

CoRDE:
Cosserat Rod Elements for the
Animation of Interacting Elastic Rods

Dissertation

zur Erlangung des Doktorgrades der Fakultät
für Angewandte Wissenschaften der Albert-
Ludwigs-Universität Freiburg im Breisgau

vorgelegt von

Jonas Spillmann

September 2008

1. Gutachter
2. Gutachter
Dekan

Prof. Dr. Matthias Teschner
Prof. Dr. Gábor Székely
Prof. Dr. Bernhard Nebel

Datum der Promotion **29. August 2008**

Abstract

The physically-based simulation and animation of one-dimensional deformable objects, also known as *elastic rods*, is an important research field in both mechanics and computer graphics. One-dimensional deformable objects are characterized in having the shape of a curve in space, where the radius is small compared to the length. Consequently, elastic rods have large global deformations, even if the local strains are small. In computer graphics, they represent e. g. ropes, threads or hair strands. Due to the mandatory rotational degrees-of-freedom and due to their negligible volume, the simulation and contact handling of elastic rods is challenging.

This thesis presents efficient techniques for the physically plausible animation of elastic rods. Two main aspects are addressed, notably the modeling of the deformations, and the handling of collisions. The goal is to dynamically simulate elastic rods in contact at interactive rates.

First, it is investigated into a deformation model for elastic rods with material torsion. If material torsion is considered, then the orientation degrees-of-freedom of the rod must be carried along the simulation. Since the centerline is explicitly represented, an accurate contact handling is enabled. The deformation model is able to reproduce the characteristic buckling and looping phenomena at interactive rates.

If the simulation of knots is considered, then it is observed that a high mechanical accuracy is required to represent the knot. In contrast, the computational effort should be minimized in the undeformed parts of the rod. Consequently, the thesis investigates an adaptive technique for elastic rods. To improve the efficiency and to avoid post-stabilization, a novel variational subdivision approach is employed that requires the solve of a non-linear system of equations.

When addressing the contact handling, elastic rods share similarities with both rigid and volumetric deformable objects. Therefore, an approach to handle contacts of interacting volumetric deformable objects is presented. The approach maintains the non-penetration constraints and computes the contact forces directly, without employing an iterative technique. Thereby, the problems inherent to both penalty approaches and analytical methods are overcome.

The approach for volumetric deformable objects is then extended to handle elastic rods in contact. Since elastic rods are assumed to have a rigid cross-section, a global scheme is mandatory. An iterative technique solves the non-linear system of equations. Further, the proposed approach approximates Coulomb friction and

enables the simulation of complex knots at interactive rates.

Additional contributions have been achieved in the field of spatial discretization of objects and in the field of collision detection. An approach is presented that produces well-shaped tetrahedral meshes from arbitrary input surfaces, thereby avoiding any surface pre-processing. Moreover, an approach to detect collisions between geometrically deformable objects is presented. The performance of the approach depends only on the number of colliding primitives, without requiring to visit each deformed primitive.

Zusammenfassung

Die physikalisch basierte Simulation von eindimensionalen verformbaren Objekten ist ein wichtiges Forschungsgebiet in der Mechanik und der Computergraphik. Eindimensionale verformbare Objekte haben die Form einer Kurve im Raum, wobei der Radius klein ist im Vergleich zur Länge. Daher haben eindimensionale verformbare Objekte im Allgemeinen grosse globale Verformungen, obschon die lokalen Spannungen klein sind. In der Computergraphik werden diese Objekte gebraucht, um Seile, Fäden oder Haarsträhnen zu repräsentieren. Wegen den Rotationsfreiheitsgraden und wegen des vernachlässigbar kleinen Volumens gelten eindimensional verformbare Objekte als schwierig zu simulieren.

Diese Dissertation beschäftigt sich mit effizienten Techniken, um eindimensional verformbare Objekte physikalisch plausibel zu simulieren. Zwei Hauptaspekte werden adressiert, nämlich die Modellierung der Verformung und die Kontaktbehandlung. Das Ziel ist die interaktive dynamische Simulation von diesen Objekten.

Zuerst wird ein Verformungsmodell für eindimensional verformbare Objekte erforscht, welches die Materialverdrehung unterstützt. Dies bedingt, dass die Orientierungsfreiheitsgrade zusätzlich simuliert werden müssen. Die explizite Simulation der Kontrollpunkte erleichtert eine genaue Kontaktbehandlung. Das Verformungsmodell kann die charakteristischen Phänomene, welche die Verformung von eindimensionalen verformbaren Objekten auszeichnen, in interaktiven Raten simulieren.

Für die Simulation von Knoten ist eine hohe mechanische Genauigkeit notwendig, um den Knoten zu repräsentieren. Im Gegensatz dazu soll der Berechnungsaufwand in den nicht-verformten Bereichen minimiert werden. Deshalb widmet sich die Dissertation einer adaptiven Technik. Um die Effizienz weiter zu erhöhen und gleichzeitig Nachstabilisierung zu vermeiden, wird ein neuartiger Ansatz basierend auf der Variationsrechnung verwendet, der das Lösen eines nicht-linearen Gleichungssystems erfordert.

Im Kontext von der Kollisionsbehandlung wird beobachtet, dass eindimensional verformbare Objekte Ähnlichkeiten mit sowohl Starrkörpern als auch volumetrisch verformbaren Objekten haben. Deshalb wird zuerst ein Ansatz präsentiert, der Kontakte zwischen volumetrischen Objekten behandelt. Der Ansatz hält die Nichteindringungsbedingungen bei und berechnet die Kontaktkräfte direkt. Dadurch werden die Probleme überwunden, unter welchen Penalty- und iterative An-

sätze leiden.

Der Ansatz wird danach auf die Kontaktbehandlung von eindimensional verformbaren Objekten erweitert. Weil der Querschnitt als starr angenommen wird, ist ein globales Schema notwendig. Eine iterative Technik wird angewendet, um das nichtlineare Gleichungssystem zu lösen. Der Ansatz approximiert zudem Coulomb-Reibung und ermöglicht die Simulation von komplexen Knoten in interaktiven Raten.

Weitere Beiträge wurden in den Bereichen der räumlichen Diskretisierung von Objekten und im Bereich der Kollisionserkennung gemacht. Ein Ansatz wird präsentiert, der wohlgeformte Tetraedernetze von beliebigen Oberflächen berechnet. Dabei wird eine Vorverarbeitung von den Oberflächen vermieden. Überdies wird ein Ansatz präsentiert, der Kollisionen zwischen geometrisch verformbaren Objekten erkennt. Die Komplexität des Ansatzes hängt von der Anzahl der kollidierten Primitiven ab, und nicht von der Anzahl der verformten Primitiven.

Acknowledgements

First of all, I would like to thank my advisor Prof. Dr. Matthias Teschner for giving me the opportunity to pursue my Ph. D. in Computer Graphics at the University of Freiburg. His support and his expertise throughout the years were invaluable and it has always been a great pleasure to work with him.

Second, I would like to thank Prof. Dr. Gábor Székely for reviewing my thesis. I would also like to thank the members of the Ph. D. committee at the University of Freiburg for their support and their interest in my research.

Additionally, I would like to thank my fellow colleagues at the lab for giving me good and critical feedback to my papers, for helping me through the maths, and for sorting out the nasty pitfalls of the Ogre rendering engine: Markus Becker, Marc Gissler, Rüdiger Schmedding and Markus Ihmsen. Moreover, I would like to thank Cynthia Findlay for helping me finding the way through all the administrative paper works and Stefan Teister for spending hours in fixing the computer-related problems at the lab.

Many thanks go to Bruno Heidelberger who has advised my diploma thesis at ETH Zürich and thereby paved my way to computer graphics. Also, I thank Matthias Müller and Denis Steinemann for giving me good inputs and fruitful discussions.

Finally, I would like to thank my family for giving me the support in the hard days during the three years at Freiburg. And especially, I would like to thank my friends who accompanied me on many climbing trips. After all, you are responsible for my interest in rope modeling!

Contents

1	Introduction	1
1.1	Context	2
1.2	Contributions	3
1.3	Publications	5
1.4	Thesis outline	6
2	Related work	7
2.1	Background	7
2.2	Deformation modeling	8
2.2.1	Volumetric deformation models	9
2.2.2	Non-volumetric deformation models	10
2.2.3	Adaptive methods	15
2.3	Contact handling	18
2.3.1	Rigid object collision response	19
2.3.2	Volumetric deformable object collision response	21
2.3.3	Non-volumetric deformable object collision response	24
3	CORDE: Elastic rod modeling	27
3.1	Cosserat theory of elastic rods	29
3.1.1	Notation	30
3.1.2	Representation	30
3.1.3	Constitutive relations	32
3.2	The CORDE model	33
3.2.1	Discretization	33
3.2.2	Parametrization of the rotation matrix	34
3.2.3	Finite element model	36
3.3	The dynamic evolution	37
3.3.1	Numerical time-integration	38
3.3.2	Modeling internal friction	39
3.3.3	Benefits, limitations and future work	40
3.4	Cosserat nets	41
3.4.1	Theoretical background	42
3.4.2	Approach	42

3.5	Reference deformation model	43
3.5.1	Parametrization of the rotation matrix	44
3.5.2	Finite element model	45
3.5.3	Implementation notes	46
3.6	Results	47
3.6.1	Qualitative evaluation	47
3.6.2	Quantitative evaluation	49
3.6.3	Application	52
3.7	Discussion	55
4	Adaptive variational simulation	59
4.1	Approach	61
4.1.1	Deformation model	61
4.1.2	Adaptation criteria	61
4.1.3	Element refinement	62
4.1.4	Element unrefinement	64
4.1.5	Remarks	64
4.2	Adaptive CORDE	64
4.3	Results	65
4.4	Discussion	67
5	Volumetric object contact handling	69
5.1	Contact mechanics	71
5.1.1	Principles	71
5.1.2	Frictional contacts	72
5.1.3	Deformable contact mechanics	73
5.2	Approach	75
5.2.1	Object representation	75
5.2.2	Predictor-corrector simulation	75
5.2.3	Terms	76
5.2.4	Overview	77
5.2.5	Contact forces	78
5.2.6	Mathematical justification	80
5.2.7	Frictional contacts	83
5.2.8	Force post-correction	84
5.3	Results	85
5.4	Discussion	87
6	Elastic rod contact handling	91
6.1	Approach	92
6.1.1	Object representation	93
6.1.2	Predictor-corrector simulation	93
6.1.3	Constraint-based contact forces	94
6.1.4	Iterative solution	95

6.1.5	Friction	96
6.1.6	Contact forces	96
6.1.7	Contact handling in the adaptive simulation	97
6.2	Results	97
6.3	Discussion	98
7	Conclusion	101
7.1	Outlook	103
A	Spatial discretizations	105
A.1	Related work	107
A.2	Approach	109
A.3	Results	114
A.4	Discussion	117
B	Output-sensitive collision detection	119
B.1	Related work	120
B.2	Preliminary theory	122
B.3	Approach	124
B.4	Results	130
B.5	Discussion	132
C	Quaternions	135
C.1	Quaternion representation	135
C.2	Quaternion algebra	136
C.3	Quaternions and rotations	136
C.4	Quaternion conversions	137
C.5	Derivation of strain rates in terms of quaternions	139
D	Non-linear CG method	141
D.1	Principles	141
D.2	Line search	142

Chapter 1

Introduction

There is an ongoing interest in bringing the real world into the computer. The goal is to build a virtual environment on the computer, including virtual objects that behave similar to their real counterparts. To accomplish this, the physical equations that govern the statics and dynamics of the objects have to be discretized and solved accordingly. In computer graphics, the goal is often to provide the necessary visual realism, without spending too much in time in the accurate mechanical simulation.

In the context of the physical simulation of *deformable solid objects*, the static and dynamic behavior of the objects is commonly described by the mechanical laws investigated in the field of elasticity and contact mechanics. By numerically solving the resulting equations, the trajectories of the objects are obtained. Three research areas are distinguished in the simulation of deformable solids: To compute the continuously changing shape of an object under external load, a *deformation model* has to be employed. Given the state of the geometry and given the external load, the deformation model computes the forces that work against the deformation and restore the initial shape. While the static simulation is used to describe the resting state of the object, the dynamic simulation captures the motion of the object in time. As long as the object is considered in isolation, these trajectories vary continuously in time. However, as soon as multiple objects come into play, the interactions between these objects have to be modeled. To check whether two objects interfere, a *collision detection* scheme has to be employed. If a collision is reported, the motion paths of the incorporated objects have to be varied accordingly such that interpenetrations are avoided. The computation of the contact forces that maintain the non-penetration constraints is governed by a *collision response* model.

In the context of solid object simulation, three *object types* can be distinguished, notably three-dimensional, two-dimensional and one-dimensional objects. Here, these terms refer to the spatial extent of the object in its local frame. Although these objects resemble each other in the microscopic point of view, they differ significantly in their macroscopic behavior. Consequently, for each object type, an appropriate discretization has to be chosen. Then, the deformation can be modeled accordingly.

For *three-dimensional* objects, the length, the width and the height are commonly assumed to be in the same magnitude. The volume of the object is commonly discretized into tetrahedral or hexagonal elements. Then, the deformation model considers the geometric configuration of the deformed tetrahedra in order to compute the elastic forces. Three-dimensional objects are also denoted as *volumetric objects*.

In contrast, the thickness of a *two-dimensional* object, also termed *thin shell*, is assumed to be much smaller than its length and width. Hence, the volume of the object is not explicitly represented. Instead, the surface is discretized into two-dimensional elements (in most cases, triangles are chosen). The deformation model then considers the deformed triangles and the angles between adjacent triangles in order to compute the elastic forces. Thin shells are widely used to represent clothing. Likewise, for *one-dimensional* objects, one spatial dimension dominates the other two. Therefore, one-dimensional objects are commonly discretized into line segments, and the deformation model considers the configuration between consecutive segments. Two- and one-dimensional objects are also denoted as *non-volumetric objects*.

Depending on the application, the focus of the simulation is either on the accuracy or alternatively on the efficiency. For example, if the elastic forces of a suspension bridge are considered, then accumulating numerical errors cannot be tolerated. Instead, computing times of hours or days are acceptable. Such accurate simulation methods are well-investigated in mechanics. In contrast, if the physically-based *animation* of a suspension bridge in a computer game is considered, then the elastic forces must be computed at interactive rates. In computer graphics, such efficient physically-based simulation methods are well-investigated for three- and two-dimensional objects. In contrast, efficient and physically plausible methods for one-dimensional objects still provide room for investigations.

1.1 Context

In this thesis, the focus is on the physically plausible simulation and animation of one-dimensional deformable objects. Deformable objects whose configuration space is one-dimensional play an important role in many research fields, e. g. in engineering to represent cables or tubes, or in microbiology to represent DNA structures. In computer animation, one-dimensional objects represent threads, ropes, cables, sutures or hair strands. In this thesis, these objects are consistently denoted as *elastic rods*, independent of their actual material properties.

Elastic rods are commonly discretized into linear chains of nodes with positions in \mathbb{R}^3 , joined by line segments. However, what makes the simulation of elastic rods particularly challenging is the fact that the elastic forces cannot be computed from those nodes alone. Instead, the orientation of the line segments has to be represented in addition. To see why, consider e. g. a straight rope whose one end is twisted around the main axis. Although the nodes are not displaced, the resulting

shearing deformation induces elastic forces. Consequently, additional degrees-of-freedom (DOFs) have to be introduced that express the orientation of the elastic rod, and therefore allow to model the *material torsion*. Moreover, the equations that relate the material torsion to bending and stretching deformations have to be identified. The corresponding problems are challenging, especially if the dynamic evolution of the elastic rod is considered. This thesis investigates into a novel deformation model for elastic rods. The corresponding diagonalized equations of motion allow for a dynamic simulation of elastic rods at interactive rates.

