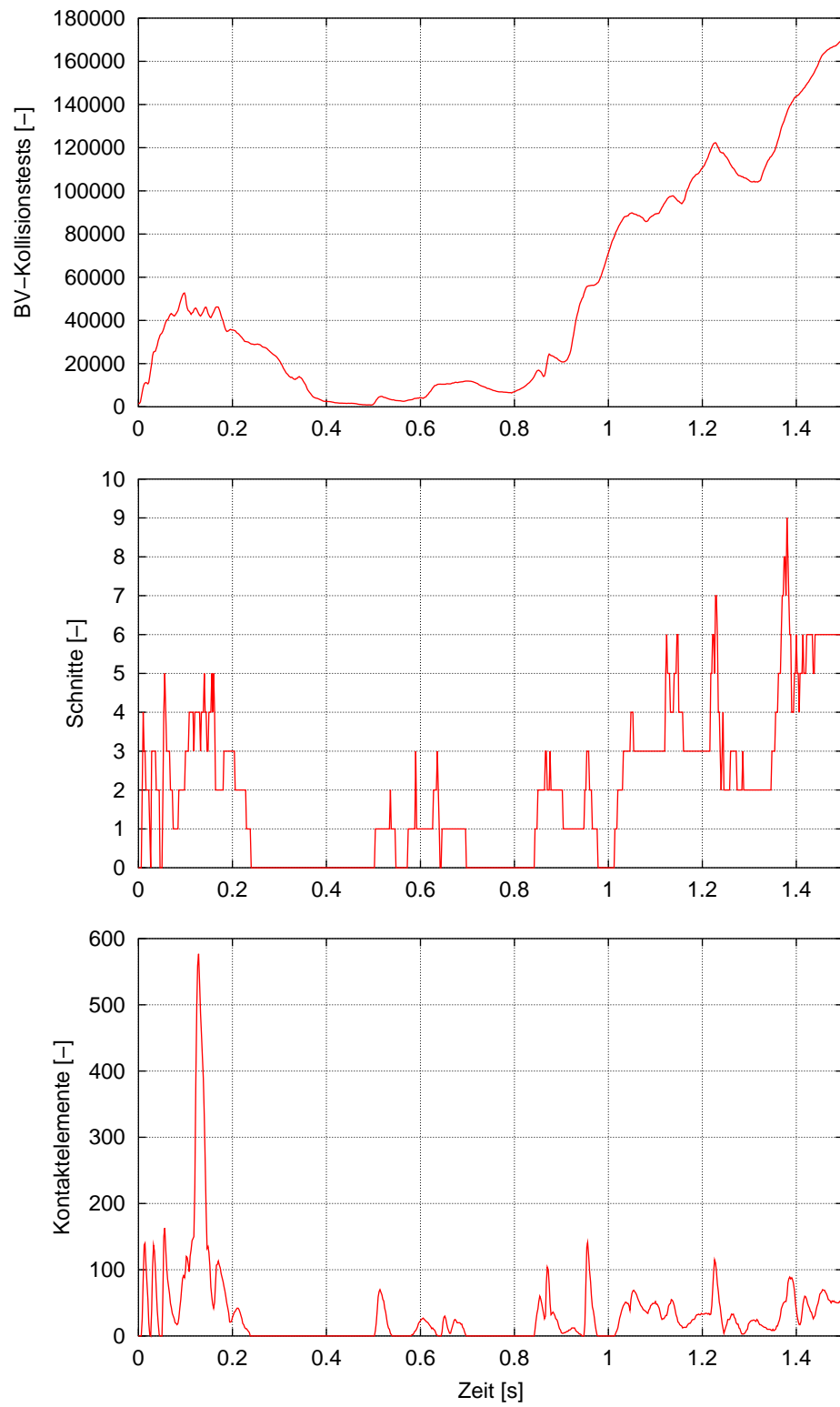


Abbildung 5.22: Anzahl der BV-Kollisionsprüfungen, Schnitte und Kontaktelemente sämtlicher Kontakte der Fußgänger-Pkw-Unfall Simulation

Die Kurve in der Mitte zeigt die Anzahl der Schnitte, d. h. der aktiven Kontaktpaarungen, an. Hier fällt auf, dass auch in der Freiflugphase Kontakte auftreten. Aus Betrachtung der einzelnen Paarungen geht hervor, dass es sich um bis zu drei gleichzeitige Berührungen zwischen dem Gesicht und dem linken Oberarm sowie einen Kontakt zwischen dem rechten Fuß und dem Fahrzeugdach handelt. Die größte Anzahl von neun Schnitten wird kurz vor 1,4 s erreicht, als der Mann mit dem ganzen Körper in Liegeposition auf dem Dach aufprallt.

Im unteren Diagramm wird die Gesamtzahl aktiver Kontaktelemente aufgetragen. Obwohl die für die Kontaktelementgenerierung maßgeblichen Oberflächen der Körperteile relativ fein diskretisiert sind, steigt der Wert selten über 100 an. Das deutliche globale Maximum von 577 bei 0,128 s ergibt sich im Wesentlichen aus dem Abrollvorgang des Oberkörpers über den linken Unterarm; die dabei auftretenden starken Druckspannungen rühren von 150 Elementen zwischen Unterarm und Motorhaube sowie 314 (wegen des 2-Pass-Verfahrens ungefähr verdoppelten) Elementen zwischen Unterarm und Oberkörper her. Die dem PCM-Konzept zugrunde liegende Annahme, dass nur ein kleiner Teil der Oberflächenpolygone in Kontakt treten, wird hier gut erfüllt.

Obwohl wegen der nur grob geschätzten Kontaktparameter von wenig realitätsnahen Ergebnissen ausgegangen werden muss, soll hier beispielhaft eine Unfall-biomechanische Bewertung erfolgen. Dazu wird der in Abbildung 5.23 dargestellte Verlauf des Betrags der Absolutbeschleunigung des Kopfschwerpunkts $a_K(t)$ betrachtet.

Eine Beurteilung der durch den Unfall zu erwartenden Schwere der Verletzung des Kopfs erlaubt z. B. das häufig verwendete *Head Injury Criterion* [94]:

$$HIC = \max \left[(t_2 - t_1) \left(\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \frac{a_K(\tau)}{g} d\tau \right)^{2,5} \right] \quad (5.16)$$

Für ein Intervall $[t_1, t_2]$ wird der Mittelwert der Kopfbeschleunigung in Vielfachen der Erdbeschleunigung g gebildet (runde Klammer), mit 2,5 potenziert und der Intervalldauer multipliziert. Das *HIC* ist definiert als Maximum über alle möglichen Zeitintervalle.

Mit einer erhöhten Abtastrate von 5000 Hz ergibt sich auf diese Weise ein *HIC* von 140 für den Zeitraum von $t_1 = 0,121$ s bis $t_2 = 0,215$ s, d. h. den Aufprall auf der Motorhaube. Dieser Wert liegt deutlich unter der Toleranzgrenze von 1000, was auf nur leichte Kopfverletzungen hindeutet [94].

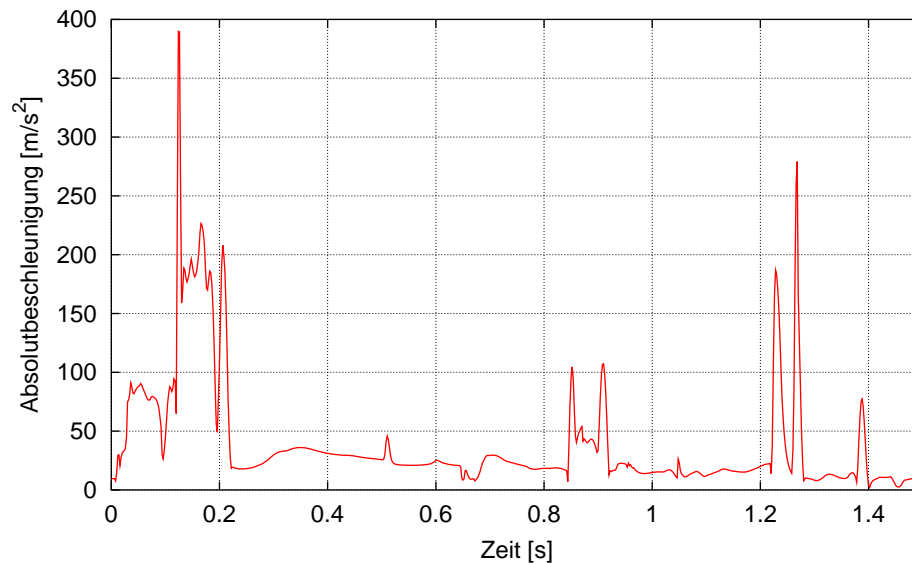


Abbildung 5.23: Absolutbeschleunigung des Kopfschwerpunkts des Manns bei der Fußgänger–Pkw–Unfall Simulation

Numerik und Effizienz

Das vorliegende MKS besitzt 29 Gelenkfreiheitsgrade des Manns und keine geschlossenen kinematischen Schleifen. Somit ergibt sich ein System gewöhnlicher Differentialgleichungen mit 58 Zustandsgrößen erster Ordnung. Neben den inneren Kräften der Gelenke wird die Bewegung des Manns durch 14 als Gelenkansschläge wirkende Krafterelemente, 27 PCM–Kontaktpaarungen und die Erdbeschleunigung bestimmt.

Wie bei Unfallsimulationen üblich, dauert der hochdynamische Vorgang nur kurze Zeit und weist einen entsprechend extremen Zeitmaßstab auf. Dies ist bei der Beurteilung von Numerik und Effizienz zu berücksichtigen.

Abbildung 5.24 zeigt die Verläufe der vom Integrator automatisch gewählten Schrittweite und Ordnung. Die Schrittweite bewegt sich bis auf wenige kleine Ausreißer stets zwischen 0,1 und 5 ms. Angesichts des schnellen, durch stark unterschiedlich dynamische Phasen gekennzeichneten Bewegungsablaufs erscheint dieser Bereich recht günstig. Auch die meist zwischen zwei und vier liegende Ordnung deutet darauf hin, dass keine erheblichen Probleme der Ordnungs- und Schrittweitensteuerung auftreten.

Zur Beurteilung der Effizienz wird in Abbildung 5.25 der zugehörige Echtzeitfaktor als Funktion der Zeit dargestellt. Er bewegt sich meist im Bereich von 100, weist aber recht viele starke Abweichungen um bis zu zwei Größen-

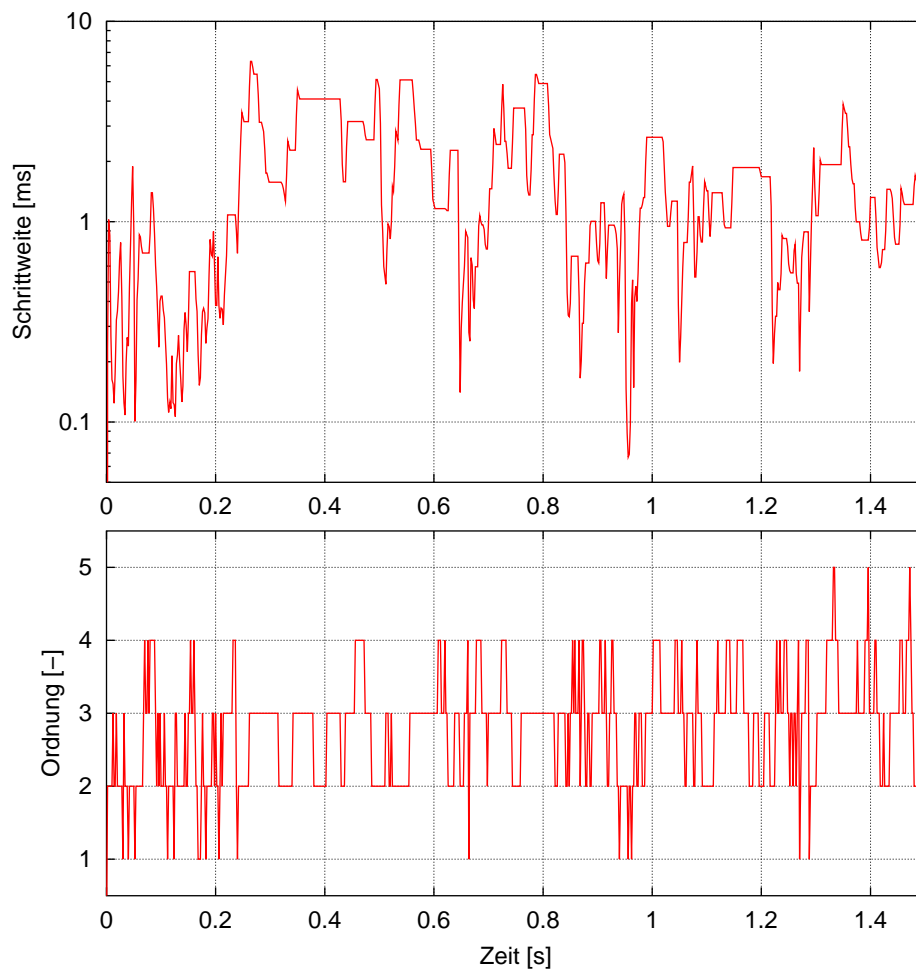
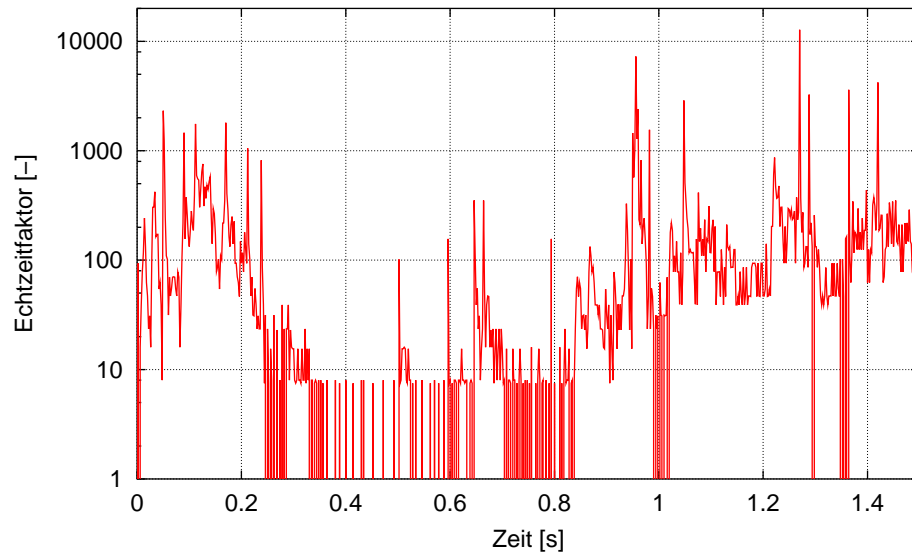


Abbildung 5.24: Schrittweite und Ordnung des Integrators DASSL bei der Fußgänger-Pkw-Unfall Simulation

ordnungen in beide Richtungen auf. Dies ist einerseits auf den ungleichförmigen Lösungsverlauf, andererseits auf das BDF-Integrationsverfahren mit variabler Schrittweite zurückzuführen.

In Tabelle 5.5 werden statistische Größen der Fußgänger-Pkw-Unfall Simulation für verschiedene Integrationsverfahren gegenübergestellt. Dieser Vergleich ist sinnvoll, weil sich das betrachtete System im Gegensatz zum Bouncing Bubbles Beispiel ausgesprochen stabil verhält. So treten für die unterschiedlichen Integratoren trotz recht grober Toleranzvorgaben über die gesamte Simulationszeit nur kleine Abweichungen in den Ergebnissen auf (Abbildung 5.26).

Abbildung 5.25: Echtzeitfaktor¹ der Fußgänger-Pkw-Unfall Simulation

Simulationszeit: 1,5 s	DOPRI5	LSODE [92]		DASSL	RADAU5
	[95]	Adams	Gear	[45]	[44]
Mittlere Schrittweite	2,00 ms	0,49 ms	0,79 ms	0,75 ms	1,32 ms
Funktionsauswertungen	4501	4563	32715	5020	25268
Jacobimatrixauswertungen	–	–	530	30	306
CPU-Zeit ¹	197,4 s	210,1 s	1517,4 s	232,1 s	1220,5 s
Mittlerer Echtzeitfaktor ¹	131,6	140,1	1011,6	154,7	813,7

Tabelle 5.5: Integrationsstatistik verschiedener Integratoren bei der Fußgänger-Pkw-Unfall Simulation

Betrachtet man die Gegenüberstellung hinsichtlich der Effizienz, fällt zunächst der erhebliche Unterschied zwischen den kurzen Rechenzeiten (< 4 min) der Verfahren DOPRI5, LSODE/Adams und DASSL und den vielfach größeren von LSODE/Gear und RADAU5 auf. Dabei ist gut erkennbar, dass der erhöhte Rechenaufwand nicht durch kleinere Schrittweiten, sondern zahlreiche numerische Berechnungen der Jacobimatrix mit entsprechend vielen Funktionsauswertungen verursacht wird.

¹PC, Win 2k, Xeon 2400 MHz Prozessor, 1024 MB RAM

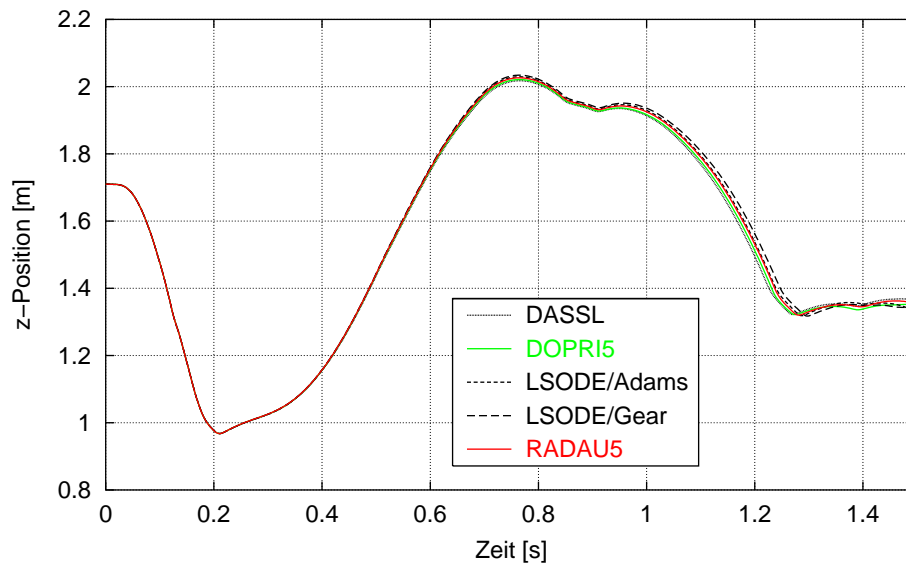


Abbildung 5.26: Absolute Höhe des Kopfschwerpunkts des Manns bei der Fußgänger-Pkw-Unfall Simulation für verschiedene Integrationsverfahren

Die schnellste Auswertung liefert DOPRI5, ein explizites Einschrittverfahren fünfter Ordnung vom Runge-Kutta-Typ mit variabler Schrittweite [95]. Hierbei kommt zum Tragen, dass das System nicht steif ist, sodass auch für explizite Verfahren relativ große Schritte möglich sind.

Nur wenig mehr Aufwand verursacht das ebenfalls nur für nicht-steife Systeme geeignete, auf der Adams-Bashforth-Moulton-Methode basierende Prädiktor-Korrektor-Verfahren LSODE mit Schrittweiten- und Ordnungssteuerung [92]. Bei deutlich kleinerer mittlerer Schrittweite und niedrigerer Ordnung führen etwas mehr Funktionsaufrufe zu entsprechend längerer Rechenzeit.