Elastic rods are characterized in having large global deformations under external load, even if the local strains are small. Consequently, self-collisions occur frequently. However, it has to be considered that the volume of elastic rods is negligible, which implies that the cross-section of elastic rods is rigid. Thus, the penalty method that relates the contact forces to an interpenetration measure is not a favorable choice. Moreover, if e. g. threads are tied to knots, then the pressures can be arbitrary large. In order to prevent interferences that immediately result in an inconsistent state, a global collision response approach is often the only admissible choice. Again, solving the resulting system of equations offers a challenging playground for investigations.

1.2 Contributions

In this thesis, the main contributions are achieved in the field of deformation modeling, adaptive simulation and contact handling for elastic rods. In addition, some important results in other research areas round out the thesis.

Deformation modeling Existing deformation models for elastic rods have discretized the elastic rod into nodes that are connected with springs. Then, the elastic forces are computed from the stretched springs, and from the angles between consecutive springs. Still, this complicates the consistent treatment of twisting and bending forces [PLK02, WBD*05, KPGF07]. Other deformation models have reconstructed the geometric configuration of the elastic rod by defining the boundary values at the start and end of the rod, and solving the corresponding boundary value problem (BVP). The resulting schemes are difficult to implement though, especially if contacts are considered [Pai02, BAC*06]. In this thesis, CORDE is proposed, which is a novel deformation model for the physically plausible simulation of elastic rods. The deformation model treats bending and twisting deformation in a consistent manner and allows to reproduce the buckling and looping phenomena that characterize the deformation of elastic rods. In contrast to many previous approaches, the simulation of the elastic rod is fully dynamic. Moreover, since a finite element (FE) method is employed to compute the elastic forces, the deformed shape does not depend on the discretization (up to discretization errors) and adaptive methods are facilitated. The resulting equations of motion are decoupled and allow for an efficient solve. Therefore, elastic rods can be simulated at

interactive rates.

Adaptive simulation If the simulation of knots is considered, then it is observed that a high mechanical accuracy is mandatory to accurately reproduce the knot. In contrast, few DOFs are usually sufficient to represent the undeformed parts of the rod. Therefore, it is desirable to arrange the DOFs during the simulation, based on the temporally varying geometric configuration of the rod. In turn, this calls for an adaptive technique. In this thesis, an approach is proposed to dynamically refine an elastic rod. In contrast to previous approaches, a variational technique is employed to compute the positions of the new control points. Thus, any post-stabilization such as damping is avoided.

Contact handling The inherently large global deformation of elastic rods calls for a robust contact and self-contact handling scheme. Since in contrast to thin shells, elastic rods do neither cover a volume nor a surface, interpenetrations cannot be determined with local tests, each configuration is *per se* valid. Consequently, interpenetrations must be avoided, even for large stresses. In turn, this forbids penalty approaches.

Elastic rods share similarities with volumetric deformable objects, because the geometry of elastic rods is subject to large deformations. Still, elastic rods are assumed to have a rigid cross-section, which makes the collision response similar in spirit to rigid object collision response. To account for this twofold nature of elastic rod contact handling, the thesis first proposes a predictor-corrector approach for the contact handling of volumetric deformable objects. In contrast to previous penalty methods, a collision-free configuration is maintained throughout the simulation. In contrast to previous analytical methods, the contact problem is decoupled, which enables a direct and efficient computation of the contact forces.

Later, it is shown that a similar predictor-corrector approach can be employed to compute contact forces for interacting elastic rods. To avoid interpenetrations, the non-penetration constraints are stacked in a system of equations, which is solved iteratively. In contrast to previous approaches, the non-penetration constraints are maintained throughout the simulation, even for large time-steps. In turn, this enables the efficient simulation of complex knots.

Additional contributions The simulation of deformable objects consists of additional mandatory tasks which have to be solved accordingly. In order to simulate deformable objects, the object domains must be discretized accordingly. In many cases, the objects are implicitly defined by the boundary surfaces. Thus, in the appendix, an approach is proposed to spatially discretize the domains bounded by arbitrary input surfaces, where the focus is on the robustness and efficiency. In contrast to previous discretization approaches, the proposed scheme processes also non-manifold and unoriented surfaces, without requiring any pre-processing.

Moreover, in order to carry out the contact handling, the collisions must be known in advance. The geometric problem of the collision detection is often the performance bottleneck of the simulation. In the appendix, an output-sensitive scheme to detect collisions between geometric deformable objects [MHTG05] is proposed. Although this approach has no apparent relation to the simulation of one-dimensional deformable objects, it has nevertheless provided an important step in the learning pipeline of this thesis.

1.3 Publications

This thesis bases on a series of peer-reviewed publications in conference proceedings and journals.

- Basics on the contact surface model for the collision response of deformable objects have been published in [ST05]:
J. Spillmann, M. Teschner, *Contact Surface Computation for Coarsely Sampled Deformable Objects*, Proc. Vision, Modeling and Visualization, 2005.
- The framework to produce tetrahedral meshes from arbitrary triangle soups has been published in [SWT06]:
J. Spillmann, M. Wagner, M. Teschner, *Robust Tetrahedral Meshing of Triangle Soups*, Proc. VMV, 2006.
- The efficient refitting method for bounding sphere hierarchies has been published in [SBT07a]:
J. Spillmann, M. Becker, M. Teschner, *Efficient Updates of Bounding Sphere Hierarchies for Geometrically Deformable Models*, Journal of Visual Communication and Image Representation, 2007.
- The contact model for volumetric deformable objects has been published in [SBT07b]:
J. Spillmann, M. Becker, M. Teschner, *Non-iterative Computation of Contact Forces for Deformable Objects*, Journal of WSCG, 2007.
- The deformation model for one-dimensional deformable objects has been published in [ST07]:
J. Spillmann, M. Teschner, *CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-dimensional Elastic Objects*, ACM SIGGRAPH/Eurographics Symposium of Computer Animation, 2007.
- The adaptive simulation, and the global contact model of one-dimensional deformable objects have been published in [ST08]:
J. Spillmann, M. Teschner, *An Adaptive Contact Model for the Robust Simulation of Knots*, Computer Graphics Forum (Proc. Eurographics), 2008.

- The extension of the CORDE model to networks, and the comparison of CORDE to a reference model are going to be published in [ST09]:
J. Spillmann, M. Teschner, *Cosserat Nets*, IEEE Transactions on Visualization and Computer Graphics, to appear.

1.4 Thesis outline

The remainder of the thesis is organized as follows: In Chapter 2, the related work is discussed. First, the focus of this discussion is on the deformation modeling of elastic rods. In addition, important advances in the fields of volumetric deformation modeling are mentioned. The second part of Chapter 2 discusses the constrained motion of deformable objects. Since elastic rods share similarities with the contact handling of both rigid and deformable objects, both fields will be discussed extensively.

In Chapter 3, CORDE, a novel deformation model for elastic rods, is presented. After a brief introduction into the Cosserat theory of elastic rods, both the static and the dynamic simulation of CORDE are discussed. Further, the deformation model is carefully evaluated and compared to a reference model.

In order to improve the efficiency of the simulation, and in order to improve the mechanical accuracy in certain regions of interest, an adaptive framework for elastic rods is proposed in Chapter 4. The adaptive framework comes with a novel strategy to place the new control points that is governed by some energy functional.

While the former two chapters have been discussing the unconstrained evolution, Chapter 5 moves to the constrained motion. First, the fundamentals of contact mechanics are briefly revised. Then, the predictor-corrector strategy is exemplified with a contact model for volumetric deformable objects. The goal of this chapter is to give the reader an idea on how predictor-corrector strategies can be applied to handle collisions, and how to guarantee the conservation of momentum. In order to generate the meshes for the volumetric deformable objects, a novel approach discussed in Appendix A is employed.

Equipped with this knowledge, the reader is ready to understand the contact model for elastic rods which is presented in Chapter 6. The scheme solves a non-linear system of inequalities with an iterative approach and allows to simulate resting states of stacked rods. Together with the adaptive model presented in Chapter 4, the simulation of knots is enabled.

The thesis concludes with Chapter 7 that summarizes the benefits and limitations of the presented techniques and provides an outlook to future work.

Chapter 2

Related work

The physically-based simulation and animation of elastic rods can be classified into two research areas, notably the deformation modeling and the contact handling. Therefore, in a first part, works related to deformation modeling of elastic rods are discussed. Since elastic rods share similarities with volumetric deformable objects, and since the contact handling approach in Chapter 5 bases on volumetric objects, the thesis also mentions important advances in the field of volumetric deformation modeling.

In the second part, the works related to contact handling of elastic rods are discussed. Since elastic rods are assumed to be incompressible but deformable, the contact handling shares similarities with rigid, volumetric deformable and non-volumetric deformable objects. Consequently, the contact handling of all three object categories are discussed in a consistent manner.

2.1 Background

The deformation modeling and the computation of the motion paths of the deformable objects are problems related to classical mechanics. Together, they constitute the physically-based simulation of deformable objects. The movement of each mass point of the deformable object is governed by its kinetic and potential energy. That means that the computation of the dynamics of a deformable object can be understood as a variational problem, notably to seek the motion path of each mass point that minimizes an integral expression. The variational formulation results in the Lagrangian equation of motion,

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{g}}} - \frac{\partial T}{\partial \mathbf{g}} + \frac{\partial V}{\partial \mathbf{g}} + \frac{\partial D}{\partial \dot{\mathbf{g}}} = \mathbf{f} \quad (2.1)$$

In this equation, T refers to the total kinetic energy of the solid, V refers to the total potential energy of the object, and D refers to the dissipation energy of the object. V sums all potential energies of the object, e. g. its deformation energy or its potential energy with respect to a reference frame. Further, \mathbf{g} refers to the vector

gathering the degrees-of-freedom (DOFs) of the object, and the vector \mathbf{f} collects the external forces that act on the object. By symbolically differentiating the kinetic and the potential energy according to (2.1), the equations of motion are obtained,

$$\mathbf{M}\ddot{\mathbf{g}} - \mathbf{k}(\dot{\mathbf{g}}, \mathbf{g}) = \mathbf{f} \quad (2.2)$$

where \mathbf{M} is the mass matrix, and \mathbf{k} is a function that computes the internal forces, i. e. the elastic forces, the damping forces, and eventually coriolis forces. A recommendable starter into the Lagrangian dynamics is the book of Goldstein [Gol81].

Since years, works have been published that propose accurate and efficient methods to solve the Lagrangian equation of motion. *Deformation models* venture into ways to derive the deformation energy V of a deformable object, given its current configuration \mathbf{g} . Usually, a deformation model gives a relation between the geometric configuration of the discretized object, and the restitution forces that restore the resting configuration of the object.

Simulation methods offer ways to solve the Lagrangian equation of motion. Here, two classes of methods are distinguished: Static methods neglect the kinetic energy of the object and compute the coordinates \mathbf{g} that minimize the potential energy of the object, given a static load \mathbf{f} . In contrast, dynamic methods time-integrate (2.2), thereby obtaining the motion paths $\mathbf{g}(t)$ which constitute the dynamic motion of the object. The initial configuration is usually defined by $\mathbf{g}(0)$. This is accomplished by choosing an adequate *numerical time-integration scheme*. Explicit schemes compute the subsequent coordinates $\mathbf{g}(t+h)$ by a single evaluation of the stiffness function $\mathbf{k}(\dot{\mathbf{g}}(t), \mathbf{g}(t))$ in the current time step t . In contrast, multi-step approaches perform multiple evaluations of the stiffness function at different points in time to improve the accuracy of the integration. An implicit integration schemes compute the subsequent coordinates by solving a linear system, and by employing the gradient $\nabla_{\mathbf{g}}\mathbf{k}$ of the stiffness function. For implicit schemes, the stiffness function must be linearized. Consequently, for implicit schemes, the deformation modeling cannot be considered in isolation from the underlying numerical time-integration method. Details on the various numerical integration schemes are found in e. g. the book of Schwarz [Sch97b].

If constraints are considered, then an appropriate *constraint technique* has to be employed which computes motion paths $\mathbf{g}(t)$ that satisfy the constraints. This is accomplished by e. g. augmenting (2.1) with Lagrange multipliers, by considering reduced coordinates or by computing constraint forces that constitute additional external forces. The contact handling constitutes a special case of constrained simulation, where the contact forces maintain the time-varying non-penetration constraints.

2.2 Deformation modeling

The deformation model depends on the type of the object which is modeled, i. e. whether it has a volumetric or a non-volumetric shape. For both types, there exists

a large variety of modeling methodologies.

2.2.1 Volumetric deformation models

In the field of mechanics, the deformation modeling of volumetric deformable objects is well-investigated, but these approaches often study either the static deformation of the elastic object under load, or they investigate into the vibration modes by analyzing the mass-matrix. The efficient computation of the dynamics of deformable objects is of minor interest. In contrast, in the field of computer graphics, the goal is often to plausibly reproduce the dynamics of deformable objects at the best-possible efficiency. Here, the approaches can be grouped into physically-based deformation models, and non-physical deformation models.

Physically-based deformation modeling Physically-based deformation models are motivated by the elasticity theory and continuum mechanics. The goal is to reproduce the physically plausible behavior of deformable objects. Terzopoulos *et al.* [TPBF87] have been among the first to investigate into physically-based deformation modeling. In order to compute the elastic forces, *finite element deformation models* have become the quasi-standard to model the static equilibria [GTT89, BNC96] and the dynamic evolution [OH99, OBH02, DDCB01, WDGT01] of deformable volumetric objects. Since the first-order approximation of the Green's strain tensor works only well for small deformations, co-rotational formulations have gained increasing interest [CGC*02b, MDM*02, MG04]. Moreover, large deformations and geometric changes such as cuttings may result in badly-shaped tetrahedra, which has been addressed in [ITF04, BWHT07, WBG07]. Alternatively, the domain can be re-meshed frequently to avoid the sliver tetrahedra [WT08]. This issue is discussed later in Sec. 2.2.3. Instead of modeling the volumetric deformation, it is as well possible to model the surface deformation, resulting in a boundary value problem [BNC96, JP99].

In contrast to FE deformation models, the *reduced deformation models* approximate the deformation by a superposition of displacement fields [JP04]. This is usually done by extracting the vibrational modes from the mass-matrix [PW89, HSO03, BJ05]. Reduced deformation models allow for an efficient simulation on the GPU, as proposed in [JP02].

Besides mesh-based methods, there exist a couple of approaches that simulate a particle cloud and compute forces based on the neighborhood-relationship [DG95, MKN*04, PPG04]. Particle-based methods can also be combined with FE methods [SSIF07].

Non-physical deformation modeling In many cases, the goal is not the physically-based modeling of the deformations, but an efficient and simple deformation model that results in a plausible deformation behavior. Such deformation models constitute the class of non-physical deformation models that include mass-spring deformation models and geometric deformation models.

In *mass-spring deformation modeling*, the volume is discretized into particles that are linked by springs, where the material behavior is governed by the spring constants. Mass-spring models have been employed to compute facial and muscle deformations [PB81] or to animate melting objects [TPF89]. The original mass-spring approach has been extended by Teschner *et al.* by additionally considering volume- and surface-preservation potentials [THMG04], and can be computed on the GPU [GEW05].

The class of *geometric deformation models* is basically made up by two recent publications: In 2005, Müller *et al.* have proposed their meshless deformation model, which computes goal positions based on an estimated global deformation. This allows for an unconditionally stable integration [MHTG05]. This approach has later been enhanced by Rivers and James by reducing the cluster blending artifacts [RJ07]. In Appendix B.2, the approach of Müller *et al.* is detailed.