Ebenfalls noch sehr effizient ist die Auswertung mit dem im vorigen Abschnitt als Referenz benutzten Integrator DASSL (s. Abschnitt 2.3). Gegenüber DOPRI5 und LSODE/Adams bietet er die erheblichen Vorzüge, auch für steife Systeme geeignet zu sein und algebraische Nebenbedingungen berücksichtigen zu können. Obwohl für die Fußgänger-Pkw-Unfall Simulation 30-mal die System-Jacobimatrix ausgewertet wird, benötigt DASSL nur 18% mehr Rechenzeit als DOPRI5.

Deutlich höheren Rechenaufwand verursachen die Integratoren LSODE bei Verwendung des impliziten BDF-Verfahrens nach Gear [92] sowie der auf einem impliziten Runge-Kutta-Verfahren basierende, auch für DAEs geeignete RADAU5 [44]. Ursache hierfür ist die entsprechend größere Anzahl von Auswertungen der Jacobimatrix.

Die deutlichen Unterschiede zu DASSL sind allerdings nicht allein aufgrund der prinzipiellen Verfahrensmerkmale erklärbar. Als mögliche Ursache kommt die Wahl des Inkrements zur numerischen Bestimmung der partiellen Ableitungen der Jacobimatrix durch Differenzenapproximation in Betracht. Hier verfolgen die Algorithmen verschiedene Strategien, wobei das PCM offenbar besonders günstig mit DASSL zusammenspielt. Wird das Inkrement dagegen zu klein gewählt, wird zwar der theoretische Approximationsfehler verringert, gleichzeitig jedoch numerische Ungenauigkeiten, wie sie das relativ aufwändige PCM verursacht, überproportional verstärkt [96].

Kapitel 6

Zusammenfassung

6.1 Inhalt

Die vorliegende Arbeit beschreibt Grundlagen, Modellierung, Implementierung und Anwendung eines als Polygonal Contact Model (PCM) bezeichneten Verfahrens zur Analyse von Kontaktvorgängen zwischen komplex geformten Körpern im Rahmen der Mehrkörperdynamik.

Kapitel 1 führt kurz in den behandelten Problembereich ein. Die Methode der Mehrkörpersysteme und die Kontaktmechanik werden charakterisiert und eingeordnet. Nach Formulierung von Motivation und Ziel der Arbeit folgt eine knappe Übersicht der wichtigsten als Grundlage verwendeten Literaturstellen, bevor die weitere Gliederung erläutert wird.

Kapitel 2 stellt wichtige Grundlagen aus dem Bereich MKS-Simulation zusammen. Hierbei wird speziell auf MKS-Modellierungselemente, Struktur und automatische Erstellung der Bewegungsgleichungen sowie numerische Verfahren zu deren Lösung im Zeitbereich eingegangen. Nach einer Charakterisierung der klassischen MKS-Elemente werden Anwendung und Implementierung von Krafterelementen näher beleuchtet. Weiter werden die wichtigsten Begriffe und Merkmale von MKS-Formalismen erläutert und die Struktur der resultierenden Differentialgleichungssysteme dargelegt.

Die Lösung des Anfangswertproblems durch numerische Integration wird ausführlicher behandelt. Nach formaler Einteilung in gewöhnliche und differential-algebraische Systeme werden die wesentlichen Eigenschaften der in der Mehrkörperdynamik gebräuchlichen Integrationsverfahren erklärt, wobei Ordnung und Stabilität hervorgehoben werden. Schließlich wird die Funktionsweise des DAE-Integrators DASSL beschrieben und kurz auf Schaltfunktionen für die Integration unstetiger Systeme eingegangen.

Modelle und Methoden zur rechnerischen Analyse von Kontaktvorgängen sind Gegenstand des Kapitels 3. Nach Erläuterung grundlegender Prinzipien der Kontaktanalyse kommen einige Schwierigkeiten zur Sprache, die insbesondere bei der numerischen Berechnung von Bedeutung sind. Weiter werden bekannte Kontaktmodelle erläutert und in Hinblick auf den Einsatz in der Mehrkörperdynamik diskutiert.

Zu den Festkörpermodellen, bei denen der Verformungszustand der berührenden Körper vollständig bestimmt wird, zählen Elastizitätstheorie und Finite-Elemente-Methode. Während die Elastizitätstheorie Gleichungen liefert, die nur für einige spezielle Konfigurationen analytisch lösbar sind, ermöglicht die FEM wirklichkeitsnahe Simulationen auch für komplexe Oberflächengeometrien und Materialeigenschaften, ist aber mit sehr hohem Rechenaufwand verbunden.

Die Klasse der Oberflächenmodelle ist durch zweidimensionale Betrachtung der Verformungen und Spannungen im Kontaktgebiet gekennzeichnet. Die Halbraumapproximation basiert auf Diskretisierung in Kontaktelemente, deren elastisches Verhalten über Einflusskoeffizienten gekoppelt ist und mittels iterativer Verfahren oder Variationsrechnung bestimmt wird. Dagegen liefert das einfachere Randschichtmodell, bei dem von starren Körpern mit elastischer Randschicht ausgegangen wird, direkt den Spannungszustand für eine gegebene Kontaktsituation.

Den höchsten Abstraktionsgrad weisen die häufig in der MKS-Simulation verwendeten eindimensionalen Modelle auf, welche den Kontaktvorgang auf die Relativkinematik der betrachteten Körper in einem angenommenen Kontaktpunkt reduzieren. Die Bestimmung des Kontaktpunkts ist bei komplex geformten Körpern problematisch, weil Kontaktpunktsprünge, Mehrpunktkontakt oder Schmiegunge auftreten können. Zur Modellierung nachgiebiger Kontakte kommen Krafterelemente zum Einsatz, welche die Kontaktnormalkraft durch viskoelastische Kraftgesetze beschreiben. Beim Starrkörperkontakt wird dagegen von unverformbaren Oberflächen ausgegangen, und dynamische Vorgänge werden anhand von Stoßhypothesen dargestellt, was in der numerischen Simulation erheblichen Aufwand verursacht.

Das Kapitel schließt mit einer bewertenden Gegenüberstellung der Kontaktmodelle hinsichtlich physikalischer Modellqualität, Effizienz, Komplexität, Robustheit und Flexibilität, wobei sich zur Anwendung in der Mehrkörperdynamik die Oberflächenmodelle als besonders vorteilhaft erweisen.

Im Kapitel 4 wird der im Rahmen dieser Arbeit entwickelte MKS-Kontaktalgorithmus Polygonal Contact Model beschrieben. Zur Darstellung der Oberflächenprofile der Körper einer betrachteten Kontaktpaarung wird von aus Dreiecken gebildeten Polygonflächen ausgegangen. Die Kontaktanalyse für einen gegebenen Systemzustand umfasst bis zu vier Schritte.

Als erster Schritt wird eine Kollisionserkennung der beiden Flächen durchgeführt, die sämtliche Schnittlinien zwischen den Dreiecken der Oberflächen ermittelt. Um dies auch bei umfangreichen Flächen mit hoher Effizienz zu ermöglichen, kommen quaderförmige, körperfeste Hüllvolumina zum Einsatz, die die zugehörigen Flächen anhand einer hierarchischen Binärstruktur sukzessive mit wachsender räumlicher Auflösung unterteilen. Indem zunächst nur zwischen diesen einhüllenden Quadern systematisch nach Kollisionen gesucht wird, vermindert sich der Rechenaufwand gegenüber einer Berechnung ohne Hüllvolumenhierarchie um mehrere Größenordnungen.

Falls die Kollisionserkennung eine Berührung zwischen den Körpern ergibt, wird im zweiten Schritt das zugehörige Schnittvolumen konstruiert, welches durch die ineinander eingedrungenen Gebiete der unverformten Profile gebildet wird. Dazu werden zunächst die Schnittlinien mit einem auf Abstandsmessungen basierenden Algorithmus zu einem geschlossenen Schnittpolygonzug mit definierter Umlaufrichtung zusammengesetzt. Dann erfolgt die Bestimmung der aktiven Polygone der beiden Flächen, die das Schnittvolumen begrenzen. Unter Beachtung der Orientierung der Oberflächen werden diese durch Nachbarschaftsbeziehungen bestimmt, welche anhand geeigneter B-Rep-Datenstrukturen vorab ermittelt werden. Die Komplexität der zur Konstruktion der Schnittvolumina dienenden Verfahren erhöht sich erheblich wegen der Berücksichtigung mehrfacher und mehrfach berandeter Berührgebiete.

Im dritten Schritt, der Bestimmung der Kontaktelemente, wird die gemeinsame Berührfläche der Körper approximiert und in Kontaktelemente unterteilt. Dazu wurden zwei unterschiedliche Verfahren implementiert. Bei der Master-Slave-Methode wird zu jedem Dreieck der als Master definierten Fläche ein Kontaktelement generiert. Flächeninhalt und Normalenrichtung werden vom Master-Polygon übernommen, und als Position wird derjenige Punkt auf der Normale durch den Dreiecksschwerpunkt bestimmt, welcher den Abstand zu den Oberflächenprofilen entsprechend dem Verhältnis der Kontaktsteifigkeiten der beiden Körper teilt. Für die alternative Variational-Interpolation-Methode werden zuerst die beiden aktiven Flächenteile durch Variational Interpolation interpoliert, um dann durch Linearkombination in Abhängigkeit vom Verhältnis der Kontaktsteifigkeiten eine Näherung für den Verlauf der Berührfläche zu gewinnen. Diese in impliziter Darstellung vorliegende Fläche wird innerhalb des Schnittpolygonzugs anhand eines flächenbasierten Triangulierungsverfahrens in Dreiecke diskretisiert, welche ähnlich wie bei der Master-Slave-Methode jeweils ein Kontaktelement ergeben.

Zur Bestimmung der Kontaktkraft wird schließlich im vierten Schritt für jedes Kontaktelement ein Kontaktkraftvektor ermittelt, der sich aus Normal- und Tangentialkraft zusammensetzt. Die Normalkraft ergibt sich, unter An-

nahme konstanter Normalspannung über das Element, aus dem um kombinierte Randschichtsteifigkeiten erweiterten Randschichtmodell. Analog zum dabei auftretenden linearen Zusammenhang zwischen Eindringtiefe und elastischer Normalkraft wird ein linear-viskoser Dämpfungsterm mit flächenbezogener Dämpfungskonstante eingeführt. Die Tangentialkraft ergibt sich aus einer im Haftbereich regularisierten Version des Coulombschen Reibgesetzes. Durch Zusammenfassung der Kraftvektoren aller Kontaktelemente wird letztlich der resultierende Kontaktkraftwinder berechnet.

Kapitel 5 demonstriert die Anwendung des Polygonal Contact Model an vier Simulationsbeispielen.

Im ersten Beispiel wird die quasistatische Eindringung eines einfach geformten Körpers in einen Halbraum untersucht. Während sich bei Verwendung der Master-Slave-Methode ein plausibler Weg-Kraft-Verlauf ergibt, liefert die im Folgenden nicht weiter betrachtete Kontaktelementgenerierung mittels Variational-Interpolation infolge ausschwingender Approximation der Berührfläche unrealistische Kraftsprünge.

Als zweites Beispiel dient ein ebenfalls quasistatischer Kontaktvorgang zweier Kugeln. Hierbei werden die numerischen Ergebnisse der analytischen Auswertung mit dem Hertzschen Kontaktmodell sowie dem Randschichtmodell gegenüber gestellt. Mit wachsender Auflösung der Diskretisierung konvergiert dabei das PCM gegen das Randschichtmodell.

Das dritte Beispiel umfasst ein ausführliches Studium der Simulation dreier hantelförmiger Körper, die aufgrund der Schwerkraft gegen einen pyramidenförmigen Trichter prallen. Unter Berücksichtigung von Dämpfung und Reibung ergeben sich zahlreiche Kontaktvorgänge mit mehrfachen, mehrfach berandeten und schmiegenden Berührungen. Anhand zahlreicher Ergebnisdigramme werden die verfahrensnahen Ausgabegrößen des PCM erläutert. Im Rahmen der folgenden Effizienzanalyse wird eine Parameterstudie vorgestellt, die den Einfluss der Diskretisierung der Polygonflächen sowie der Wahl des Bezugssystems untersucht.

Zur Demonstration der Anwendung des PCM auf eine praxisnahe, umfangreiche Simulationsaufgabe wird im vierten Beispiel ein Fußgänger-Pkw-Unfall betrachtet. Ein 15-gliedriges Menschmodell mit komplexen Körperoberflächen wird seitlich von einem Sportwagen erfasst. Durch einen Vergleich der Rechenergebnisse verschiedener Integrationsverfahren werden die numerischen Eigenschaften des Systems herausgearbeitet.

Im Anhang A wird mit einer Erläuterung der Schnittstellen und der Struktur des Quellcodes kurz auf die Software-Implementierung des PCM eingegangen.

6.2 Bewertung und Einordnung

Mit dem PCM liegt ein neues Verfahren zur Kontaktanalyse in MKS vor, das in vieler Hinsicht erhebliche Vorteile gegenüber bekannten Methoden birgt. Die folgende Wertung bezieht sich auf das Master–Slave–Verfahren.

Wegen seines allgemeinen Ansatzes und seiner konsequent dreidimensionalen Implementierung kann das PCM für ein sehr weites Anwendungsspektrum eingesetzt werden. Seine Stärken zeigt es bei der Analyse von Kontakten beliebig komplex geformter Körper: Konkave Oberflächenabschnitte sowie daraus resultierende schmiegende, mehrfache und mehrfach berandete Berührungen können prinzipiell ohne Einschränkungen behandelt werden.

In der Praxis stellt die Verwendung von Polygonflächen zur Beschreibung der Körperoberflächen eine wesentliche Vereinfachung im Vergleich zu anderen Ansätzen dar, denn dieser Flächentyp ist bereits Standard zur grafischen Darstellung von Körpern in MKS wie in anderen Bereichen des CAE.

Ein wichtiges Kriterium bei der Entwicklung des PCM war Einfachheit. Auch wenn etwa zur Konstruktion der Schnittvolumina teilweise recht komplexe Algorithmen verwendet werden, ist die für die Kontaktkraftbestimmung wesentliche Grundidee der Diskretisierung der Berührgebiete durch Kontaktelemente leicht nachzuvollziehen. Die gegenwärtig implementierten Kraftgesetze sind ebenfalls einfach gehalten. Folglich werden Phänomene und Ergebnisse der Simulation auch für unerfahrene Anwender durchschaubar.

Trotzdem liefert das PCM i. Allg. realitätsnähere Ergebnisse als die bisher in MKS–Programmen zur Verfügung stehenden eindimensionalen Modelle; denn bei der Zusammenfassung von Kontaktkinematik und -kraftgesetzen in einen einzigen Kontaktpunkt gehen i. d. R. wesentliche physikalische Phänomene verloren. Dagegen bildet das PCM die Kontaktspannungen im gesamten Berührgebiet ab, sodass es beispielsweise Kipp– oder Bohrmomente ohne zusätzliche Modellierungen berücksichtigt. Die Parametrierung ist dabei im Prinzip identisch und insbesondere nicht aufwändiger als bei einseitigen Krafterelementen.

Mangelnde modellierungsbedingte oder numerische Robustheit ist ein häufig beobachteter Schwachpunkt von Algorithmen zur Kontaktanalyse. Die vorliegende Implementierung des PCM erwies sich dagegen in vielen hundert Testsimulationen mit unterschiedlichen Modellen, Integrationsverfahren und auf verschiedenen Rechnerplattformen als ausgesprochen robust; es traten weder Programmabstürze noch Abbrüche der Zeitintegration auf. Ein für die Anwendung des PCM tendenziell problematisches Szenario können allenfalls schmiegende, sehr steife Kontakte mit extrem kleinen Eindringtiefen darstellen. Hierzu liegen bislang keine praktischen Erfahrungen vor.

Die Beurteilung der Effizienz hängt stark von den anwendungsspezifischen Anforderungen ab. So arbeitet der Algorithmus um mehrere Größenordnungen schneller als die FEM. Andererseits kann das PCM höchstens sehr einfache Kontaktprobleme in echtzeitkritischen Umgebungen wie *Virtual Reality* oder *Hardware in the Loop* lösen. Ein direkter Vergleich zu den in der MKS-Welt üblichen eindimensionalen Kontaktmodellen ist schwierig, weil u. a. die Diskretisierung der verwendeten Polygonflächen eine wichtige Rolle spielt. Nach den bisherigen Erfahrungen ist von einer Erhöhung der Rechenzeiten um den Faktor zehn bis 100 auszugehen. Das erklärte Ziel, ein leistungsfähigeres Verfahren zu schaffen, das dennoch MKS-typische Effizienz aufweist, wurde damit in vollem Umfang erreicht. Das Verhältnis zwischen Aufwand und Modellqualität erscheint ausgewogen.

Neben der methodischen Dokumentation durch die vorliegende Arbeit ist über [74] der vollständig dokumentierte Programmcode zum PCM verfügbar. Die Programmierschnittstellen erlauben eine einfache Integration in existierende MKS-Software. Damit kann die Entwicklung des PCM mit minimalem Aufwand von der Wissenschaftlergemeinschaft weiterverfolgt werden. Im Rahmen dieser Arbeit wurde exemplarisch die Einbindung in SIMPACK diskutiert, die auch Grundlage der Testrechnungen im Kapitel 5 ist.

Neben den positiven Eigenschaften des PCM traten im Laufe der Entwicklung und Testsimulationen sowie beim Einsatz in anderen Projekten auch einige Nachteile zu Tage.

Das PCM stellt hohe Anforderungen an die Qualität der verwendeten Polygonflächen. So müssen sie zur erfolgreichen Generierung der DCEs (s. Abschnitt 4.2.2) folgende Eigenschaften erfüllen: Ihre Polygone müssen *konsistente Orientierung* aufweisen und sie müssen *einfach zusammenhängend, einmannigfaltig* (d. h. höchstens zwei Polygone je Kante) und von *fehlerfreier Topologie* (d. h. ohne mehrfach definierte ortsgleiche Eckpunkte) sein. Darüber hinaus dürfen zur fehlerfreien Bestimmung der aktiven Elemente *keine Lücken, Löcher oder Ränder im Berührgebiet* auftreten (s. Abschnitte 4.2.2 und 4.3). Weiter ist zu beachten, dass bei Verwendung der Master-Slave-Methode die Master-Fläche die Diskretisierung des Kontaktgebiets bestimmt, indem in jedem aktiven Dreiecksschwerpunkt ein Kontaktelement generiert wird. Schließlich wird empfohlen, nur aus Dreiecken bestehende Flächen zu verwenden, um ungünstige Polygonzerlegungen bei der automatischen, geometrisch unkontrollierten Umwandlung durch das PCM zu vermeiden.

Prinzipiell ist Vorsicht geboten bei der Definition von Kontaktoberflächen mit spitzen Kanten und Ecken. Zwar liefert das PCM auch hier i. d. R. plausible Ergebnisse, man verlässt allerdings den Gültigkeitsbereich des zugrunde liegenden Randschichtmodells und somit auch den Bereich physikalisch be-

gründeter Kontaktmodellierung.

Wie bei den Beispielen im Kapitel 5 bereits deutlich wurde, kann sich die Bestimmung einiger Parameter des PCM problematisch gestalten. Hierbei ist zu beachten, dass es sich prinzipiell um Größen handelt, die auch bei eindimensionalen Kraftelementen geeignet vorgegeben werden müssen.

Zuerst sind in diesem Zusammenhang die Randschichtsteifigkeit c_l und die zugehörige flächenbezogene Dämpfung d_l zu nennen (s. Abschnitt 4.4.1). Zwar lässt sich c_l im Prinzip aus den Materialkonstanten und der Dicke der Randschicht bestimmen, doch weisen Körper in praktischen Anwendungen meist nicht tatsächlich elastische Randschichten auf. Damit können zwar die physikalischen Phänomene der Verformung im Kontaktgebiet näherungsweise beschrieben werden, geeignete Werte für Steifigkeit und Dicke der Randschicht stehen jedoch nicht direkt zur Verfügung. Außerdem ist in der Praxis unabhängig vom verwendeten Kontaktmodell die quantitative Beschreibung der Dämpfung ein offenes Problem. Um realitätsnahe Simulationsergebnisse zu erhalten, sind meist Verifikationsrechnungen zur Bestimmung von c_l und d_l zweckmäßig. Falls keine verwertbaren Messungen verfügbar sind, kann man sich (wie z. B. im Abschnitt 5.4) mit Gedankenexperimenten behelfen, um die richtigen Größenordnungen zu treffen.