2.2.2 Non-volumetric deformation models

In this section, deformable objects are considered that have a negligible volume, i. e. thin shells and elastic rods. *Thin shells* are characterized in that two dimensions, namely the length and the width, dominate the third dimension, namely the thickness. Consequently, triangle meshes are (usually) employed to represent the surface covered by the solids. Thin shells are widely employed to represent cloth. A broad survey is found in e. g. [CK05a]. Since this thesis does not address the simulation of thin shells, they are not further discussed here.

In the case of *elastic rods*, one spatial dimension, namely the length, dominates the other two dimensions that constitute the cross-section. The visual appearance of these objects is that of a curve in space, and consequently linear chains of control points constitute the discretization. Motivated by the governing elasticity theory, such objects are termed “elastic rods”, and they are employed to represent e. g. threads, ropes, or hair-strands.

Similar to the case of thin shells, the volume of elastic rods is negligible. Still, the constitutive restitution laws are governed by the deformed material of the rod. If a rod is bent, then the compressed material induces a bending moment that works against the deformation. Moreover, if the rod is twisted around the centerline, then the material is sheared. This torsional deformation induces a torque that works against the deformation. The handling of material torsion is identified as the main challenge in the modeling of elastic rods.

In the following summary, the related work in elastic rod modeling in the field of computer graphics is discussed. This allows to categorize the deformation model that is being presented in Chapter 3. Before, some works in the field of mechanics are mentioned.

Mechanics In the field of mechanics, researchers have been investigating into analytical and numerical solutions for the static and dynamic equilibria of elastic rods. A comprehensive introduction is provided in the book of Antman [Ant95].

The deformation of an elastic rods is usually studied by defining the position and the orientation of its start- and end points, which results in a boundary value problem. The analysis and numerical solution of the corresponding system of ordinary differential equations is addressed in e. g. [Keh97]. The dynamic evolution of an elastic rod is discussed in e. g. [Dic94]. Finite element methods to numerically simulate elastic rods have been considered in e. g. [BP04] or [CLW06]. In Sec. 3.5, a deformation model for elastic rods is presented that bases on [CLW06].

If the end-to-end rotation of an elastic rod is varied, then the rod exhibits a buckling behavior. Under continued torsional load, the rod comes in self-contact and twists around itself, thereby forming loops that are also known as plectonemes. The corresponding static solutions are studied in e. g. [vdHNGT03]. The resulting systems of differential equations are usually solved by employing shooting techniques. In contrast, in Chapter 3, this thesis proposes a contact model that allows for the efficient dynamic simulation of rods in self-contact.

Computer graphics In the field of computer graphics, one-dimensional deformable objects have mainly been modeled in order to represent threads in virtual suturing or hair strands for animation purposes. Three classes of deformation models can be identified, namely mass-spring deformation models, spline-based deformation models, and Cosserat deformation models.

Mass-spring deformation models discretize the elastic rod into a set of mass-points that are linked by segments. Linear spring forces preserve the length of the segments, and angular springs preserve the resting angles between the segments. One of the first approaches that employs this technique in order to address hair simulation has been proposed by Rosenblum *et al.* [RCT91]. They employ a mass-spring model to compute the dynamics. A physically-based approach has later been proposed by Anjyo *et al.* [AUK92]. They identify the bending deformation of a single hair of being identical to the numerical simulation of a cantilever beam simulation. Since shear deflection is neglected, the elastic laws result in a particularly simple second-order differential equation that can be analytically solved in order to obtain the dynamics of the mass points. The same deformation model has later been employed by Daldegan *et al.* [DMTKT93] where a complete framework for the simulation of hair is presented.

However, these deformation models neglect material torsion. This is a fair assumption in the context of hair simulation, since hair has a high torsion resistance, as pointed out by Plante *et al.* [PCP01]. They address the interactions between hair strands. Likewise, Phillips *et al.* [PLK02] has employed a spring-based deformation model to simulate knot tying. Their model is adaptive, which facilitates the computation of knots, as discussed later in Chapter 4. Brown [BLM04] has presented a non-physical deformation model for the simulation of knotted ropes. Similar to Phillips *et al.*, the rope is modeled from nodes that are connected by springs. The dynamic evolution is obtained by employing a “follow the leader” methodology: If one node is grasped, then the positions of the other nodes are

iteratively computed such that the lengths of the springs are preserved.

Recently, some remarkable works have been published that model torsional deformation by employing a mass-spring system. To accomplish this, Wang *et al.* [WBD*05] have proposed to express the material torsion of each segment with a scalar torsion angle parameter. The spatial difference between two torsion angles results in both a torque acting on the torsion parameters and in a spatial force acting on the mass points. A similar strategy has been proposed by Kubiak *et al.* [KPGF07], where the position-based dynamics approach from Müller *et al.* [MHHR07] is adopted to compute the evolution of the mass-points. A torsion constraint governs the torsional torques. And recently, Selle *et al.* have employed a mass-spring model for simulating hair, where ‘altitude’ springs handle the torsional restitution. To avoid the singularity that occur in straight hair strands, additional particles that do not lie on the main axis are employed [SLF08]. Still, it is unclear how the coupling between the torsion and the bending moments is realized. An immersed discussion on the coupling between twisting and bending moments is given in Sec. 3.2.

The approaches of Wang *et al.*, Kubiak *et al.* and Selle *et al.* all suffer from the same problem, notably that the bend and the twist forces are obtained in fundamentally different ways. Although this might lead to satisfying results in certain applications, an approach who directly minimizes the angular difference between succeeding material frames is favorable, because it allows to reproduce the buckling and looping phenomena. In this case, the material frames, being representatives of the group of orthonormal rotations $SO(3)$, serve as DOFs of the rod. However, the material frames must be *adapted* to the rod centerline, i. e. the third axis of the material frame must conform to the tangent of the centerline, which corresponds to the cross-section being perpendicular to the centerline. This idea has been presented in two recent publications. Choe *et al.* simulates the rod as a chain of segments and springs [CCK05]. Each segment i is simulated as a rigid body, having a center of mass (COM) $\mathbf{x}_i \in \mathbb{R}^3$, and an orientation $\mathbf{R}_i \in SO(3)$, where the orientations are parameterized by quaternions. The springs between the rigid bodies exert forces on the COMs, and torques on the orientations. The resulting deformation model is employed to simulate hair strands. Similar in spirit is the approach of Grégoire and Schömer [GS06], since they also employ quaternions to represent the orientations of the segments. However, they do not simulate the COMs of the segments. Instead, they simulate the mass points that link the segments, and they treat the segments as springs. The conceptual differences are illustrated in Fig. 2.1. However, this representation requires to adapt the quaternions explicitly to the centerline. To accomplish this, they employ the penalty method based on a constraint energy. The deformation model that is presented in this thesis in Chapter 3 is motivated by this approach.

There exist some approaches that propose to simulate an *inextensible* elastic rod as a chain of articulated rigid bodies linked by joints. This idea has been proposed by e. g. Hadap [Had06]. He identifies the differential algebraic equation (DAE) based formulation to correspond to a linear complementary problem (LCP).

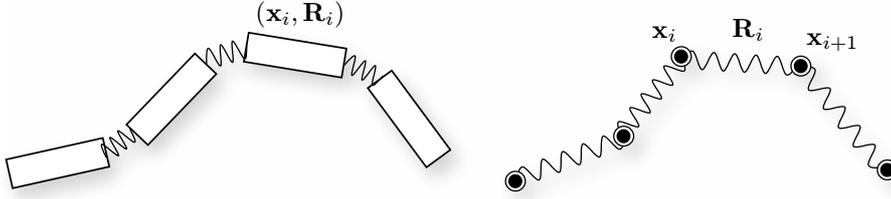


Figure 2.1: Illustration of the conceptual differences between the approaches of Choe *et al.* [CCK05] and Grégoire and Schömer [GS06]. Left: Choe *et al.* discretize the rod into rigid bodies $(\mathbf{x}_i, \mathbf{R}_i)$ that are linked by springs. Right: In contrast, Grégoire and Schömer discretize the rod into mass-points linked by springs, and pure orientations \mathbf{R}_i that are adapted to the mass-points by the penalty method.

The LCP is then solved with an iterative scheme. Later, Gayle *et al.* [GLM06] have proposed an adaptive scheme that bases on Featherstone’s divide-and-conquer scheme. Both approaches perform well and show impressive results. However, due to the hard constraints, the resulting equations are stiff and require advanced techniques to solve the corresponding systems.

Spline-based deformation models, also known as continuous deformation models, discretize the rod into nodes that constitute the control points of a spline curve. Perhaps the first spline-based deformation model has been proposed by Terzopoulos [TPBF87], without having gone much into details. The advantage of a spline-representation is that concepts such as curvature and (geometric) torsion are readily available in the theory. Thus, the elastic forces that govern the temporal evolution can be easily obtained, as illustrated by Qin and Terzopoulos in [QT96] for the case of non-uniform rational B-splines. Later, Rémion *et al.* have employed the Lagrangian formalism to obtain the dynamic evolution of a spline curve. They have employed the splines to model knitted clothing [RNG00]. Based on this work, Lenoir *et al.* have proposed an approach to model surgical threads [LMGC02], showing that real-time performance can be achieved with their deformation model. Recently, Kaldor *et al.* have addressed the simulation of knitted cloth at the yarn level, where the yarn is simulated with a single spline curve [KJM08].

While these models achieve a good performance, they are not able to handle material torsion (i. e. the twist of the rod around its centerline), since the DOFs of the rod are exclusively given by the control points. In contrast, the Cosserat approaches that are described in the subsequent section consider the material frames when establishing the strain-stress relations. A recent work has combined a spline-based deformation model with a Cosserat approach: Theetten *et al.* [TGAB07] start with the well-known spline formula. In contrast to the previous approach, they further consider in each control point a fourth DOF, namely its rotation field. Then, they have observed that the twisting force can be expressed as the sum of the geo-

metric twisting (that depends only of the geometry of the curve), and the material torsion (that depends of the spatial difference of the rotation field). To dynamically evolve the DOFs, they employ a diagonal 4×4 inertia tensor, where the fourth entry conforms to the inertia of the cross-section. The equations of motion result in a linear system in the accelerations, where the mass-matrix is banded. They propose an LU decomposition to solve the system. Their approach is *geometrically exact*, i. e. the geometry of the rod and its balance equations are treated exactly, and the only approximations that are accepted concern the spatial and temporal discretization [SFR90]. However, since they do not carry an explicit representation of the material frames, they need to update the material direction in order to visualize the rods, which can be difficult. This contrasts approaches such as [CCK05] or [GS06] where the material frames are directly considered as DOFs of the model. Later, Theetten *et al.* have proposed an extension of their deformation model that can dynamically switch between a static and a dynamic solution, depending on the required simulation context [TGDM07].

Cosserat deformation models consider an oriented curve, i. e. in each point of the continuous curve, they think of an orthonormal basis that conforms to the material frame. The stresses then relate to the spatial difference between those frames. Pai has introduced the Cosserat model in the field of computer graphics. In [Pai02], he establishes the relations between the strains, and the resulting forces and torques. By comparing the spatial evolution of the material frame along the centerline to the temporal evolution of a rigid body, he arrives at a set of ordinary differential equations (ODEs). As usual in mechanics, he specifies the positions and orientations of the start- and end point, which allows to solve the ODE in two sweeps. It is important to be aware of the conceptual difference to the previously discussed approaches: Pai does not explicitly simulate the mass points comprising the ‘visual’ configuration of the rod, but he reconstructs the configuration of the rod from the boundary conditions. However, this complicates the collision handling.

While Pai employs the stress-strain relationship to reconstruct the deformed rod, Wakamatsu and Hirai derive the elastic energy from the strain rates [WH04]. Then they solve the corresponding energy minimization problem in order to obtain the configuration of the deformed rod. They employ the Euler angles that parameterize the material frames as DOFs of the rod. Similar to Pai, their approach is limited to computing the static deformation. Moreover, the Euler angles are subject to singularities at the poles [SM06].

This limitation has later been alleviated by Bertails *et al.* [BAC*06] in order to simulate the dynamics of hair strands. Their approach shares similarities with Wakamatsu and Hirai’s approach in that they employ an energy formulation in order to solve for the rod configuration. However, in contrast to the former, Bertails *et al.* have considered the curvature, i. e. the Darboux-vectors, as DOFs of the rod, which has the advantage of avoiding the intrinsic singularities of the Euler angles. To obtain the equations of motion, they have employed the Lagrangian formalism, which results in a linear system in the curvature accelerations. However, their approach has the drawback that the mass-matrix is dense. Consequently, to compute

the dynamics of a hair strand discretized into N segments, the number of operations grows with $\mathcal{O}(N^2)$. An extension of Bertails *et al.*'s approach to model meshes of Cosserat beams has been proposed by Chang *et al.* [CSZ07]. They focus on extracting the vibrational modes from the generalized mass-matrix. In order to carry out this process on a full mesh, they simplify the mesh, then solve the eigenvalue problem on the simplified mesh, and finally project the deformations back on the initial mesh.

The research on elastic rods cumulates with a recent publication of Bergou *et al.* which has considered the dynamic evolution of discrete elastic rods with torsion [BWR*08]. Similar to the approach presented in this thesis, they discretize the centerline into mass points. To represent the material torsion, they observe that the velocity of the twist waves is much larger than the velocity of the bending waves. Consequently, they do not carry the material torsion for each centerline segment. Instead, they represent the material torsion only at the boundary segments. Similar to Theetten *et al.* [TGAB07], they then derive the twist forces from the spatial derivative of the material torsion, which is constant. The main challenge is to compute the variation in the material frames, which is related to the parallel transport of the material frame along the centerline. To accomplish this, they consider the discrete *holonomy*, which is a concept of discrete differential geometry. Being geometrically exact, their model reproduces both static and dynamic phenomena of twisting elastic rods.

Cosserat models such as [Pai02, BAC*06] commonly provide a geometrically exact configuration of the deformed elastic rod. However, their main drawback is that they reconstruct the centerline instead of explicitly simulating it. Thus, geometric and contact constraints along the centerline are hard to maintain. In contrast, discrete deformation models and spline-based deformation models simulate the control points and thereby ease the constraint maintenance. On the other hand, these deformation models commonly lack of the physical plausibility since torsion is not handled in a consistent way. In Chapter 3, a deformation model is proposed that combines both techniques. It results in a physically plausible deformation model for elastic rods which enables an efficient and accurate contact handling.

2.2.3 Adaptive methods

In the context of deformation modeling, a well-known methodology to make a method more accurate is the *adaptivity*. It bases on the observation that the required number of DOFs to reproduce a deformation depends mainly on the current deformed configuration. In the extreme case, an undeformed object can be simulated as a rigid body, requiring as few as six DOFs. If, in turn, the object is subject to external forces, e. g. arising from collisions, then it deforms, thereby requiring additional DOFs to represent the deformation.

If the underlying spatial discretization is considered (for example, this can be a tetrahedral mesh, a triangle mesh, or a chain of control points), then the goal of an adaptive method can be reformulated as “reducing the DOFs in regions where

mechanical or numerical accuracy is not mandatory”. Consequently, an adaptive method must define a *adaptation-* or *error criterion* that measures the adequacy of a given resolution. Based on this error criterion, the resolution is then increased or decreased to meet the required accuracy. In the literature, adaptive methods are well-established. They address three-, two- and one-dimensional deformable objects and propose a wealth of different techniques to measure the adequacy and to adapt the resolution.

One of the first works in the field of computer graphics that employed an adaptive strategy has been presented by Hutchinson *et al.* [HPH96] in the context of cloth simulation. They model the cloth with a mass-spring network, and the physical properties are averaged during refinement. However, since the material properties in a mass-spring deformation model depend on the discretization and are not easily mapped to different resolutions, the proposed technique does not guarantee a globally consistent behavior. This problem has been alleviated by Debunne *et al.* [DDBC99]. The elastic forces are computed by employing constitutive relations from continuum mechanics that are discretized with the method of finite differences (FD), thereby guaranteeing that the material properties do not vary along the different resolutions. Later, Debunne *et al.* [DDCB00, DDCB01] have published two articles describing a system which allows for the adaptive dynamic simulation of deformable objects. They employ finite elements to compute the deformations.

Adaptive schemes can be distinguished with respect to the way they store and manage the different resolutions. Some approaches pre-compute the meshes at different resolutions, and switch between the resolutions during the simulation. This approach is fast and further guarantees that the element quality does not suffer from the refinements [DDCB01]. However, it has the limitation that the maximum number of DOFs is given by the most dense mesh, as pointed out by Wu *et al.* [WDGT01]. A prominent representative example are the *multigrid* approaches that store a nested hierarchy of meshes. They prolong the solution at a coarse level to correct the solution at a finer level [BH00]. Multigrid schemes have been successfully applied for three-dimensional deformation modeling [OGRG07] and for thin-shell simulation [GTS02]. A multigrid approach on the GPU has recently been proposed by Georgii and Westermann [GW05].

Instead of carrying a hierarchy of pre-computed meshes throughout the simulation, it is also possible to re-mesh the domain in order to meet the spatially and temporally varying accuracy. Such refinement is denoted as *unstructured* refinement, since it carries no history of refinement, and a previous mesh cannot be reconstructed. This technique has been successfully applied in the context of highly deformable objects that merge upon collision [BWHT07], or fluid simulations with dynamically changing boundaries [KFCO06]. As they employ a conforming mesh, the re-meshing is expensive. This has been alleviated in [WT08] where the meshes are non-conforming. The authors argue that by employing a high-resolution embedded surface, the boundary of the (invisible) mesh does not disturb the visual quality of the animation.

In contrast, subdivision schemes perform a *structured* refinement by subdivid-

ing the elements if additional DOFs are required. Thus, the coarse mesh can always be reconstructed. For 2D meshes, the subdivision comes with a well-established theory and a corresponding set of simple rules [ZS*99]. In cloth simulation, the goal is often to increase the visual quality when cloth is draped over sharp edges, as e. g. proposed by Bridson *et al.* [BFA02]. Later, Thomaszewski *et al.* have employed a Loop's subdivision scheme to improve the accuracy of cloth simulation [TWS06]. Subdivision is also necessary in the context of cutting or fracture. In this case, the elements that are located on the cutting- or fracture-plane have to be subdivided in order to accurately represent the changed geometry. O'Brien and Hodgins achieve this with a local remeshing of the tetrahedra surrounding the fracture plane [OH99].

However, while subdivision is easy for triangles in e. g. cloth simulation, it is a much harder job for tetrahedra. This comes from the fact that the elements in a spatially adaptive simulation are usually split in isolation, but this inevitably leads to a lack of *compatibility* that exhibits in the so-called T-vertices. To overcome this problem, either neighboring tetrahedra have to be bisected accordingly, with e. g. a red-green triangulation rule [MBTF03]. Still, the resulting rules are complicated and cumbersome to implement. Alternatively, constraint forces can be employed to prevent the formation of cracks. However, constraint techniques usually require the solution of a system of equations, and add stiffness to the simulation. Therefore, some approaches resort to simple techniques such as linear interpolation of the physical properties along the interface [DDCB01]. Recently, an alternative technique gained increasing attention, namely to employ "virtual particles" that bridge non-conforming meshes [EEHS00, SSIF07]. A different way has been opened by Capell *et al.* where the surface of a deformable object is embedded into a hexagonal grid, thereby facilitating the subdivision [CGC*02a]. Further, this allows for an efficient representation with octrees [BPWG07] and can also be employed to refine clusters for geometric deformation models [SOG08]. And instead of refining the elements and therewith running into the incompatibility problems, Grinspun *et al.* have proposed to refine the basis function governing the FE interpolation [GKS02].

These approaches have in common that they refine the elements by introducing new DOFs in order to meet the accuracy requirements. In contrast, one could also think of an adaptive scheme that simplifies the dynamic motion of a deformable object by rigidifying the regions where the relative motion can be neglected. Such an approach has been proposed by Redon *et al.* in the context of articulated body simulation [RGL05]. This approach has later been extended by Gayle *et al.* to handle contacts of articulated robots [GLM06]. A related approach has been proposed by Bertails *et al.* in the context of hair simulation, where individual hair strands are split and merged according to the dynamic movement of the hair [BKCN03].

Adaptive elastic rods Due to the inherently simple structure of one-dimensional deformable objects, adaptive schemes do not need to handle incompatibility prob-

lems. Approaches that reconstruct the configuration of the elastic rod from boundary conditions can choose the step-size of the spatial numerical integration scheme based on an error criterion [Pai02]. In contrast, spline-based approaches can resort to the vast amount of literature on spline curve subdivision. Lenoir *et al.* have proposed an approach for the adaptive simulation of a spline-based deformation model [LGCM05]. In order to simulate knots, the DOFs are inserted based on the current contact configuration.

In contrast to spline methods, mass-spring and Cosserat methods cannot introduce the new control points based on geometric rules. This is because the geometric configuration of the rod centerline is not governed by geometric rules, but depends on the underlying constitutive relations of the deformation model. Consequently, if a new control point is introduced at the barycenter of two old control points, then the deformation energy will increase. In turn, this quickly makes the simulation instable if the refinement operations are cascaded. This problem has rarely been addressed in the physically-based simulation. Phillips *et al.* have reported this instability in the context of elastic rod simulation, and they have proposed to add artificial damping in order to dissipate the deformation energy [PLK02]. However, this procedure has the drawback that the dynamics of the elastic rod is disturbed.

It would be favorable to have a scheme that determines the control point positions by minimizing the deformation energy functional. Such schemes are also known as variational subdivision schemes and have been addressed frequently in the geometry-community. Kobbelt and Schröder have described a variational approach to produce smooth curves by minimizing some quadratic energy functional [KS98]. This results in a linear system which can be solved adequately. In Chapter 4, an approach is proposed that computes the control nodes governed by the minimization of a non-linear energy functional. Therefore, any post-stabilization in the spirit of [PLK02] is avoided, and the stability of the simulation is improved.

2.3 Contact handling

In Sec. 2.2, approaches have been discussed that compute the motion paths of deformable objects by time-integrating the Lagrange equations of motion (2.1). The motion paths of the nodes of the discretized object are governed by the internal forces resulting from the potential energies, and by the external forces arising from e. g. user interactions. In this case, the motion of the nodes has been assumed as unconstrained.

However, in many situations, the positions of the objects are restricted to certain regions within the simulation domain. If e. g. the simulation domain is bounded by a floor and by walls, then the objects must not interfere with those boundaries. Further, if the simulation of multiple objects is considered, then the objects must not interfere with each other. Moreover, a deformable object must not interfere

with itself under deformation.

In this section, approaches to handle rigid and deformable object contact handling are reviewed. First, approaches are discussed that address the rigid object contact handling. Although no contribution in this field is being presented in this thesis, it turns out to be important in the field of elastic rod contact handling. Second, approaches that focus on the deformable volumetric object contact handling are discussed. Finally, approaches that handle collisions for elastic rods are covered. Since here, the cross-section is assumed to be rigid while the centerline is deformable, these approaches effectively combine rigid and deformable contact handling.

Throughout the discussion, the approaches are classified into *non-penetration methods* and *penalty methods*. The term ‘non-penetration methods’ is employed for methods that guarantee that the constraints are maintained within a certain tolerance, which corresponds to constraint forces that directly cancel out the contributions of the other forces that will violate the constraint. These methods are sometimes denoted as ‘constraint methods’, although this term is misleading, since also penalty methods are employed to handle constraints. Non-penetration methods contrast the penalty methods that relate the constraint forces to a constraint violation measure. Those penalty forces only act if the constraint is violated, and consequently, constraints can never be maintained exactly. A good introduction into constrained simulation are the course notes of Witkin [Wit01].

2.3.1 Rigid object collision response

The rigid object contact handling is a famous and difficult problem in mechanics, robotics and animation, which comes mainly from the fact that a contact force changes the momentum of the whole object discontinuously. This contradicts the otherwise continuous nature of the simulation. Contact forces in rigid object simulations must be computed globally, requiring to solve a system of equations.

Contact forces can be understood as a special case of constraint forces, since they enforce the non-penetration constraints that are imposed in the multi-body simulation. However, there are some subtleties that have to be considered, notably that contact forces are impulsive in dynamic collisions, since the velocities of the incorporated objects change discontinuously. In contrast, the contact forces are continuous in resting contacts, since here, the incorporated objects have no velocity [Bar89]. The decision whether non-impulsive contact forces are sufficient to resolve the collision is NP-complete for perfectly rigid objects [Bar91]. In practice, many approaches employ impulses extensively, since the inaccuracy is tolerable. For example, Guendelman *et al.* compute impulses to first stop the collisions. Then they compute the contact forces on the contact points that are still colliding, thereby assuming that they constitute the resting contacts [GBF03].

For the case of frictionless contacts, the non-penetration constraints form a quadratic program (QP) whose solution is NP-hard. Baraff has illustrated that the QP corresponds to a linear complementary problem (LCP) [Bar94]. He solves the

LCP by employing a variant of Dantzig's algorithm. A good and intuitive explanation why the employ of a quadratic objective results in a realistic motion compared to linear objectives is given by Milenkovic and Schmidl in [MS01]. Later, Redon *et al.* have transformed the LCP to a nearest point problem, which, being mathematically equivalent, has advantages from a computational point-of-view [RKC02].

The incorporation of friction further complicates the problem, mainly due to the non-linear behavior of the contact forces, and because they live in the tangential space. The later observation implies that the tangent and the normal space become coupled, and consequently friction forces can change the contact state [DDKA06]. Thus, the constraints arising from Coulomb friction lead to very difficult formulations [Bar91]. To treat contact and friction in a unified manner, the Coulomb friction cone is usually approximated with a k -sided pyramid, as illustrated by Anitescu *et al.* [MFD99] and adopted by e. g. Duriez *et al.* in the field of deformable contact handling [DDKA06]. However, many authors identify friction to be an issue for which a convincing, correct and efficient solution is still missing.

Having the accentuated discontinuous nature of rigid object contact handling in mind, there has been a shift towards impulse-velocity level based contact methods in the past years. Those methods treat both, collisions and resting contacts, by applying impulses on the incorporated objects. In order to prevent that e. g. a box resting on an inclined plane slips down, Mirtich and Canny have proposed to model micro-collisions [MC94]. In contrast, Guendelman *et al.* have fought this problem with a time-stepping scheme that separates collision and contact [GBF03]. To accelerate the global solve for feasible impulses, they employ a contact graph, i. e. they process the objects in the order they are resting on each other. Recently, a velocity-level simulation has been proposed by Kaufman *et al.*, where the expensive formulation of previous optimization-based approaches is reduced to two convex, separable QPs per object (as opposed to one global QP) [KEP05]. They have further proposed to approximately solve the QPs with an iterative method. An alternative has been proposed by Bender and Schmitt, where iteratively computed impulses handle resting contacts [BS06].

In addition to force- and velocity-level simulations, there exists a third class of simulation methods, notably position-based simulations. Milenkovic and Schmidl have proposed to compute the constrained positions directly by moving the objects as close to their unconstrained target positions as possible, considering the imposed non-penetration constraints. After the positions are found, the velocities are updated accordingly [MS01]. The position-based approach of Müller *et al.* is similar in spirit, although it does not guarantee to find feasible positions in all situations [MHHR07]. Moreover, the friction model is approximative.

In the context of constrained simulations, there has been recently a development in the direction of *predictor-corrector methods*. To motivate this approach, one has to reconsider that the idea behind the constraint enforcement is to compute a force (or an acceleration or impulse) such that the contribution of the other forces (or impulses) that work against the constraint are canceled out. While the theory holds for continuous system, it comes with an unavoidable problem if the

time discretization is considered. By numerically integrate the force with a time step h , the constraint will not be met exactly, but there is a numerical drift in the order $\mathcal{O}(h)$. Even worse, this numerical drift accumulates during the simulation. This problem has already been reported by Baumgarte in 1972 [Bau72], where he proposes to add stiff springs that keep the numerical drift at least within certain bounds. However, the Baumgarte stabilization has the drawback that a spring- and a damping constant have to be found.

The predictor-corrector methodology overcomes this problem in a natural way, notably by computing the unconstrained position and the targeted constrained position in the subsequent time step $t + h$. Then, a force or an impulse can be computed such that, numerically re-integrated at t , the constraint is exactly met at $t + h$. This technique has been employed frequently in the community. In [WTF06], Weinstein *et al.* use the term “pre-stabilization”, in contrast to the Baumgarte post-stabilization. A similar technique is employed by e.g. Cirak and West in the context of mechanical simulations of elastic objects [CW05], Duriez *et al.* in the context of haptic simulations [DAK04], and by Choe *et al.* in the context of hair simulation [CCK05]. While they deal with linear constraints for which a closed form exists [GBT06], Weinstein *et al.* involve angular constraints that require the iterative solution of a non-linear system [WTF06].

Although the predictor-corrector methodology can be considered as physically-based since constraint violations are avoided, it nevertheless provides but an approximation of the true collision course, with the error being linearly dependent on h . This comes from the fact that in contact handling, the target constrained positions are usually obtained from the unconstrained positions, which are predicted by numerical integration. In contrast, an exact scheme would back up the time of collision, and compute the answer to the collision based on the configuration of the geometry at that time. Amongst others, Guendelman *et al.* have been aware of that problem. They have proposed to employ the predicted positions to compute the impulses, but to employ the original positions to compute the contact forces, since resting contacts require more accuracy [GBF03].

2.3.2 Volumetric deformable object collision response

The contact handling for deformable objects shares many similarities with the contact handling of rigid objects. However, there is one important difference which results in some simplifications. If two deformable objects collide, then the volumes become compressed for a non-zero amount of time. Consequently, if a stack of deformable objects is considered, then the impulse that is imparted from one object to the other is not immediately transferred through the stack. In turn, the collisions can be resolved locally for each *contact surface* [PPG04]. The term ‘contact surface’ refers to the union of the collided nodes on the surfaces of the incorporated objects.

As a consequence, in the contact handling of deformable objects, a collision does not discontinuously change the momentum of the whole object, but only in

a small localized area around the collision, depending on the elasticity of the object [BW92]. Consequently, the employ of the penalty method is particularly popular in the contact handling of deformable objects. Still, other approaches have attacked the problem with physically-based methods, inspired by rigid object contact handling.

Penalty methods Penalty methods relate the constraint force to a measure of constraint violation. Consequently, the resulting penalty force acts only if the system violates the constraint. In the case of contact handling, the contact forces are related to an interpenetration measure. This has the unfortunate consequence that, in order to satisfy the constraints, the penalty force constant must be large, which in turn results in stiff differential equations that require small time steps [MW88]. Moreover, it is impossible to guarantee a certain bound on the constraint violation. Nevertheless, due to their intrinsic simplicity, penalty methods have always been popular, especially in the context of interactive animations [HTK*04, KMH*04, DMG05]

In order to produce smooth contact forces, penalty methods must rely on a robust computation of the interpenetration measure. To accomplish this, most approaches compute the penetration depth, i. e. the shortest distance that a penetrated node has to be displaced in order to obtain a collision-free configuration. The penetration depth direction then constitutes the contact space. Fisher and Lin have proposed to compute a distance field within the volume. The exact distances are given at the nodes and linearly interpolated within the volume elements. The penetration depth is then easily obtained from the interpolated distances [FL01]. Later, Hirota *et al.* have extended this approach to handle self-collisions [HFS03]. However, in the case of large time-steps and large relative velocities, the penetrations can be large. In turn, the inconsistent penetration depths and the corresponding contact forces result in an implausible behavior. This problem has been attacked by Heidelberg *et al.* where a consistent penetration depth is iteratively computed in an advancing-front fashion [HTK*04].