Zwei weitere Parameter, die nicht direkt aus den Eigenschaften des zu simulierenden Systems hervorgehen, sind die Eindringtiefe u_d zur Einblendung der Dämpfung (s. Abschnitt 4.4.1) sowie die tangentielle Grenzrelativgeschwindigkeit v_ε zur Regularisierung der Coulombschen Reibung (s. Abschnitt 4.4.2). Hierbei handelt es sich um Größen, die zur Verstetigung der Kraftgesetze dienen, um numerische Probleme zu umgehen. Es ist vorteilhaft, dass sie physikalisch interpretierbar sind und vom versierten Anwender meist mit praktikablen Erfahrungswerten versehen werden können.

Schließlich erweist sich das regularisierte Reibgesetz (4.34) mitunter als problematisch, wenn v_ε unterschritten und damit der Coulombsche Gleitreibungsbereich verlassen wird. Das dann wie ein steifer Dämpfer wirkende Kraftgesetz approximiert den in manchen Anwendungen bedeutsamen Haftzustand durch viskoses Kriechen. Auch kann dadurch das System der Bewegungsgleichungen numerisch steif werden, sodass sich klassische, nur für nicht-steife Probleme geeignete Integrationsverfahren ineffizient bzw. instabil verhalten.

6.3 Ausblick

Bei der Entwicklung und im Umgang mit dem PCM ergaben sich einige Ideen und Wünsche für Erweiterungen, die im Folgenden kurz zusammengestellt werden.

Vor allem für Anwendungen, bei denen nur selten Berührungen stattfinden, könnte die Implementierung der Kollisionserkennung als Schaltzustand vorteilhaft sein. Bei Verwendung eines Integrators mit Schaltfunktionalität wäre es damit möglich, die eigentliche PCM-Kontaktanalyse nur in Phasen der Kollision aufzurufen. Zur Auswertung der Schaltfunktion könnte die Kollisionserkennung im vereinfachten und effizienteren Modus betrieben werden, in dem der Algorithmus sofort abbricht, sobald eine im Schnitt befindliche Polygonpaarung gefunden wird (s. Abschnitt 4.1.3). Dabei muss allerdings sichergestellt sein, dass der verwendete Algorithmus zur iterativen Bestimmung der Schaltzeitpunkte das diskrete Ergebnis der Kollisionserkennung (unstetiger Sprung zwischen wahr und falsch) verarbeiten kann.

Für entsprechende Einsatzzwecke könnte eine Modifikation des Randschichtmodells interessant sein. So wäre es einfach möglich, die den Kontaktelementen zugrunde liegende Parallelschaltung von linearer Feder und linearem Dämpfer durch andere rheologische Modelle zu ersetzen, sodass z. B. viskoses Kriechen oder plastische Verformung abgebildet werden kann. Falls dabei kontaktelementspezifische Zustände anfallen, müssen entsprechende Datenstrukturen geschaffen werden, die einen Zugriff über die entsprechende Polygon-ID erlauben.

Um auch Haftreibung realitätsnah simulieren zu können, wäre ein alternatives Kraftgesetz zur Bestimmung der Tangentialkräfte wünschenswert. Im Prinzip kommt dafür das im Abschnitt 3.5.1 vorgestellte Bürstenmodell in Frage. Dazu müsste allerdings für jedes aktive Kontaktelement der Schaltzustand Haften/Gleiten verwaltet und berechnet werden, was die entsprechenden Integrationsverfahren in so großer Anzahl erfahrungsgemäß nicht effizient leisten können.

Einige der erwähnten offenen Probleme im PCM ließen sich lösen, indem das Randschichtmodell durch die im Abschnitt 3.4.1 vorgestellte Halbraumapproximation ersetzt würde: Statt der schwer zu handhabenden Randschichtsteifigkeit könnte direkt mit stets bekannten Materialkonstanten gearbeitet werden, und auf das regularisierte Coulombsche Reibgesetz mit seinen Schwächen im Haftbereich ließe sich durch die Bestimmung konsistenter Haft- und Gleitspannungen verzichten. Als weiterer Vorteil ist die exaktere und den meisten technischen Anwendungen besser entsprechende Modellvorstellung elastischer Halbräume zu nennen.

Mit der Generierung von Kontaktelementen im Berührgebiet liefert das PCM die erforderlichen Voraussetzungen zur Anwendung der Halbraumapproximation. Allerdings stellt die Bestimmung der Einflusskoeffizienten aus (3.17) für deutlich gekrümmte Oberflächen ein schwieriges Problem dar, das tiefgehende Überlegungen theoretischer Art erfordert. Außerdem wäre der numerische Aufwand in jeder Kontaktanalyse erheblich, um eine konsis-

tente Verteilung der Kontaktspannungen zu ermitteln, sodass mit deutlichen Effizienzeinbußen gegenüber der auf dem Randschichtmodell basierenden Lösung zu rechnen ist.

Ein von Anwenderseite häufig formulierter Wunsch ist die Erweiterung des PCM auf elastische Körper. Da sich die Topologie einer Polygonfläche bei Verformung nicht verändert, sind dafür kaum Änderungen an den komplexen Algorithmen zur Bestimmung der Kontaktvolumina erforderlich. Umfangreichere Modifikationen sind dagegen bei den Datenstrukturen und insbesondere in der Kollisionserkennung nötig, wo man nicht mehr wie bei starren Körpern die konstanten Positionen der Eckpunkte gezielt ausnützen kann [97].

Anhang A

Implementierung

Über die PCM-Webseite [74] steht ein Archiv mit dem kommentierten Quellcode (Programmiersprache C) und Simulationsmodellen zur Verfügung. Im Folgenden werden in knapper Form Schnittstellen und Struktur der Implementierung erläutert.

A.1 Schnittstellen

Das PCM besitzt Schnittstellen zur Anbindung an MKS-Programme als Kraftelement. Dazu werden drei Funktionsaufrufe zur Verfügung gestellt:

`pcm_defcont` dient zur Definition einer Kontaktpaarung. Hierbei werden sämtliche Kontakt- und Verfahrensparameter festgelegt und die Pfade und Namen der Dateien mit den Polygonflächen im *Wavefront*-Format (Dateinamenserweiterung *obj*) übergeben. Während der Ausführung werden die Dateien eingelesen und Vorabrechnungen (Pre-Processing) wie die Erstellung der BV-Hierarchien und der DCELS ausgeführt. Als Rückgabe erhält man eine ganzzahlige Kennung, mit der die Kontaktpaarung bei den folgenden Aufrufen identifiziert wird.

`pcm_calcont` dient zur Analyse einer Kontaktpaarung für einen gegebenen Systemzustand. Als Eingabeparameter werden die relativkinematischen Größen der Koppelmarker auf Lage- und Geschwindigkeitsebene übergeben. Als Rückgabewerte erhält man neben den Kontaktkraft- und -momentenvektoren zusätzliche Zwischenergebnisse wie z. B. die Anzahl der Schnittlinien oder die Kontaktfläche.

`pcm_freemem` dient zur Freigabe des Arbeitsspeichers nach der Simulation.

Um die Anbindung sowohl an mit FORTRAN als auch mit C/C++ implementierte Programme zu ermöglichen, wurden sämtliche Schnittstellenparameter durch Pointer (*call by reference*) definiert, und die Unterprogramme liefern keinen Funktionswert (*void*).

pcm_defcont

```
void pcm_defcont( int      *task
                 , char   *fncse
                 , char   *fnscsf
                 , double *sfe
                 , double *sff
                 , int    *modee
                 , int    *modof
                 , double *ymode
                 , double *ymodf
                 , double *prate
                 , double *pratf
                 , double *ldepe
                 , double *ldepf
                 , double *adamp
                 , int    *dtmod
                 , double *dtdpt
                 , double *frcoe
                 , double *frvel
                 , double *grids
                 , int    *idcnt
                 , int    *err
                 , long   nfnscse
                 , long   nfnscsf
                 )
```

task (Eingabe) definiert den Betriebsmodus:

- 0 = Nur Kollisionserkennung
- 1 = Randschichtmodell: 2-Pass Master-Slave
- 2 = Randschichtmodell: Master/Slave = Kontaktfläche E/F
- 3 = Randschichtmodell: Master/Slave = Kontaktfläche F/E
- 4 = Randschichtmodell: Variational-Interpolation

fncse (Eingabe) definiert Pfad und Namen der *Wavefront*-Datei, welche die Polygonfläche *E* beschreibt.

fncsf (Eingabe) definiert Pfad und Namen der *Wavefront*-Datei, welche die Polygonfläche F beschreibt.

sfe (Eingabe, [-]) definiert den Skalierungsfaktor ($v := \mathbf{sfe} \cdot v$) für die Koordinaten der Polygonfläche E .

sff (Eingabe, [-]) definiert den Skalierungsfaktor ($v := \mathbf{sff} \cdot v$) für die Koordinaten der Polygonfläche F .

modee (Eingabe) definiert die Orientierung der Polygone der Fläche E :
 0 = Orientierung der *Wavefront*-Datei
 1 = Umgekehrte Orientierung der *Wavefront*-Datei

modf (Eingabe) definiert die Orientierung der Polygone der Fläche F :
 0 = Orientierung der *Wavefront*-Datei
 1 = Umgekehrte Orientierung der *Wavefront*-Datei

ymode (Eingabe, [N/m²]) definiert den Elastizitätsmodul E_E der Randschicht des Körpers E , wobei **ymode** = 0 für einen Starrkörper ohne elastische Randschicht steht.