A different way has been proposed by Keiser *et al.* [KMH*04]. They first compute the contact surface, governed by the static equilibrium. Then the penalty force is related to the distance of the penetrated surface to the contact surface. However, since these forces are equally distributed on both collided surface in order to guarantee force-equilibrium, the use of the proposed procedure is questionable.

Non-penetration methods Non-penetration methods differ from penalty methods in that they compute the constraint forces *before* the constraint is violated. The procedure is similar to the rigid object contact handling, notably to impose the non-penetration constraints and then to solve for the constraint forces or impulses.

In the rigid object contact handling, the non-penetration constraints for a pair of colliding objects can be written as implicit functions in the 12 DOFs of the involved objects. However, in the case of deformable objects, the constraint functions must

consider the deformation of the surfaces. For the static and frictionless case, this is known as Signorini's problem which states that the normal contact force and the normal stress arising from the deformation must sum to 0. The resulting equations can be combined with the generic contact force conditions [Bar89], which results in an LCP. Duriez *et al.* have proposed an iterative Gauss-Seidel-like solution to solve the frictionless case [DAK04]. Since the Signorini's problem relates the contact force to the pressure, and since Duriez *et al.* assume a linear elasticity model, the contact forces become linear, which allows for an efficient implicit integration. A similar LCP formulation has been proposed by Pauly *et al.* in order to compute contact forces acting on point-sampled quasi-rigid objects [PPG04]. They solve the resulting system with an iterative scheme.

Similar to rigid object simulation, the incorporation of friction complicates the problem. Based on their prior work, Duriez *et al.* have incorporated dry friction by approximating Coulomb's law with a k -sided pyramid. The resulting system is about k times larger compared to the frictionless case [DDKA06]. In contrast, Pauly *et al.* have assumed that the friction force do not result in significant secondary deformations [PPG04]. Consequently, they decouple contact and friction, and add the resulting friction contribution to the contact forces resulting from the LCP.

Motivated by character animation, a widespread way is to model deformable objects with a rigid core, combined with a deformable skin. James and Pai have proposed a boundary element method to discretize the deformable skin, but have at that time been limited to static collisions between one deformable and one rigid object [JP99]. Later, they have proposed to deform the skins on the GPU [JP02]. Similar in spirit is the approach of Galoppo *et al.* [GOM*06]. They employ a pure impulse-based formulation, and solve for the impulses with a Sherman-Morrison-Woodbury update of the corresponding matrix. They only consider forces that couple the rigid core with the corresponding deformation texture nodes, but neglect the elastic forces between the texture nodes. They have later extended their approach to handle collisions between skinned skeletons, thereby addressing the coupling between contact forces acting on the skin, and the resulting forces and torques on the skeleton [GOT*07].

In this thesis, a predictor-corrector is presented that first computes the contact surface governed by the dynamic contact equilibrium, before computing impulses such that the non-penetration constraints are met. Earlier, Cirak and West have proposed a predictor-corrector scheme to compute impulses in the context of FE simulations of interacting elastic objects. To maintain the non-penetration constraints, they displace the colliding nodes after the computation of the contact impulses [CW05]. In contrast, the approach proposed in Chapter 5 computes impulsive contact forces that directly displace the colliding primitives, thereby maintaining the non-penetration constraints. In contrast to previous penalty-based approaches such as [HTK*04, KMH*04], the approach comes without user-defined stiffness constants. Thereby, the overshooting problem related to large time steps is avoided.

2.3.3 Non-volumetric deformable object collision response

In the preceding section, approaches have been discussed that dealt with the contact handling of volumetric deformable objects. However, when addressing the contact handling of non-volumetric deformable objects such as cloth or elastic rods, it needs to be considered that these objects are assumed to be incompressible. This comes from the observation that the thickness of the objects is assumed to be small compared to their extent, and as a consequence, allowing the volume to be compressed would immediately result in interpenetrations. In turn, the contact handling cannot be anymore computed locally, but it must be solved globally, similar to the rigid object contact handling.

Alternatively, instead of considering the case of a small and incompressible volume, one can as well think of a virtual compressible volume covering the objects. Then, forces are computed that handle the collisions between these volumes. This is the theme in most penalty methods, where the resulting contact forces are often denoted as *repulsion forces*. Many approaches that address the contact handling of non-volumetric deformable objects cannot be unconditionally assigned to either penalty or non-penetration methods, since they often mix repulsion and constraint techniques.

Repulsion methods Since for non-volumetric objects, interferences result quickly in an inconsistent state, penalty methods define a virtual volume around the non-volumetric objects, and then compute repulsion forces that prevent a collision. This has been e. g. done by Baraff and Witkin [BW98], or by Bridson *et al.* in the context of cloth simulation [BFA02]. Later, the method of Baraff and Witkin has been adopted by Choi and Ko by formulating constraints based on the proximity between the nodes [CK05b].

The penalty method has also been employed in the contact handling of elastic rods. Choe *et al.* have proposed the admissive cylinder model that effectively conforms to a tubical volume that is laid around the hair wisps. Repulsion forces then prevent that two hair wisps collide [CCK05]. A similar contact model has been considered by Bertails *et al.* [BKCN03, BAC*06].

Non-penetration methods In contrast to repulsion-methods, exact methods consider the volume as incompressible, which makes the collision handling a global problem. The global nature of cloth contact handling has been addressed by Volino and Magnenat-Thalmann in a series of articles [VCT95, VMT00]. They have proposed a geometric scheme that successively alternates the acceleration, velocity and position of the colliding nodes in a temporally delayed way. In order to realize the coupling of the collisions, the resulting equations are gathered in a linear system. However, the delayed contact resolution is non-physical and can lead to artifacts. In contrast, Provot has proposed to compute impulses that change the velocities of the involved nodes discontinuously. Again, the computation is approximative, and the friction impulses are simply added. In order to account for

secondary collisions, this process is iteratively repeated in order to let the impulses quickly travel through the stack of cloth [Pro97]. Later, Huh *et al.* have adopted Provot's approach, but in contrast to him, they consider the momenta of the involved particles. They impose the condition that the contact impulses must be momentum-conserving, and minimal in their magnitude [HMB01]. This results in an LCP which they solve with a conjugate gradient method.

When cloth is colliding, then it tends to clump together, thereby eliminating the relative velocities. These clumps will then behave like rigid objects. This observation has been made by Provot who proposes to bundle colliding nodes to rigid impact zones [Pro97]. Based on this work, Bridson *et al.* have later proposed a three stage approach to handle cloth collisions: First, collisions are avoided by modeling repulsion forces with a penalty method. Second, impulses are computed that change the velocity of those nodes discontinuously that are "too close together". This process is repeated iteratively in order to account for secondary collisions. Third, nodes that are still sticking together after the iterative process are bundled to rigid zones, and treated accordingly [BFA02]. Numerous consecutive works have employed their response approach [TWS06, BMWG07]. To handle the non-physical dissipative behavior of the impact-zone rigidifying, Harmon *et al.* have proposed an approach that cancels impact but not the relative sliding motion. Basically, they project the normal velocities, thereby preserving as much of the tangential motion as possible [HVTG08].

In a recent publication of Müller *et al.*, the authors have proposed to model the elastic forces of cloth and the contact forces arising in cloth collisions in a unified way. They assemble the constraints and compute the feasible target positions with an iterative approach, thereby resolving conflicting constraints. Since the target positions are known, the numerical integration becomes unconditionally stable [MHHR07]. In this thesis, this approach is adopted to compute contact forces for the case of colliding elastic rods.

In the field of contact handling for elastic rods, the exact maintenance of the non-penetration constraints has rarely been addressed. Of note is the approach of Plante *et al.* that proposed an impulse-level contact handling of hair wisps, but secondary collisions are not addressed [PCP01]. In Chapter 6, an approach is presented that applies an iterative global scheme to colliding elastic rods. In contrast to [PCP01], secondary collisions are handled properly. In contrast to the approaches based on repulsion forces [BKC03, CCK05, BAC*06], the approach comes without user-defined stiffness constants, and the non-penetration constraints are maintained, even for large stresses. Moreover, the overshooting problem related to large time steps is avoided.

Chapter 3

CORDE: A deformation model for elastic rods

In this chapter, a novel simulation method for elastic rods is presented. After introducing the continuous constitutive relations, it is discussed how to discretize the rod and how to obtain the elastic forces that govern the static equilibrium of the rod. In addition, the dynamic evolution of the elastic rod is discussed extensively. Together, the methods allow for the robust and efficient physically-based simulation of elastic rods.

In the past, several approaches have been proposed to model elastic rods. They differ in the representation of the curve and in the way the material torsion is handled. The easiest way is to model the rod using control points linked by springs. The N control points $\mathbf{r}_i \in \mathbb{R}^3$ define the configuration of the centerline. Bending springs restore a straight resting configuration [BLM04, SLF08].

Other approaches employ geometric curves such as splines to model the elastic rod [QT96, LMGC02, KJM08], having the advantage that geometric torsion can be taken into consideration. The geometric torsion measures the deviation of the curve to a planar configuration, and can be computed from the N control points alone.

However, the consideration of the material torsion complicates the problem. This comes from the observation that the $3N$ degrees-of-freedom (DOFs) provided by the N control points do not suffice to describe the configuration of the rod. Instead, additional DOFs are necessary to express the orientation of the centerline of the rod. An elastic rod with an oriented centerline is termed “Cosserat rod”. Consequently, Cosserat rods can reproduce both bending and twisting deformation (see Fig. 3.1). The difficult interplay between these deformation modii, and the resulting restitution equations have first been described by the Cosserat brothers in the 19th century. Since then, many works in both mechanics and computer graphics have been proposed that model and animate Cosserat rods [Keh97, CLW06, Pai02].

In this chapter, CORDE, a Cosserat Rod Element approach, is presented. The

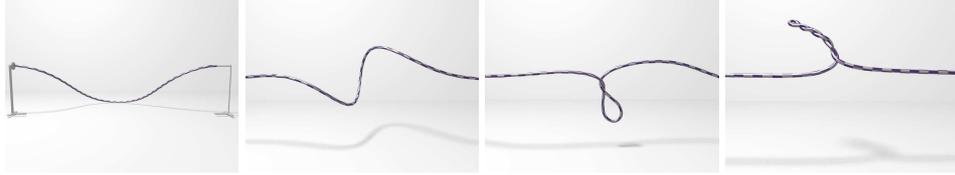


Figure 3.1: Dynamic looping phenomenon of a rod under torsional strain: A rod is spanned between two anchors, and its ends are clamped. A torque transducer on the left varies the end-to-end rotation of the rod. A bifurcation sequence is observed that results in a looping with an increasing number of self-contacts, also known as plectonemes [BWR*08]. This behavior can not be reproduced by approaches that do not consider material torsion.

term ‘CORDE’ comes from the french word for ‘rope’ and it is a dynamic deformation model for elastic rods with torsion. In contrast to previous Cosserat-based models that reconstruct the centerline by solving a boundary value problem (e. g. [Pai02, BAC*06]), the centerline of the rod is explicitly represented by a set of mass points. This simplifies the collision handling that can be carried out directly on the mass points. To model the torsion, a quaternion field is employed that represents the directed material frames of the discretized rod. A penalty method is employed to *adapt* the material frames to the centerline. By employing FE methods to compute the elastic energies in the discrete setting, the deformation becomes independent from the underlying discretization and enables adaptive methods, as illustrated in Chapter 4. To compute the dynamic evolution, the equations of motion are decoupled. This allows for an efficient local solve of the equations, resulting in a scheme that is linear in the number of nodes. Moreover, the treatment of material torsion allows to reproduce buckling and looping phenomena, as illustrated in Fig. 3.1.

Further, there exists a large class of objects that cannot be modeled with linear elastic rods alone. Instead, they are composed of elastic rods that are linked by elastic joints. For example, while the branches of a tree can be thought as elastic rods, the joints that link the branches and thereby forming the tree are not modeled as easily. In addition, in many cases, it would be desirable to link an elastic rod to itself, thereby forming a ring. These rings could be further combined to large networks with complex topologies, as illustrated in Fig. 3.2.

Therefore, an extension of CORDE, termed *Cosserat nets*, is discussed. The Cosserat nets allow for the simulation of networks of elastic rods. The struts in the networks are simulated as elastic rods. A methodology to model the elastic joints is then proposed. These joints have a given resting configuration. If the adjacent struts are rotated relatively to each other, then bending and twisting moments restore the reference shape. The representation comes without additional constraint equations and allows for an efficient local solution.

This chapter is organized as follows. First, the Cosserat theory of elastic rods



Figure 3.2: Cosserat nets are graph-like structures that consist of elastic joints linked by elastic rods. In this chapter, CORDE is employed to model such objects. Cosserat nets have a broad spectrum of applications in the field of animation. (a) An animation of a chain modeled from linked elastic rings. (b) CORDE is suited for simulating coarse nets.

is concisely introduced in Sec. 3.1. Based on this material, the static simulation of CORDE is discussed in Sec. 3.2. A representation of orientation is proposed and the resulting continuous energy formulations are discretized accordingly. Later, the extension to the fully dynamic case is addressed in Sec. 3.3, which includes an immersed discussion on the approximation compromises accepted by CORDE. In Sec. 3.4, the one-dimensional CORDE deformation model is extended in order to model loops and branches. This allows to simulate complex networks. Further, in order to evaluate the behavior of CORDE, a comparison to a reference deformation model is carried out. The implementation of this physically-based, *geometrically exact* [SFR90] reference model is explained in Sec. 3.5. The evaluation of CORDE is found in Sec. 3.6. The goal is to illustrate that CORDE can be employed to reproduce the important non-linear mechanical effects that characterize elastic rods, despite of the approximative nature of the deformation model. The chapter concludes with a discussion on the deformation model in Sec. 3.7.

3.1 Cosserat theory of elastic rods

In this section, a brief introduction of the Cosserat theory of elastic rods is given. The goal is that the reader becomes acquainted with the concepts and the notation of the configuration of elastic rods. More information is found in the book of Antman [Ant95] and in several publications in the field of mechanics [Dic94, Keh97].

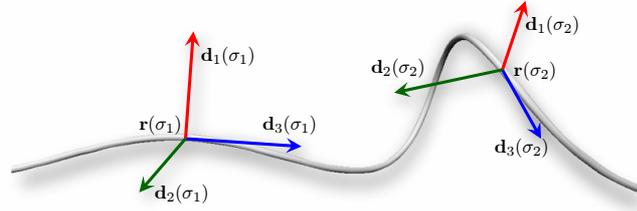


Figure 3.3: The configuration of the rod is defined by its centerline $\mathbf{r}(s)$. Further, the orientation of each mass point of the rod is represented by an orthonormal basis, called the directors. $\mathbf{d}_3(s)$ is constrained to be parallel to $\mathbf{r}'(s)$

3.1.1 Notation

If $f(s, t)$ is a function depending on the line parameter s as well as the time t , then $f'(s, t)$ denotes the spatial derivative $\frac{\partial f(s, t)}{\partial s}$, and $\dot{f}(s, t)$ denotes the temporal derivative $\frac{\partial f(s, t)}{\partial t}$.

3.1.2 Representation

An elastic rod can be thought as a long and thin deformable body. Since the volume of the rod is negligible compared to its length, the centerline of the rod is represented by a function $\mathbf{r} = \mathbf{r}(s) : [0, 1] \rightarrow \mathbb{R}^3$ that maps a line parameter s to a position in the space. If \mathbf{r} is considered to be a C^3 -differentiable function, the expressions for the bending and geometric torsion can be derived by employing differential geometry. The geometric torsion measures the deviance of \mathbf{r} from lying in a plane. For example, a circle in space has zero torsion and constant curvature.

However, with \mathbf{r} alone, the material torsion, i. e. the ‘roll’ of the curve around the centerline, cannot be represented. Thus, the concept of oriented curves has to be introduced: In each s , an orthonormal basis \mathbf{d}_k , $k = 1, 2, 3$ is thought, where the \mathbf{d}_k are called directors. The directors are *adapted* to the curve, which means that the third director $\mathbf{d}_3(s)$ is always parallel to the tangent $\mathbf{r}'(s)$ of the curve (see Fig. 3.3). The first and second directors \mathbf{d}_1 and \mathbf{d}_2 then indicate the orientation of the centerline. The basis \mathbf{d}_k is termed *material frame* of the rod. The directors $\mathbf{d}_k(s)$ constitute the columns of a rotation matrix $\mathbf{R}(s) \in \mathbb{R}^{3 \times 3}$, i. e. $\mathbf{R} = (\mathbf{d}_1 \ \mathbf{d}_2 \ \mathbf{d}_3)$. The parametrization of the rotation matrix (i. e. how to obtain the directors) is discussed in Appendix C.4.

To establish the stress-strain relation, a quantity has to be introduced that measures the rate of change in the position and orientation when traveling along the rod centerline. The rate of change in the position of the centerline is a strain vector $\mathbf{v} = (v_1 \ v_2 \ v_3)^T$. As common in the treatment of elastic rods, translational shearing is neglected, thus $v_1 = v_2 = 0$. v_3 is the stretch along the centerline, and it is measured as

$$v_3 = \|\mathbf{r}'\| \quad (3.1)$$

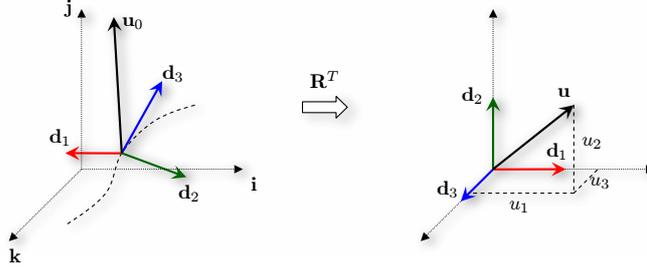


Figure 3.4: To obtain the quantities $u_k, k = 1, 2, 3$ related to bending and torsional strain, the Darboux vector \mathbf{u}_0 is rotated into the local frame, which is accomplished by multiplying \mathbf{u}_0 with \mathbf{R}^T .

Without loss of generality, the length of the unstretched rod is assumed to be 1. Consequently, $v_3 = 1$ for the unstretched rod.

Obtaining the orientational rate of change is slightly more involved. In differential geometry, this quantity is called the *Darboux vector*. It is a vector $\mathbf{u}_0 \in \mathbb{R}^3$, and it is assembled from the areas that are swept by the directors when proceeding from s to $s + \Delta s$:

$$\mathbf{u}_0(s) = \sum_{k=1}^3 \lim_{\Delta s \rightarrow 0} \frac{\mathbf{d}_k(s) \times \mathbf{d}_k(s + \Delta s)}{2\Delta s} \quad (3.2)$$

$$= \frac{1}{2} \sum_{k=1}^3 \mathbf{d}_k(s) \times \mathbf{d}'_k(s) \quad (3.3)$$

The Darboux vector measures the change of orientation in the reference frame. To relate the Darboux vector to bending and twist, it is rotated into the local frame (see Fig. 3.4),

$$\mathbf{u} = \mathbf{R}^T \mathbf{u}_0 \quad (3.4)$$

or, written in terms of the directors \mathbf{d}_k ,

$$u_k = \mathbf{d}_k \cdot \mathbf{u}_0, \quad k = 1, 2, 3 \quad (3.5)$$

with $\mathbf{u} = (u_1 \ u_2 \ u_3)^T$. When dealing with the dynamic case, quantities like velocity and angular velocity will come into play. The temporal derivative $\dot{\mathbf{r}}(s)$ of the centerline at s denotes the translational velocity of the centerline. The angular velocity ω_k of the directors around the k -th axis is obtained likewise as

$$\omega_k = \mathbf{d}_k \cdot \omega_0 \quad (3.6)$$

with

$$\omega_0 = \frac{1}{2} \sum_{k=1}^3 \mathbf{d}_k \times \dot{\mathbf{d}}_k \quad (3.7)$$

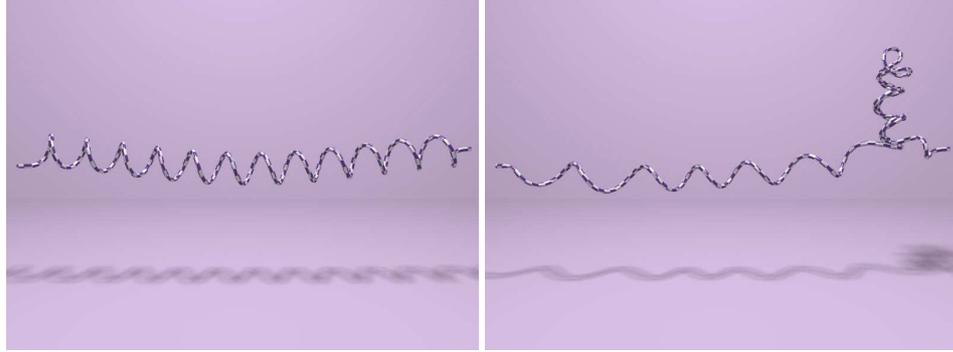


Figure 3.5: An elastic spiral is subject to continued torsional load. This load makes the spiral twisting around itself in order to balance the twisting and bending strains. The spiral is an elastic rod with an intrinsic curvature $\hat{\mathbf{u}} > \mathbf{0}$.

3.1.3 Constitutive relations

To investigate into the static equilibria of elastic rods, a stress-strain relationship has to be found. To accomplish this, a scalar-valued, convex energy function $V(v_3 - 1, \mathbf{u} - \hat{\mathbf{u}})$ with a minimum at $(0, \mathbf{0})$ is defined. Here, $\hat{\mathbf{u}} = (\hat{u}_1 \hat{u}_2 \hat{u}_3)^T$ corresponds to the intrinsic bending and torsion of the rod and allows to model e. g. spirals, as illustrated in Fig. 3.5. The stresses (i. e. the restitution forces) are then obtained by differentiating the energy with respect to the coordinates.

$V = V_s + V_b$ is the sum of a (translational) stretch energy V_s and a bending energy V_b . The stretch energy V_s is defined to be a quadratic form in the stretch v_3

$$V_s = \frac{1}{2} \int_0^1 K_s (v_3 - 1)^2 ds \quad (3.8)$$

K_s is the stretching stiffness constant that is computed from a stretching Young's modulus E_s and the radius r with $K_s = E_s \pi r^2$. Likewise, the bending energy V_b is a quadratic form in the strain rate vector \mathbf{u} ,

$$V_b = \frac{1}{2} \int_0^1 (\mathbf{u} - \hat{\mathbf{u}}) \cdot \mathbf{K} (\mathbf{u} - \hat{\mathbf{u}}) ds \quad (3.9)$$

The matrix $\mathbf{K} = (K_{kk}) \in \mathbb{R}^{3 \times 3}$ is the stiffness tensor. Since the rod is assumed to have a uniform cross-section, the off-diagonal terms in \mathbf{K} can be neglected. From textbooks on mechanics, the diagonal entries of \mathbf{K} are obtained as

$$K_{11} = K_{22} = E \frac{\pi r^2}{4}, \quad K_{33} = G \frac{\pi r^2}{2} \quad (3.10)$$

with E denoting the Young's modulus governing the bending resistance, G denoting the shear modulus governing the torsional resistance, and r denoting the radius of the rod's cross-section.

Balance of strain rates At this point, it is worth making a note on the balance of the strain rates. By minimizing V , the strain rates u_k and v_3 are balanced. Thus, if an elastic rod suffers from torsional strain, then this strain will immediately balance, resulting in a bending deformation. This coupling between the strain rates is responsible for the looping phenomenon of elastic rods under torsional strain (see Fig. 3.5). Now consider a perfectly straight rod that is subject to torsional load (Fig. 3.6, left). In which direction does it start to bend? The answer is that it will stay in the straight configuration: it is in an unstable equilibrium. As soon as the centerline is slightly disturbed, it will immediately pop out, striving towards a stable equilibrium between the torsion, bending and stretching strains (Fig. 3.6, right). This experiment provides a simple way to evaluate the plausibility of an (extensible) elastic rod deformation model. In Sec. 3.6, it is illustrated that the CORDE model exhibits this behavior. In contrast, approaches that do not treat bending and torsion in a unified manner such as [KPGF07, SLF08] cannot reproduce this behavior.



Figure 3.6: A perfectly straight rod that is subject to a torsional load is in an unstable equilibrium (left). As soon as the centerline is slightly disturbed, the rod starts to bend until an equilibrium between torsion, bending and stretching strain is reached (right). Approaches that do not treat bending and torsion in a unified manner cannot reproduce this behavior. This experiment provides a simple way to verify the physical plausibility of an elastic rod deformation model.

3.2 The CORDE model

In the previous section, the continuous deformation energies, being functions of the strain rates, have been spelled out. In this section, a discretization of the rod into control points is proposed. Then, the formulations for the deformation energies per element are derived. To accomplish, finite element methods are employed.

3.2.1 Discretization

The centerline of the rod is discretized into N spatial control points $\mathbf{r}_i \in \mathbb{R}^3, i \in [1, N]$. Thus, there are $N - 1$ centerline segments. To represent the orientations of these centerline segments, $N - 1$ material frames $\mathbf{R}_j \in \mathbb{R}^{3 \times 3}, j \in [1, N - 1]$ are additionally employed, as illustrated in Fig. 3.7. In the subsequent subsection, it is shown how to parameterize the rotation matrices. The k -th director of the j -th material frame is denoted by $\mathbf{d}_k(\mathbf{R}_j)$, which is obtained as the k -th column of the rotation matrix \mathbf{R}_j .

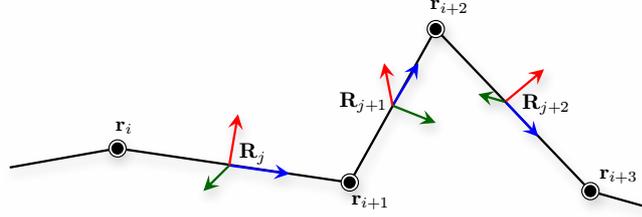


Figure 3.7: The centerline of the rod is discretized into nodes \mathbf{r}_i . In addition, $N - 1$ material frames \mathbf{R}_j are considered to express the orientation of the center segments.

The spatial derivative \mathbf{r}'_i of the centerline is obtained as the forward difference $\mathbf{r}_{i+1} - \mathbf{r}_i$ divided by the current segment length $\|\mathbf{r}_{i+1} - \mathbf{r}_i\|$. By assuming a high stretching stiffness, the current segment length can be replaced by its resting length $l_i = \|\mathbf{r}_{i+1}^0 - \mathbf{r}_i^0\|$, thus

$$\mathbf{r}'_i \approx \frac{1}{l_i} (\mathbf{r}_{i+1} - \mathbf{r}_i) \quad (3.11)$$

3.2.2 Parametrization of the rotation matrix

In this section, the representation of the material frames \mathbf{R}_j is discussed. From the material frames, the Darboux vectors that are related to the bending and twisting strains can in turn be computed.

From a theoretic point of view, it is possible to reconstruct \mathbf{R}_j from the centerline segment $(\mathbf{r}_i, \mathbf{r}_{i+1})$ and a single roll parameter $\varphi_i \in \mathbb{R}$. However, this cannot be done without considering the temporal coherence. A simple setup (Fig. 3.8) illustrates this problem: Consider an elastic rod whose left end point stays fixed. The first director \mathbf{d}_1 initially points towards the positive y-axis. If this rod is rotated around the z-axis, then one would intuitively end up in the first director \mathbf{d}_1 pointing towards the negative y-axis. If, in contrast, this rod is rotated around the y-axis, then the first director should intuitively keep pointing upwards. Still, in both cases, the rod has not been subject to any roll torques, thus φ will not have changed its value. As a consequence, the material frames cannot be computed unambiguously. Even worse, considering that the segment orientations in a deformed rod might cover the whole space $SO(3)$, a globally coherent material direction cannot be computed.

There do not exist straight-forward solutions to circumvent this problem. An approach that considers infinitesimal transformations has been proposed by Boyer and Primault [BP04]. Up to now, the approaches in [TGAB07] and [BWR*08] are the only ones in the field of computer graphics that employ the control points along with a roll angle as DOFs. Their approaches have the advantage that the computation of the strain rates does not rely on globally coherent directors. Instead, they compute the strain rates as the sum of the geometric torsion and the spatial change of material torsion. However, in order to visualize the rods in [TGAB07],

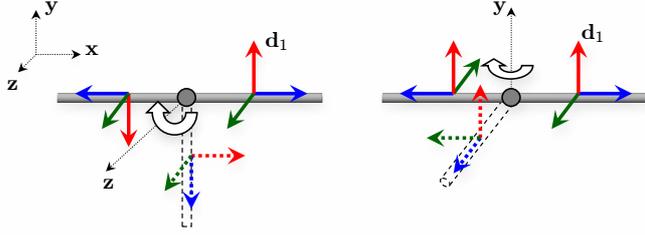


Figure 3.8: It is not possible to compute an unambiguous material direction without considering the temporal course of the simulation. In both scenarios, the rod has not been subject to any rolling during the rotation. Still, the material directions of the target states differ.

they need to update the material frames. They propose to update the first frame of the rod by employing temporal coherence, and then to propagate the material frames along the rod.

In contrast, here, a different approach is proposed that avoids the problem of explicitly updating the material frames in a natural way: Instead of reconstructing the frames from a roll angle, the material frames are expressed by *quaternions* $\mathbf{q}_j, j \in [1, N - 1]$. In contrast to other $\text{SO}(3)$ representations such as Euler angles, they provide a singularity-free parametrization of \mathbf{R}_j . The conceptual advantage of this approach is that since the material frames of the rod are now fully determined by the quaternions, the bending and twisting moments can be computed without considering the control points \mathbf{r}_i . The resulting torques accelerate the quaternions towards an energy-minimizing configuration. As a consequence, the material direction (which is given by the quaternions) is always globally coherent. In turn, this enables a straight-forward derivation of internal friction forces.

A quaternion $\mathbf{q} \in \mathbb{H}$ is a hyper-complex number with many applications in e. g. quantum mechanics or computer graphics. In this treatment, a quaternion can instead be simply considered as a four-dimensional vector $\mathbf{q} = (q_1, q_2, q_3, q_4)^T \in \mathbb{R}^4$ with the necessary constraint $\|\mathbf{q}\| = 1$. This constraint makes that the quaternions represent pure rotations. An immersed discussion on the quaternion algebra, a derivation of the spatial derivative \mathbf{q}' and the parametrization of the rotation matrix is given in Appendix C.

The strain rates u_k governing the constitutive relations can be computed as

$$u_k = 2\mathbf{B}_k \mathbf{q} \cdot \mathbf{q}', \quad k = 1, 2, 3 \quad (3.12)$$

where the $\mathbf{B}_k \in \mathbb{R}^{4 \times 4}$ are constant skew-symmetric matrices. The derivation of (3.12) is given in Appendix C.5.