ymodf (Eingabe, [N/m²]) definiert den Elastizitätsmodul E_F der Randschicht des Körpers F , wobei **ymodf** = 0 für einen Starrkörper ohne elastische Randschicht steht.

prate (Eingabe, [-]) definiert die Querkontraktionszahl ν_E der Randschicht des Körpers E .

pratf (Eingabe, [-]) definiert die Querkontraktionszahl ν_F der Randschicht des Körpers F .

ldepe (Eingabe, [m]) definiert die Dicke b_E der Randschicht des Körpers E .

ldepf (Eingabe, [m]) definiert die Dicke b_F der Randschicht des Körpers F .

adamp (Eingabe, [N/m³]) definiert die flächenbezogene Dämpfungskonstante d_i der kombinierten Randschichten.

dtmod (Eingabe) definiert den Betriebsmodus der Einblendung der Dämpfungskraft:
 0 = keine Einblendung
 1 = lineare Einblendung bis zur Eindringtiefe **dtcpt**

dtcpt (Eingabe, [m]) definiert die Eindringtiefe u_d zur Einblendung der Dämpfungskraft.

frcoe (Eingabe, [-]) definiert die zwischen den Randschichten wirksame Gleitreibungszahl μ , wobei **frcoe** = 0 für reibungsfreien Kontakt steht.

frvel (Eingabe, [m/s]) definiert die Grenzgeschwindigkeit v_ε zur Regularisierung des Coulombschen Haftzustands.

grids (Eingabe, [m]) definiert die Referenzlänge δ_t der Triangulierung bei Verwendung der Variational-Interpolation-Methode (sonst ohne Bedeutung).

idcnt (Ausgabe) dient als Kennung zur Identifikation der Kontaktpaarung.

err (Ausgabe) gibt den Fehlerstatus der Ausführung an:

0 = fehlerfrei

1 = Fehler wurde erkannt, der im Weiteren keine Simulation erlaubt

nfncse (Eingabe) gibt die Länge der Zeichenkette **fnscse** an. Dieser Wert wird nicht verarbeitet, sondern dient lediglich der Kompatibilität bei der Anbindung an mit FORTRAN implementierte Programme.

nfncsf (Eingabe) gibt die Länge der Zeichenkette **fnscsf** an. Dieser Wert wird nicht verarbeitet, sondern dient lediglich der Kompatibilität bei der Anbindung an mit FORTRAN implementierte Programme.

pcm_calcont

```
void pcm_calcont( int      *idcnt
                 , double  A_EF[3][3]
                 , double  r_EF[3]
                 , double  w_EF[3]
                 , double  v_EF[3]
                 , double  force[3]
                 , double  torque[3]
                 , double  outval[30]
                 , int      *err
                 )
```

idcnt (Eingabe) dient als Kennung zur Identifikation der Kontaktpaarung.

A_EF (Eingabe, [-]) gibt die relative Orientierung der Koppelmarker als Transformationsmatrix $A_{M_e M_f}$ an (C-Indizierung).

r_EF (Eingabe, [m]) gibt die relative Position $\mathbf{r}_{M_e M_f}$ des Koppelmarkers M_f gegenüber M_e an (Koordinatendarstellung bezüglich M_e).

w_EF (Eingabe, [rad/s]) gibt die relative Winkelgeschwindigkeit $\omega_{M_e M_f}$ des Koppelmarkers M_f gegenüber M_e an (Koordinatendarstellung bezüglich M_e).

v_EF (Eingabe, [m/s]) gibt die relative Geschwindigkeit $\mathbf{v}_{M_e M_f}$ des Koppelmarkers M_f gegenüber M_e an (Koordinatendarstellung bezüglich M_e).

force (Ausgabe, [N]) gibt den Kontaktkraftvektor an, der auf den Körper E an der Position des Koppelmarkers M_f (!) wirkt (Koordinatendarstellung bezüglich M_e).

torque (Ausgabe, [Nm]) gibt den resultierenden Momentenvektor an, der auf den Körper E wirkt (Koordinatendarstellung bezüglich M_e).

outval (Ausgabe) gibt verschiedene Zwischenergebnisse der Kontaktanalyse an:

outval [0] = Anzahl der BV-Kollisionstests
outval [1] = Anzahl der Schnitttests zwischen Dreiecken
outval [2] = Anzahl der Schnittlinien
outval [3] = Anzahl der Schnittpolygonzüge
outval [4] = Anzahl der Schnitte
outval [5] = Anzahl der Kontaktelemente
outval [6] = CPU-Zeit der Kollisionserkennung [s]
outval [7] = CPU-Zeit der Konstruktion der Schnittpolygone [s]
outval [8] = CPU-Zeit der Bestimmung der Kontaktelemente [s]
outval [9] = CPU-Zeit der Bestimmung der Kontaktkraft [s]
outval [10] = CPU-Zeit der gesamten Kontaktanalyse [s]
outval [11] = Kontaktfläche [m²]
outval [12] = Gewichtete Eindringtiefe [m]
outval [13] = Maximale Eindringtiefe [m]
outval [14] = Minimale Normalgeschwindigkeit [m/s]
outval [15] = Gewichtete Normalgeschwindigkeit [m/s]
outval [16] = Maximale Normalgeschwindigkeit [m/s]
outval [17] = Minimale Tangentialgeschwindigkeit [m/s]
outval [18] = Gewichtete Tangentialgeschwindigkeit [m/s]
outval [19] = Maximale Tangentialgeschwindigkeit [m/s]
outval [20] = Gewichtete Normalspannung [N/m²]
outval [21] = Maximale Normalspannung [N/m²]
outval [22] = Gewichtete Tangentialspannung [N/m²]
outval [23] = Maximale Tangentialspannung [N/m²]
outval [24] = x -Koordinate der gewichteten Kontaktposition [m]
outval [25] = y -Koordinate der gewichteten Kontaktposition [m]

outval[26] = z -Koordinate der gewichteten Kontaktposition [m]
 outval[27] = x -Koordinate des gewichteten Normalenvektors [-]
 outval[28] = y -Koordinate des gewichteten Normalenvektors [-]
 outval[29] = z -Koordinate des gewichteten Normalenvektors [-]

err (Ausgabe) gibt den Fehlerstatus der Ausführung an:
 0 = fehlerfrei
 1 = Fehler wurde erkannt, Rückgabewerte undefiniert

pcm_freemem

```
void pcm_freemem( int *err )
```

err (Ausgabe) gibt den Fehlerstatus der Ausführung an:
 0 = fehlerfrei
 1 = Fehler wurde erkannt

A.2 Struktur

Der Quellcode des PCM umfasst 50 Dateien (ca. 330 kB) mit etwas mehr als 10000 Zeilen Programmcode. Im Umgang damit ist es hilfreich, die Aufgaben der einzelnen Routinen und die Hierarchie ihrer Aufrufe (*call tree*) zu kennen. In den folgenden Übersichten werden nach außen unsichtbare Funktionen (*private*) durch Einklammerung gekennzeichnet. Die Variational-Interpolation-Methode bleibt hier unberücksichtigt.

pcm_defcont

```

pcm_defcont
|
|--pcm_addcosu
|  |
|  |--(pcm_readcsli)
|  |
|  |--(pcm_creacosu)
|
|--pcm_cosupre
|  |
|  |--pcm_creadcel
|  |
|  |--pcm_calcsgeo
|  |

```

```

| +-pcm_creabvtr
| |
| +- (pcm_creabvkn)
| |
| | +-pcm_centsort
| | |
| | | +- (pcm_creahslv)
| | | |
| | | +- (pcm_sortheap)
| |
|
+-pcm_addcont

```

`pcm.addcosu` liest eine Polygonfläche aus einer *Wavefront*-Datei ein.

`pcm.readcsli` liest eine Zeile aus einer ASCII-Datei.

`pcm.creacosu` erzeugt ein neues Kontaktflächen-Datenobjekt.

`pcm.cosupre` führt Vorabrechnungen für eine Kontaktfläche aus.

`pcm.creadcel` erzeugt Doubly Connected Edge Lists einer Kontaktfläche.

`pcm.calcsgeo` berechnet zusätzliche Geometriedaten einer Kontaktfläche.

`pcm_creabvtr` erzeugt die Hüllvolumenhierarchie einer Kontaktfläche.

`pcm_creabvkn` erzeugt rekursiv die Hüllvolumenhierarchie einer Kontaktfläche.

`pcm_centsort` sortiert die Koordinaten der Polygonschwerpunkte einer Kontaktfläche.

`pcm_creahslv` erzeugt rekursiv einen *Heapsort*-Baum.

`pcm_sortheap` sortiert rekursiv einen *Heapsort*-Baum.

pcm_calcont

```

pcm_calcont
|
+-pcm_coldet
| |
| +--(pcm_cdcheck)
| |
| +--(pcm_ticheck)
| |
| +-pcm_tritri
|
+-pcm_creaispo
| |
| +--(pcm_eudisqr)
| |
| +--(pcm_swpmvec)
|
+-pcm_cractelm
| |
| +--(pcm_addifip)
| | |
| | +--(pcm_addactid)
| |
| +--(pcm_srchmvip)
| | |
| | +--(pcm_disinsec)
| | |
| | +--(pcm_addifip)
| |
| +--(pcm_srchinae)
| | |
| | + +--(pcm_srchmvip)
| | |
| | +--(pcm_addactid)
| |
| +--(pcm_linactid)
|
+-pcm_crcemasl
| |
| +--(pcm_addactid)
| |
| +-pcm_calcntel
|

```

```
+pcm_calctfrc  
|  
+pcm_frcntmem
```

`pcm_coldet` führt eine Kollisionserkennung zweier Kontaktflächen durch.

`pcm_cdcheck` führt rekursiv eine Kollisionserkennung zweier Kontaktflächen durch. Entspricht `Schnitt` und `BVCheck` aus Abschnitt 4.1.3.

`pcm_ticheck` führt einen Schnittttest zweier Dreiecke durch. Entspricht `PolyCheck` aus Abschnitt 4.1.3.

`pcm_tritri` führt einen Schnittttest zweier Dreiecke durch. Diese Funktion wurde von [73] übernommen.

`pcm_creaispo` erzeugt die Schnittpolygonzüge der Kontaktflächen.

`pcm_eudisqr` berechnet das Quadrat des Euklidischen Abstands zweier Punkte im Raum.

`pcm_swpvec` vertauscht die Koordinaten zweier Vektoren.

`pcm_cractelm` erzeugt die aktiven Flächenelemente der Kontaktflächen.

`pcm_addifip` fügt einem Schnitt ein geschnittenes Dreieck hinzu.

`pcm_addactid` fügt einem Binärbaum aktiver Dreiecke oder Eckpunkte eine Kennung hinzu.

`pcm_srchmvip` sucht nach zugehörigen Schnittpolygonzügen eines Schnitts und fügt sie ihm ggf. hinzu.

`pcm_disinsec` bestimmt auf einer Kante einer Kontaktfläche den Abstand eines Punkts zum am nächsten gelegenen, auf der inneren Seite des Schnittvolumens liegenden Schnittpolygonzug.

`pcm_srchinae` sucht rekursiv nach aktiven Flächenelementen innerhalb von Schnittpolygonzügen.

`pcm_linactid` überträgt die Kennungen der aktiven Flächenelemente vom temporären Binärbaum zu den globalen Datenobjekten.

`pcm_crcemasl` bestimmt die Kontaktelemente nach der Master–Slave–Methode.

`pcm_calcntel` bestimmt ein Kontaktelement nach der Master–Slave–Methode.

`pcm_calctfrc` bestimmt den Kontaktkraftvektor und das resultierende Moment.

`pcm_frcntmem` gibt den temporären Arbeitsspeicher der Kontaktanalyse frei.

pcm_freemem

```
pcm_freemem
|
+-pcm_frcntmem
|
+-(pcm_frbvtmem)
```

`pcm_frcntmem` gibt den temporären Arbeitsspeicher der Kontaktanalyse frei.

`pcm_frbvtmem` gibt den Arbeitsspeicher einer Hüllvolumenhierarchie frei

Globale Daten

`pcm_surfaces.h` enthält Datenstrukturen und -objekte der Kontaktflächen.

`pcm_contacts.h` enthält Datenstrukturen und -objekte der Kontaktanalyse.

Sonstiges

`pcm_mathlib.h` enthält elementare mathematische Funktionen.

`pcm_platform.h` enthält plattformspezifische Substitutionen für die Anbindung an mit FORTRAN implementierte Programme.

Literaturverzeichnis

- [1] W. Trautenberg. *Bidirektionale Kopplung zwischen CAD- und Mehrkörpersimulationssystemen*. VDI Fortschritt-Berichte Reihe 20, Nr. 342, VDI Verlag Düsseldorf, 2001.
- [2] W. Kortüm, P. Lugner. *Systemdynamik und Regelung von Fahrzeugen*. Springer-Verlag Berlin, Heidelberg, 1994.
- [3] W. Rulka. *Effiziente Simulation der Dynamik mechatronischer Systeme für industrielle Anwendungen*. Dissertation, Institut für Mechanik, Technische Universität Wien, 1998.
- [4] A. Sjö. *Numerical aspects in contact mechanics and rolling bearing simulation*. Dissertation, Department of Computer Science, Lund University, 1996.
- [5] J. Butterfass. *Eine hochintegrierte multisensorielle Vier-Finger-Hand für Anwendungen in der Servicerobotik*. Dissertation, Shaker Verlag Aachen, 2000.
- [6] F. Pfeiffer, J. Steuer, Th. Rossmann. Legged walking machines. In *Autonomous robotic systems*, Control and Information Sciences 236, Springer-Verlag London, 1998.
- [7] H. Pacejka. *Tyre mechanics and vehicle dynamics*. Butterworth-Heinemann, 2002.
- [8] H. Netter. *Rad-Schiene-Systeme in differential-algebraischer Darstellung*. VDI Fortschritt-Berichte Reihe 12, Nr. 352, VDI Verlag Düsseldorf, 1998.
- [9] S. Lehner. *3D-Simulation des menschlichen Kniegelenks*. Diplomarbeit, Fachbereich Feinwerk- und Mikrotechnik, Fachhochschule München, 1995.

- [10] A. Sporrer. *Grundlagen und Modellierung eines biomechanischen Mehrkörpersystems des Menschen zur Computersimulation von Bewegungsabläufen bei rechtsmedizinischen Fragestellungen*. Dissertation, Institut für Rechtsmedizin, Universität München, 2000.
- [11] C. Glocker. *Dynamik von Starrkörpersystemen mit Reibung und Stößen*. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, 1995.
- [12] T. Klisch. *Kontaktmechanik in Starrkörpersystemen*. Dissertation, Shaker Verlag Aachen, 1997.
- [13] H. Hertz. Über die Berührung fester elastischer Körper. In *Journal für reine und angewandte Mathematik*, 92:156–171, 1882.
- [14] A.E.H. Love. *A treatise on the mathematical theory of elasticity*. Dover Publications New York, 1927.
- [15] P.D. Panagiotopoulos. A nonlinear programming approach to the unilateral contact and friction–boundary value problem in the theory of elasticity. In *Ingenieur Archiv*, 44:421–432, 1975.
- [16] J.M. de Mul, J.J. Kalker, B. Fredriksson. The contact between arbitrarily curved bodies of finite dimensions. In *Journal of tribology*, 108:140–148, 1986.
- [17] J.J. Kalker. *Three–dimensional elastic bodies in rolling contact*. Kluwer Academic Publishers, 1990.
- [18] *SIMPACK Reference*, Intec GmbH Weßling, 2003.
- [19] *DADS contact modeling guide*, LMS CADSI Leuven, Belgien, 1997.
- [20] S.O. Hartmann. *Simulation von Mehrkörpersystemen mit Kontakten komplexer Geometrie*. Dissertation, Shaker Verlag Aachen, 2002.
- [21] S. Goyal, E.N. Pinson, F.W. Sinden. Simulation of dynamics of interacting rigid bodies including friction I: General problem and contact model. In *Engineering with computers*, 10:162–174, 1994.
- [22] S. Goyal, E.N. Pinson, F.W. Sinden. Simulation of dynamics of interacting rigid bodies including friction II: Software system design and implementation. In *Engineering with computers*, 10:175–195, 1994.

- [23] R.A. LaFarge, C. Lewis. Contact force modeling between non convex objects using a nonlinear damping model. In *Proceedings of international ADAMS user conference*, Ann Arbor, Michigan, 1998.
- [24] C.A. Tenaglia, D.E. Orin, R.A. LaFarge, C. Lewis. Toward development of a generalized contact algorithm for polyhedral objects. In *Proceedings of the IEEE international conference on robotics and automation*, Detroit, Michigan, 1999.
- [25] P. Eberhard. *Kontaktuntersuchungen durch hybride Mehrkörpersystem / Finite Elemente Simulationen*. Habilitationsschrift, Shaker Verlag Aachen, 2000.
- [26] A. Joukhadar, A. Wabbi, C. Laugier. Fast contact localisation between deformable polyhedra in motion. In *Proceedings of the IEEE computer animation conference*, Genf, 1996.
- [27] G. Hippmann. Messung der Relativkinematik in SIMPACK. Interner Bericht, Deutsches Zentrum für Luft- und Raumfahrt, Oberpfaffenhofen, 2001.
<http://www.pcm.hippmann.org/doc/relativmessung.pdf>
- [28] N. Orlandea. *Development and application of node-analogous sparsity-oriented methods for simulation of mechanical dynamic systems*. PhD Thesis, University of Michigan, 1973.
- [29] G. Leister. *Beschreibung und Simulation von Mehrkörpersystemen mit geschlossenen kinematischen Schleifen*. VDI Fortschritt-Berichte Reihe 11, Nr. 167, VDI Verlag Düsseldorf, 1992.
- [30] R.E. Roberson, R. Schwertassek. *Dynamics of multibody systems*. Springer-Verlag Berlin, 1988.
- [31] R. von Schwerin. Multibody system simulation — Numerical methods, algorithms and software. In *Lecture notes in computational science and engineering*, 7, Springer-Verlag Berlin, Heidelberg, 1999.
- [32] W. Schiehlen. *Technische Dynamik*. Teubner-Verlag Stuttgart, 1986.
- [33] P. Lugner, H. Springer, K. Desoyer. *Dynamik von Mehrkörpersystemen*. Skriptum zur Vorlesung, Technische Universität Wien, 2000.
- [34] F. Pfeiffer. *Einführung in die Dynamik*. Teubner-Verlag Stuttgart, 1992.

- [35] H. Brandl, R. Johanni, M. Otter. A very efficient algorithm for the simulation of robots and similar multibody-systems without inversion of the mass-matrix. In *Proceedings IFAC/IFIP international symposium on theory of robots*, Springer-Verlag Wien, 1986.
- [36] W. Rulka. Bewegungsgleichungen für hybride mechanische Systeme mit freiheitsgradproportionalem Rechenaufwand. Interner Bericht 515-86/12, Deutsches Zentrum für Luft- und Raumfahrt, Oberpfaffenhofen, 1986.
- [37] E. Eich-Soellner, C. Führer *Numerical methods in multibody dynamics*. Teubner-Verlag Stuttgart, 1998.
- [38] U.M. Ascher, L.R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM Philadelphia, 1998.
- [39] M. Arnold. *Numerische Verfahren in der Mehrkörperdynamik*. SIM-PACK Academy Seminarunterlagen, Intec GmbH Weßling, 2004.
- [40] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. In *Computer Methods in Applied Mechanics and Engineering*, 1:1-16, 1972.
- [41] C.W. Gear, B. Leimkuhler, G.K. Gupta. Automatic integration of Euler-Lagrange equations with constraints. In *J. Comp. Appl. Math.*, 12&13:77-90, 1985.
- [42] M. Gipser. *Systemdynamik und Simulation*. Teubner-Verlag Stuttgart, 1999.
- [43] A. Fuchs. *Korrekturiteration für implizite Zeitintegrationsverfahren in der Mehrkörperdynamik*. Diplomarbeit, Fakultät für Mathematik, Technische Universität München, 2002.
- [44] E. Hairer, G. Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*. Second edition, Springer-Verlag, 1996.
- [45] K.E. Brenan, S.L. Campbell, L.R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. Elsevier New York, 1989.
- [46] M. Otter. *Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter*. VDI Fortschritt-Berichte Reihe 20, Nr. 147, VDI Verlag Düsseldorf, 1995.