Up to now, the centerline \mathbf{r}_i and the orientations \mathbf{q}_j have been treated as separated entities. However, in contrast to the rigid body simulation, the positions and the orientations cannot be simulated independently. Instead, the material frames parameterized by \mathbf{q}_j and the control points \mathbf{r}_i are coupled by the constraint that the

third director $\mathbf{d}_3(\mathbf{q}_j)$ is parallel to the tangent \mathbf{r}'_j , i. e.

$$\frac{\mathbf{r}'_i}{\|\mathbf{r}'_i\|} - \mathbf{d}_3(\mathbf{q}_j) = 0 \quad (3.13)$$

In the past, several approaches have been proposed to handle constrained mechanical systems, e. g. by employing Lagrangian multipliers or reduced coordinates. For the sake of efficiency and simplicity, the penalty method is employed, which has the further advantage that the constraints can be handled locally. Thus, the holonomic constraint (3.13) is modeled with a convex energy function

$$\begin{aligned} E_p[i] &= \frac{1}{2} \int_0^{l_i} \kappa \left(\frac{\mathbf{r}'_i}{\|\mathbf{r}'_i\|} - \mathbf{d}_3(\mathbf{q}_i) \right) \cdot \left(\frac{\mathbf{r}'_i}{\|\mathbf{r}'_i\|} - \mathbf{d}_3(\mathbf{q}_i) \right) d\xi \\ &= \frac{l_i}{2} \kappa \left(\frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} - \mathbf{d}_3(\mathbf{q}_i) \right) \cdot \left(\frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} - \mathbf{d}_3(\mathbf{q}_i) \right) \end{aligned} \quad (3.14)$$

By differentiating this penalty energy with respect to the coordinates \mathbf{r}_i and \mathbf{q}_j , penalty forces are obtained that maintain the constraints. κ is a spring constant that should be in the same order as K_s . The penalty energy is treated as an additional potential energy term. If the method of Lagrange multipliers is employed, the introduced stiffness could (at least partially) be removed. However, this would require to solve a banded linear system, whose inversion is in turn less efficient. Moreover, it does not solve the ‘ghost inertia’ problem related to the dynamic case that is discussed Sec. 3.3.

To enforce the quaternion unit constraint $\|\mathbf{q}_j\| = 1$, the simple method of *coordinate projection* is employed, i. e. the quaternions are normalized in each simulation step. Although this procedure does not conserve the energy, experiments indicate that it is a good compromise between efficiency and accuracy.

3.2.3 Finite element model

In the previous sections, the rod has been discretized into disjoint centerline elements $i = (\mathbf{r}_i, \mathbf{r}_{i+1})$ and disjoint orientation elements $j = (\mathbf{q}_j, \mathbf{q}_{j+1})$. The DOFs of the rod are constituted by the $3N + 4(N - 1)$ DOFs of these control points and quaternions.

A finite element (FE) method usually solves the weak form of the static stress analysis for kinematically admissible coordinates \mathbf{r}_i and \mathbf{q}_j ,

$$\int_0^1 V(v_3 - 1, \mathbf{u} - \hat{\mathbf{u}}) ds - \sum_{i=1}^N \mathbf{r}_i \cdot \mathbf{f}_i = 0 \quad (3.15)$$

that satisfy the given boundary conditions. \mathbf{f}_i are the external forces acting on the points, external torques are neglected. Having in mind that the dynamic equilibrium of the rod finally has to be solved, the static solution is not discussed further. Instead, the FE method is employed to compute the restitution forces $\boldsymbol{\sigma}_i = \frac{\partial V[i]}{\partial \mathbf{r}_i}$ and

torques $\tau_j = \frac{\partial V[j]}{\partial \mathbf{q}_j}$ per element. These forces and torques accelerate the elastic rod towards the static equilibrium. The goal is to derive the potential energy $V[i]$ and $V[j]$ per centerline element i and orientation element j , respectively. By differentiating these energies with respect to the local coordinates, the forces and torques acting on the control points and material frames are obtained.

To compute the per-element potential energy, the unknown displacement field has to be interpolated within the elements. For simplicity, constant shape functions are employed that interpolate the coordinates \mathbf{r}_i and \mathbf{q}_j linearly and result in constant stretch- and strain rates within the elements. They result in particularly simple expressions when integrated over the element length. Experiments indicated that higher-order shape functions do barely increase the accuracy of the model. Further, constant shape functions are employed to interpolate the spatial derivatives \mathbf{r}'_i and \mathbf{q}'_j , i. e. they are assumed to be constant within the elements.

The stretch energy $V_s[i]$ per centerline element i is obtained by integrating the stretch over the element length l_i

$$\begin{aligned} V_s[i] &= \frac{1}{2} \int_0^{l_i} K_s (\|\mathbf{r}'_i\| - 1)^2 d\xi \\ &= \frac{1}{2} l_i K_s \left(\frac{1}{l_i} \sqrt{(\mathbf{r}_{i+1} - \mathbf{r}_i) \cdot (\mathbf{r}_{i+1} - \mathbf{r}_i)} - 1 \right)^2 \end{aligned} \quad (3.16)$$

Likewise, the bending energy per orientation element j is obtained by integrating over the element length l_j

$$\begin{aligned} V_b[j] &= \frac{1}{2} \int_0^{l_j} \sum_{k=1}^3 K_{kk} (2\mathbf{B}_k \bar{\mathbf{q}}_j \cdot \mathbf{q}'_j - \hat{u}_k)^2 d\xi \\ &= \frac{l_j}{2} \sum_{k=1}^3 K_{kk} (\mathbf{B}_k (\mathbf{q}_j + \mathbf{q}_{j+1}) \cdot \frac{1}{l_j} (\mathbf{q}_{j+1} - \mathbf{q}_j) - \hat{u}_k)^2 \end{aligned} \quad (3.17)$$

By employing FE methods to compute the restitution forces, it is guaranteed that the static equilibria are independent from the underlying discretization (at least as long as the DOFs of the rod suffice to reproduce the deformation). This property is particularly interesting in the context of adaptive simulations, as pointed out in Chapter 4.

Summarizing, the potential energy per element have now been computed. By symbolically differentiating the sum $V[i] = V_s[i] + E_p[i]$ with respect to \mathbf{r}_i and \mathbf{r}_{i+1} , and by differentiating the sum $V[j] = V_b[j] + E_p[j]$ with respect to \mathbf{q}_j and \mathbf{q}_{j+1} , the restitution forces and torques that act on \mathbf{r}_i , \mathbf{r}_{i+1} , \mathbf{q}_j and \mathbf{q}_{j+1} are obtained. In the next section, a description on the computation of the trajectories of the control points and quaternions that are governed by the restitution forces is given.

3.3 The dynamic evolution

In this section, the focus is on the effects that come into play when the dynamic evolution of the rod is considered. First, it is shown how to obtain the simplified

rod dynamics that avoid the solution of a system of equations. Then, a way to compute internal friction forces is proposed that plausibly damp relative motion within the rod. Finally, the artifacts that are caused by the approximative dynamic CORDE model are discussed, and it is shown how to reduce those artifacts.

3.3.1 Numerical time-integration

The dynamic equilibrium of a mechanical system is usually obtained by feeding the Lagrange equation of motion, a variational formulation of the equations of motion for the generalized coordinates \mathbf{g} . This results in a system of the form $\mathbf{M}(\mathbf{g})\ddot{\mathbf{g}} - \mathbf{k}(\mathbf{g}, \mathbf{g}) = \mathbf{f}$, where \mathbf{M} is the mass-matrix, \mathbf{k} a stiffness function that computes the restitution forces, and \mathbf{f} the external forces. When considering rotating mechanical systems, the mass-matrix \mathbf{M} will be block-diagonal sparse or dense. In any case, inverting the mass-matrix requires the iterative solution of a system of equations which can be expensive.

However, an intrinsically simpler way to compute the dynamics can be obtained by looking at the solution that has been presented in the previous section. The rod can be considered as a chain of control points \mathbf{r}_i , and a chain of quaternions \mathbf{q}_j conforming to the material frames. These chains are loosely coupled by penalty forces that accelerate them towards a valid configuration. Still, the dynamic evolution of the mass points is decoupled from the dynamic evolution of the quaternions. As a consequence, the control points can be time-integrated as if they were mass-points in a mass-spring system, and the quaternions as if they represented the orientation of rigid bodies.

The masses $m_i = \frac{1}{2}\rho\pi r^2(l_{i-1} + l_i)$ are assumed to be lumped in the control points \mathbf{r}_i , which conforms to the mass being constant within the elements, as pointed out in [WDGT01]. Here, ρ is the density of the rod, and r is its the radius. Then the numerical evolution of the mass points is governed by the equation of motion $m_i\ddot{\mathbf{r}}_i - \boldsymbol{\sigma}_i = \mathbf{f}$, where $\boldsymbol{\sigma}_i$ is the sum of the internal forces on \mathbf{r}_i . To evolve the mass points, a semi-implicit Euler scheme is employed.

For the evolution of the quaternions, one can stick to the wealth of literature on rigid body simulation. It is most common to express the angular velocities $\boldsymbol{\omega}_j \in \mathbb{R}^3$ along with the quaternions \mathbf{q}_j as state variables [SM06]. Then, the state equations for the quaternions \mathbf{q}_j are obtained as

$$\begin{aligned}\dot{\boldsymbol{\omega}}_j &= \mathbf{J}^{-1}(\tilde{\boldsymbol{\tau}}_j - \boldsymbol{\omega}_j \times \mathbf{J}\boldsymbol{\omega}_j) \\ \dot{\mathbf{q}}_j &= \frac{1}{2}\mathbf{Q}_j \begin{pmatrix} \boldsymbol{\omega}_j \\ 0 \end{pmatrix}\end{aligned}\tag{3.18}$$

Here, \mathbf{Q}_j is the quaternion matrix that allows to write the quaternion multiplication consistently as a matrix-vector multiplication. Details are found in Appendix C. Further, $\tilde{\boldsymbol{\tau}}_j \in \mathbb{R}^3$ are the internal torques that govern the restitution. One has to consider that the torques $\boldsymbol{\tau}_j \in \mathbb{R}^4$ with $\boldsymbol{\tau}_j = \frac{\partial V_b[j]}{\partial \mathbf{q}_j}$ are dual to the quaternions. The translation into the spatial torques $\tilde{\boldsymbol{\tau}}_j$ is accomplished by a multiplication with the

transposed quaternion matrix \mathbf{Q}_j^T ,

$$\begin{pmatrix} \tilde{\boldsymbol{\tau}}_j \\ 0 \end{pmatrix} = \frac{1}{2} \mathbf{Q}_j^T \boldsymbol{\tau}_j \quad (3.19)$$

Details can be found in [SM06]. $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of the rod segment. For rods, it is assumed to be a diagonal matrix $\mathbf{J} = (J_{kk})$, $k = 1, 2, 3$ with $J_{11} = J_{22} = \frac{1}{4} l_j \rho \pi r^2$ and $J_{33} = \frac{1}{l_j} (J_{11} + J_{22})$. Further, for diagonal matrices, the coriolis force term $\boldsymbol{\omega}_j \times \mathbf{J} \boldsymbol{\omega}_j = \mathbf{0}$. Thus, the simple quaternion state equations are obtained as

$$\begin{aligned} \begin{pmatrix} \dot{\boldsymbol{\omega}}_j \\ 0 \end{pmatrix} &= \frac{1}{2} \mathbf{J}^{-1} \mathbf{Q}_j^T \boldsymbol{\tau}_j \\ \dot{\mathbf{q}}_j &= \frac{1}{2} \mathbf{Q}_j \begin{pmatrix} \boldsymbol{\omega}_j \\ 0 \end{pmatrix} \end{aligned} \quad (3.20)$$

that are numerically integrated with a semi-implicit Euler scheme.

3.3.2 Modeling internal friction

In order to plausibly model elastic rods, the internal friction has to be considered. Internal friction forces damp relative motion in the rod. As Baraff and Witkin pointed out, friction forces that reduce stretch oscillations should confine themselves solely to relative stretch motions, and not to any other relative motions [BW98]. The same criterion applies for angular friction forces. Thus, a translational dissipation energy is defined per element i

$$D_t[i] = \frac{1}{2} \int_0^{l_i} \gamma_t \mathbf{v}_i^{(rel)} \cdot \mathbf{v}_i^{(rel)} d\xi = \frac{l_i}{2} \gamma_t \mathbf{v}_i^{(rel)} \cdot \mathbf{v}_i^{(rel)} \quad (3.21)$$

where γ_t is the translational friction coefficient. Moreover, $\mathbf{v}^{(rel)}$ is the relative velocity of the two control points constituting the centerline element, projected on the tangent \mathbf{r}' of the centerline,

$$\begin{aligned} \mathbf{v}_i^{(rel)} &= \frac{1}{\underbrace{\|\mathbf{r}'_i\|^2}_{\approx 1}} \mathbf{r}'_i (\dot{\mathbf{r}}'_i \cdot \mathbf{r}'_i) \\ &= \frac{1}{l_i^3} (\mathbf{r}_{i+1} - \mathbf{r}_i) ((\dot{\mathbf{r}}_{i+1} - \dot{\mathbf{r}}_i) \cdot (\mathbf{r}_{i+1} - \mathbf{r}_i)) \end{aligned} \quad (3.22)$$

where the simplifying assumption $\|\mathbf{r}'_i\| \approx 1$ has been made, which is valid for rods with a large stretching stiffness. The angular dissipation energy derives from the spatial derivative $\boldsymbol{\omega}^r$ of the angular velocity $\boldsymbol{\omega}^r$. Here, $\boldsymbol{\omega}^r = (\omega_1^r \ \omega_2^r \ \omega_3^r)^T$ is the angular velocity of the material frames expressed in the reference frame that is computed as

$$\boldsymbol{\omega}_k^r(\dot{\mathbf{q}}, \mathbf{q}) = 2\mathbf{B}_k^0 \dot{\mathbf{q}} \cdot \mathbf{q} \quad (3.23)$$

It is important to consider the angular velocities in the reference frame since quantities cannot be compared in different local frames. Then, the angular dissipation energy per orientation element j is obtained as

$$D_r[j] = \frac{1}{2} \int_0^{l_j} \gamma_r \omega^{tr}(\mathbf{q}_j, \mathbf{q}_j) \cdot \omega^{tr}(\mathbf{q}_j, \mathbf{q}_j) \quad (3.24)$$

$$\begin{aligned} &= \frac{1}{2} \int_0^{l_j} \gamma_r \sum_{k=1}^3 \left(\frac{1}{l_j} 2\mathbf{B}_k^0 \mathbf{q}_{j+1} \dot{\mathbf{q}}_{j+1} - \frac{1}{l_j} 2\mathbf{B}_k^0 \mathbf{q}_j \dot{\mathbf{q}}_j \right)^2 d\xi \\ &= \frac{2}{l_j} \gamma_r \sum_{k=1}^3 (\mathbf{B}_k^0 \mathbf{q}_{j+1} \dot{\mathbf{q}}_{j+1} - \mathbf{B}_k^0 \mathbf{q}_j \dot{\mathbf{q}}_j)^2 \end{aligned} \quad (3.25)$$

A similar dissipation energy formulation is proposed in [WH04]. The damping forces are now derived by symbolically differentiating the dissipation energies with respect to the coordinates. Maple v9.5¹ is employed to carry out the calculations.