- [47] K. Magnus, H.H. Müller. *Grundlagen der Technischen Mechanik*. Teubner-Verlag Stuttgart, 1990.
- [48] D.A. Hills, D. Nowell, A. Sackfield. *Mechanics of elastic contacts*. Butterworth-Heinemann, 1993.
- [49] H.R. Schwarz. *Methode der finiten Elemente*. Teubner Verlag Stuttgart, 1984.
- [50] O.C. Zienkiewicz, R.L. Taylor. *The finite element method*. Butterworth-Heinemann, 2000.
- [51] K.J. Bathe. *Finite element procedures*. Prentice Hall New Jersey, 1995.
- [52] J. Yao, A. Salo, M. Barbu-McInnis, A. Lerner. Finite element modeling of knee joint contact pressures and comparison to magnetic resonance imaging of the loaded knee. In *Proceedings of the International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, Washington D.C., 2003.
- [53] K.L. Johnson. *Contact mechanics*. Cambridge University Press, 1985.
- [54] J.J. Kalker, Y. van Randen. A minimum principle for frictionless elastic contact with application to non-hertzian half-space contact problems. In *Journal of engineering mathematics*, 6(2):193–206, 1972.
- [55] F. Pfeiffer, C. Glocker. *Multibody dynamics with unilateral contacts*. Wiley&Sons New York, 1996.
- [56] D.W. Marhefka, D.E. Orin. A compliant contact model with nonlinear damping for simulation of robotic systems. In *IEEE transactions on systems, man and cybernetics*, 29(6):566–572, 1999.
- [57] F. Pfeiffer, P. Wolfsteiner. Relative kinematics of multibody contacts. In *Proceedings of the International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, Dallas, Texas, 1997.
- [58] B. Simeon, C. Führer, P. Rentrop. Differential algebraic equations in vehicle system dynamics. In *Survey on Mathematics in Industry*, 1:1–37, 1991.
- [59] A. Tasora, P. Righettini. Sliding contact between freeform surfaces. In *Proceedings of EUROMECH 427: Computational techniques and applications in nonlinear dynamics of structures and multibody systems*, Cachan, France, 2001.

- [60] D. Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. In *Computer Graphics*, 24(4):19–28, 1990.
- [61] A. Curnier, Q.C. He, A. Klarbring. Continuum mechanics modelling of large deformation contact with friction. In *Proceedings of the contact mechanics international symposium*, 145–158, Plenum Press New York, 1995.
- [62] *AC3D v3.6 user manual*, Inivis Ltd., Ely, Cambs, United Kingdom, 2003. <http://www.ac3d.org/>
- [63] P.M. Hubbard. *Collision detection for interactive graphics applications*. PhD Thesis, Brown University, 1995.
- [64] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. In *IEEE Transactions on Visualization and Computer Graphics*, 4(1), 1998.
- [65] J.T. Klosowski. *Efficient collision detection for interactive 3D graphics and virtual environments*. PhD thesis, State University of New York, 1998.
- [66] G. Zachmann. *Exact and fast collision detection*. Diplomarbeit, Fachgebiet Praktische Informatik, Technische Universität Darmstadt, 1994.
- [67] G. Zachmann. Rapid collision detection by dynamically aligned DOP-trees. In *Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS)*. Atlanta, Georgia, 1998.
- [68] G. Zachmann. *Virtual reality in assembly simulation — Collision detection, simulation algorithms and interaction techniques*, Kap. 3. Dissertation, Fachbereich Informatik, Technische Universität Darmstadt, 2000.
- [69] G. van den Bergen. A fast and robust GJK implementation for collision detection of convex objects. In *Journal of graphics tools*, 4(2):7–25, 1999. <http://www.win.tue.nl/~gino/solid/>
- [70] S. Gottschalk, M.C. Lin, D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. Technical report TR96-013, Department of computer science, University of N. Carolina. In *Proceedings of ACM SIGGRAPH*, 1996. <http://www.cs.unc.edu/~geom/OBB/OBBT.html>

- [71] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowrizal, K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. In *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.
<http://www.ams.sunysb.edu/~jklosow/quickcd/QuickCD.html>
- [72] P. Eberhard, S. Jiang. Collision detection for contact problems in mechanics with a boundary search algorithm. In *Mathematical modelling of systems*, 3, 4, 1997.
- [73] T. Möller. A fast triangle–triangle intersection test. In *Journal of graphics tools*, 2(2):25–30, 1997.
<http://www.acm.org/jgt/papers/Moller97/>
- [74] *PCM Webseite*: <http://www.pcm.hippmann.org>
- [75] G. Hong. *Virtual-Reality Methoden für ein MKS-Kontaktmodell*. Diplomarbeit, Lehrstuhl für Angewandte Mechanik, Technische Universität München, 2003.
- [76] G. Turk, J.F. O’Brien. Variational implicit surfaces. Technical Report GIT-GVU-99-15, Graphics, Visualization, and Useability Center, Georgia Institute of Technology, 1999.
- [77] G. Yngve, G. Turk. Creating smooth implicit surfaces from polygonal meshes. Technical Report GIT-GVU-99-42, Graphics, Visualization, and Useability Center, Georgia Institute of Technology, 1999.
- [78] H.Q. Dinh, G. Turk, G. Slabaugh. Reconstructing surfaces using anisotropic basis functions. In *International Conference on Computer Vision*, 606–613, Vancouver, 2001.
- [79] H.Q. Dinh, G. Turk, G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371, 2002.
- [80] G. Turk, H.Q. Dinh, J.F. O’Brien, G. Yngve. Implicit surfaces that interpolate. In *ACM Transactions on Graphics*, 21(4):855–873, 2002.
- [81] G. Turk, J.F. O’Brien. Modelling with implicit surfaces that interpolate. In *ACM Transaction on Graphics*, 21(4):855–873, 2002.
- [82] G. Turk, J.F. O’Brien. Shape transformation using variational implicit functions. In *Proceedings of ACM SIGGRAPH*, 335–342, 1999.

- [83] W.E. Lorensen, H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of ACM SIGGRAPH*, 21(4):163–169, 1987.
- [84] G.M. Treece, R.W. Prager, A.H. Gee. Regularised marching tetrahedra: Improved iso-surface extraction. In *Computers and Graphics*, 23(4):583–598, 1999.
- [85] A. Hilton, A.J. Stoddart, J. Illingworth, T. Windeatt. Building 3D graphical models of complex objects. 14th Eurographics Conference, Vol. 1, 193–204, London, 1996.
- [86] A. Hilton, A.J. Stoddart, J. Illingworth, T. Windeatt. Marching triangles: Range image fusion for complex object modelling. IEEE International Conference on Image Processing, Vol. II, 381–384, Lausanne, 1996.
- [87] A. Hilton, J. Illingworth. Marching triangles: Delaunay implicit surface triangulation. CVSSP Technical Report, Vision, Speech and Processing Group, Department of Electronic and Electrical Engineering, University of Surrey, Guildford, 1997.
- [88] E. Hartmann. A marching method for the triangulation of surfaces. In *The Visual Computer*, 14:95–108, 1998.
- [89] T. Karkanis, A.J. Stewart. High quality, curvature dependent triangulation of implicit surfaces. In *IEEE Computer Graphics and Applications*, 22(2):60–69, 2001.
- [90] *3D Cafe Webseite*: <http://www.3dcafe.com>
- [91] *Meta3D Webseite*: <http://www.exchange3d.com>
- [92] A.C. Hindmarsh. LSODE and LSODI, two new initial value ordinary differential equation solvers. *ACM-SIGNUM Newsletters*, 15:10–11, 1980.
- [93] D.A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons Inc., 1990.
- [94] H.–B. Schmiedmayer. *Unfallbiomechanik*. Skriptum zur Vorlesung, Technische Universität Wien, 1996.
- [95] E. Hairer, S.P. Nørsett, G. Wanner. *Solving ordinary differential equations I: Nonstiff problems*. Second revised edition, Springer-Verlag, 1993.

- [96] C.T. Kelley. *Solving nonlinear equations with Newton's method*. SIAM Philadelphia, 2003.
- [97] S. Ebrahimi. Polygonal contact model for elastic bodies. Zwischenbericht ZB-140, Institut B für Mechanik, Universität Stuttgart, 2004.

Gerhard Hippmann

Zur Person

Geburtsdatum	8. August 1968
Geburtsort	Augsburg
Familienstand	ledig
Staatsangehörigkeit	deutsch
Anschrift	Muldenstraße 14 d D-82237 Steinebach
Telefon/Fax	+49 700 00447743
E-mail	pcm@hippmann.org

Ausbildungs- und Berufsweg

09/1975 – 08/1979	Schrader-Volksschule Kaufbeuren
09/1979 – 05/1989	Staatliches Gymnasium Kaufbeuren
05/1989	Allgemeine Hochschulreife
06/1989 – 08/1991	Wehrdienst
10/1991 – 08/1996	Technische Universität München Studiengang Maschinenwesen Fachrichtung Konstruktion und Entwicklung
09/1996	Diplom-Ingenieur Univ.
09/1996 – 12/1996	Technische Universität München Lehrstuhl und Institut für Landmaschinen Wissenschaftlicher Mitarbeiter
05/1997	Max-Eyth-Nachwuchsförderpreis
02/1997 – 07/1998	Intec GmbH Oberpfaffenhofen Projektingenieur
07/1998 – 05/2003	Deutsches Zentrum für Luft- und Raumfahrt Oberpfaffenhofen Abteilung Fahrzeug-Systemdynamik Wissenschaftlicher Mitarbeiter
seit 06/2003	Intec GmbH Oberpfaffenhofen Methodenentwicklung und -implementierung