This relatively straight-forward derivation of internal friction is enabled since the quaternions allow to symbolically express the material direction and the angular velocity of the material in a globally coherent manner. This is in contrast to the approach of [TGAB07] that does not symbolically derive a globally coherent material direction. Thus the computation of angular friction is intrinsically more difficult. Consequently, they do not model internal friction but employ the viscous friction that is a side-effect of their implicit solver.

3.3.3 Benefits, limitations and future work

The proposed approach computes the dynamic evolution of the rod as a loosely coupled motion of a chain of masses and a chain of quaternions. The explicit simulation of the quaternions enables a convenient computation of the material direction, which allows a direct solve of the equations of motion, resulting in a comparably small implementation effort. Further, a consistent explicit representation of the material direction simplifies the computation of internal friction and also eases a texture-mapped visualization of the rod.

However, the proposed approach comes with an artifact that is denoted as ‘ghost inertia’. Consider the simple pendulum depicted in Fig. 3.8. By swinging around the vertical equilibrium position, the rod is subject to a rigid body rotation. Still, the material frames need to evolve continuously in order to adapt to the continuously changing orientation of the rod centerline. This evolution is solely governed by the parallel-constraint (3.14). The noticeable behavior is that the rod movement is subject to a rotational drag, i. e. a ghost inertia. This rotational drag increases with increasing weight of the material frames. The ghost drag is only noticeable if the rod is subject to rigid body rotation modes.

To overcome this problem, the primary goal is reconsidered, namely that the bending inertia of the rod is exclusively governed by the centerline, while the torsion inertia is governed by the quaternions. This conforms to an inertia tensor \mathbf{J}

¹www.maplesoft.com

with $J_{11} = J_{22} = 0$. Obviously this inertia tensor is singular. However, by scaling J_{11} and J_{22} with a factor $\alpha < 1$, one gets a DOF that controls the amount of ghost inertia one is willing to accept: By choosing a small $\alpha < 10^{-2}$, the ghost inertia is hardly noticeable, but the numerical integration requires a smaller time step due to the increased stiffness, which can in turn be remedied by a multi-step integration scheme for the quaternions. For the experiments, $\alpha = 10^{-2}$ has been chosen.

This section is finished with a short summary on the rod dynamics in previous approaches, and an outlook to future work. Most previous approaches compute a static or a quasi-static equilibrium configuration [Pai02, WH04] and thus do not consider dynamic effects. The approach of Bergou *et al.* [BWR*08] treats the centerline as fully dynamic, but the material frames are updated in a quasi-static way. Thereby, they neglect the moment of inertia of the rod's cross-section. The dynamic approaches of Bertails *et al.* [BAC*06] and Chang *et al.* [CSZ07] have in common that the DOFs of the model are provided by the strain rates u_k and v_3 . They then reconstruct the centerline, given the position of the first node of the rod. Both approaches propose to derive the dynamics with the Lagrangian formalism, and both approaches give the kinetic energy governing the dynamic evolution as $T = \int_0^L \rho A \dot{\mathbf{r}}^2 ds$. Again, both approaches neglect the moment of inertia of the rod's cross-section. Theetten *et al.* [TGAB07] approximates the kinetic energy as $T = \int_0^L (\rho A \dot{\mathbf{r}}^2 + J_{33} \dot{\theta}^2) ds$, where θ is the roll of the material cross-section. Thus this formulation considers the inertia of the torsion. Still, they neglect the bending and twisting moments of the angular cross-section. In contrast, the correct kinetic energy is e.g. given by Antman [Ant95] as $T = \int_0^L (\rho A \dot{\mathbf{r}}^2 + \boldsymbol{\omega} \cdot \mathbf{J} \boldsymbol{\omega}) ds$, where the second term conforms to the rotational kinetic energy of the rod cross-section.

It is not easy to quantify the error in the discrete case that is introduced by neglecting the rotational kinetic energy. To seek for an analogy, computing the kinetic energy of a discretized rigid body by summing the translational energies of the nodes provides but an approximation of the rotational energy. In order to quantify the approximation of the CORDE model, a Cosserat rod element model recently proposed by Cao *et al.* [CLW06] has been implemented that considers the exact kinetic energy (see Sec. 3.5), and compared to the presented model. The results of this comparison are given in Sec. 3.6.2. However, it has to be underlined that there is still room for improvement in terms of simulating the dynamics of elastic rods in an accurate and efficient way.

3.4 Cosserat nets

In this section, a novel extension of the previously described approach to branched and looped structures is described. The motivation comes from the observation that the CORDE model basically computes per-element forces on control points and orientations. The resulting restitution forces on the control points and material frames result from the contributions of the adjacent elements. Thus it is obvious that the last and the first control point of a rod could be merged, thereby forming a

closed loop. Moreover, instead of restricting a control point to be adjacent to two centerline elements, T-junctions could be allowed, i. e. that the rod branches in a control point.

3.4.1 Theoretical background

Here, a Cosserat net is defined to be a set of elastic joints, linked by elastic rods. The joints are elastic in the sense that the linked elastic rods are allowed to bend and twist relative to each other. The constitutive restitution is governed by given elasticity moduli. Intuitively, the static equilibrium of e. g. a T-junction with one 180° and two 90° angles is demanded to conform to the static equilibria of rods with intrinsic sharp curvatures of 180° and 90° , respectively.

Branched elastic rods have rarely been considered in the literature. In [NR04], Nadler and Rubin have proposed to model three-dimensional frames from elastic beams joined by Cosserat points. A Cosserat point can be understood as a deformable cuboid, where the deformation moduli are decoupled into homogenous and into inhomogenous deformations. More information on the theory of Cosserat points can be found in [Rub00]. In the field of computer graphics, Stam has addressed the dynamics of flexible tree-like structures in wind fields [Sta97]. The deformed positions of the child branches are obtained from the relative displacements with respect to their parent branches. Torsional torques are neglected, which is in contrast to the proposed approach.

3.4.2 Approach

Here, an intuitive approach is proposed to model these networks of elastic rods and elastic joints. In contrast to [NR04], the focus is on the plausible and efficient animation of such networks. In contrast to [Sta97], the resulting representation allows loops and torsional deformation.

First, it is demanded that joints always coincide with control points of the discretized rods. This can be start points, end points, or arbitrary control points within the rod. Notice that this does not limit the generality of the approach because control points can be adaptively inserted without making the global deformation behavior change, as illustrated in Chapter 4. Then, the Cosserat net are assembled from single elastic rods that share one or more control points.

The stretch forces and the penalty forces for the parallel-constraint are computed by integration over the centerline elements. Consequently, these forces can be computed for all centerline elements of the network. The restitution forces on the control points are obtained by summing the contributions of all adjacent elements. It is easy to verify that the proposed discretization of the domain of the Cosserat net is disjoint, thus the computation of the stretch and penalty forces is physically plausible (see Fig. 3.9 left).

Things become more difficult if the bending forces are considered, because these forces are computed by integration over the orientation elements that link two

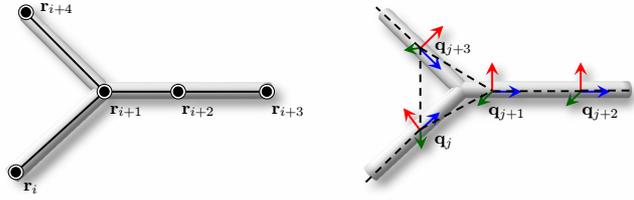


Figure 3.9: Left: A T-junction always conforms to a control point of the discretized rods. A Cosserat net is then assembled from the single elastic rods that share control points. Right: All pairs of adjacent segments are considered as orientation elements, depicted as dashed lines. Thus, for a T-junction, one ends up with the three orientation elements $(\mathbf{q}_j, \mathbf{q}_{j+1})$, $(\mathbf{q}_j, \mathbf{q}_{j+3})$ and $(\mathbf{q}_{j+1}, \mathbf{q}_{j+3})$ modeling the joint.

centerline elements (see Fig. 3.9 right). However, it is not possible to discretize a T-junction into disjoint orientation elements, as depicted in Fig. 3.9. One could, of course, omit the orientation elements in the T-junctions, but this would result in a joint that does not restore its resting configuration after deformation, which is not desirable.

Instead, a concept is proposed that is a straight-forward extension of the case of one-dimensional rods. In a joint, all pairs of adjacent centerline segments are considered, and these pairs are treated as orientation elements. For example, for a T-junction, one ends up in having three orientation elements (Fig. 3.9 right), and for a star with n adjacent segments, $\binom{n}{2}$ orientation elements are obtained. Further, these orientation elements have a discontinuous, intrinsic curvature that conforms to the resting angle between the adjacent segments. To compute the restitution torques in a joint, it is integrated over all orientation elements and the torques are summed per orientation node. While this concept lacks some physical accuracy, it meets the requirements in animation: an easy and fast solution that reproduces the behavior that is intuitively expected. Still, future work includes a more accurate handling of these joints.

3.5 Reference deformation model

In order to validate the statics and dynamics of the CORDE deformation model, a physically-based finite element deformation model for elastic rods has been implemented. The deformation model is inspired by a recent publication of Cao *et al.* [CLW06]. In contrast to CORDE, this deformation model employs the minimum number of DOFs to describe the configuration of an elastic rod, notably $3N$ DOFs for the centerline plus $N - 1$ DOFs to describe the material torsion. This results in a constraint-free representation of the rod where the material frame is always adapted to the centerline.

While this deformation model reproduces both the statics and the dynamics ac-

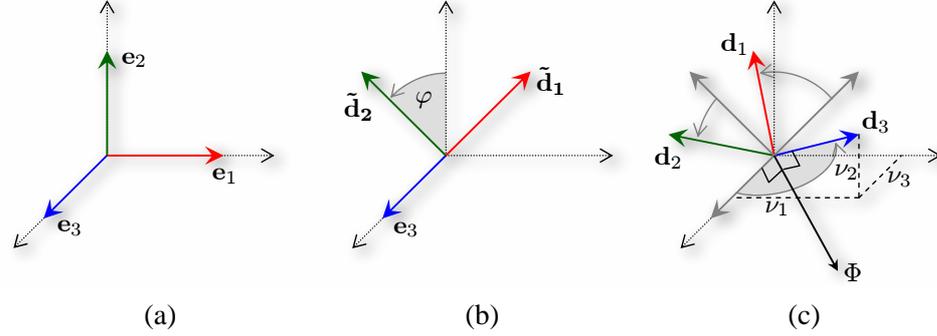


Figure 3.10: To obtain the directors \mathbf{d}_k from a line segment $(v_1, v_2, v_3)^T$ and a roll angle φ , the directors \mathbf{e}_k are first rotated about \mathbf{e}_3 by an angle φ . Afterwards, the resulting frame is rotated about the rotation axis Φ by an angle $|\Phi|$, resulting in the material frame \mathbf{d}_k .

curately, it has three striking drawbacks which make it of minor practical interest in animation: First, since the material frame is not iteratively updated during the course of the simulation, it suffers from the problem sketched in Sec. 3.2. As a consequence, a globally coherent material direction cannot be computed, which in turn disables intrinsic curvatures and coarse discretizations. Second, it suffers from a singularity at the poles, which results in numerical instability if the orientation of the rod segments approaches these poles. Third, the computation of both the restitution forces and the entries of the (configuration-dependent) mass matrix is extremely expensive, which in turn forbids interactive applications. Despite of these limitations, it models elastic rods accurately and allows to compare and evaluate CORDE with respect to the statics and dynamics.

In this section, the reference deformation model is shortly described. An extensive discussion is found in [CLW06]. The proceeding is as follows: The centerline of the rod is discretized into nodes \mathbf{r}_i . The ‘roll’ of the centerline segment $(\mathbf{r}_i, \mathbf{r}_{i+1})$ is expressed by a scalar DOF φ_i . Given the nodes \mathbf{r}_i and the roll angles φ_i , the material frame $\mathbf{d}_k, k = 1, 2, 3$ can be derived. By plugging the resulting expressions into the Lagrangian equation of motion, the force expressions and the mass matrix are obtained.

3.5.1 Parametrization of the rotation matrix

The most difficult thing is to obtain the directors. Let

$$(v_1, v_2, v_3)^T = ((\mathbf{r}_{i+1} - \mathbf{r}_i) \cdot (\mathbf{r}_{i+1} - \mathbf{r}_i))^{-\frac{1}{2}} (\mathbf{r}_{i+1} - \mathbf{r}_i) = \mathbf{d}_3 \quad (3.26)$$

be the components of \mathbf{d}_3 . Following [CLW06], the directors \mathbf{e}_1 and \mathbf{e}_2 are first rotated about \mathbf{e}_3 with the roll angle φ to obtain the directors $\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2$ and \mathbf{e}_3 (Fig. 3.10 (b)). Let the corresponding rotation matrix be $\mathbf{R}_a(\varphi)$. Now a rotation vector Φ is constructed that rotates \mathbf{e}_3 to \mathbf{d}_3 , i. e. Φ is a vector whose direction corresponds

to the axis of rotation, and whose length corresponds to the rotation angle. Consequently, Φ is perpendicular to the plane spanned by \mathbf{e}_3 and \mathbf{d}_3 (Fig. 3.10 (c)) and therefore parallel to the vector $-\nu_2\mathbf{e}_1 + \nu_1\mathbf{e}_2$. Further, the rotation angle is $|\Phi| = \arcsin \sqrt{\nu_1^2 + \nu_2^2}$, as can easily be verified in Fig. 3.10 (c). Together,

$$\Phi = \frac{\arcsin \sqrt{\nu_1^2 + \nu_2^2}}{\sqrt{\nu_1^2 + \nu_2^2}} (-\nu_2\mathbf{e}_1 + \nu_1\mathbf{e}_2) \quad (3.27)$$

The corresponding rotation matrix $\mathbf{R}_b(\nu_1, \nu_2, \nu_3)$ is now obtained by employing Rodrigues' formula [Stu64] and some algebraic transformations,

$$\mathbf{R}_b = \begin{pmatrix} \frac{\nu_1^2\nu_3 + \nu_2^2}{\nu_1^2 + \nu_2^2} & \frac{\nu_1\nu_2(\nu_3 - 1)}{\nu_1^2 + \nu_2^2} & \nu_1 \\ \frac{\nu_1\nu_2(\nu_3 - 1)}{\nu_1^2 + \nu_2^2} & \frac{\nu_2^2\nu_3 + \nu_1^2}{\nu_1^2 + \nu_2^2} & \nu_2 \\ -\nu_1 & -\nu_2 & \nu_3 \end{pmatrix} \quad (3.28)$$

The material frame $\mathbf{R} = (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$ is then obtained by composing the two rotations,

$$\mathbf{R}(\nu_1, \nu_2, \nu_3, \varphi) = \mathbf{R}_b(\nu_1, \nu_2, \nu_3)\mathbf{R}_a(\varphi) \quad (3.29)$$

Consequently, the i -th directors $\mathbf{d}_{k,i}$ depend on the 7 DOFs of the two control points $\mathbf{r}_i, \mathbf{r}_{i+1}$ and the roll angle φ_i .

The singularity at the vertical poles reveals in the divisor $\nu_1^2 + \nu_2^2$ in (3.28). Still, notice that the whole rod can be rigidly rotated in order to avoid the vertical poles. Unfortunately, an element-wise rotation in the spirit of co-rotational finite elements is not possible due to the orientation-dependent material frames.

3.5.2 Finite element model

In order to discretize the rod into disjoint finite elements, it has to be considered that the bending and torsion strain rates \mathbf{u}_i depend on the nodes $\mathbf{r}_i, \mathbf{r}_{i+1}$ and \mathbf{r}_{i+2} . Consequently, it is obvious to let the orientation element j start at $\frac{1}{2}(\mathbf{r}_i + \mathbf{r}_{i+1})$ and end at $\frac{1}{2}(\mathbf{r}_{i+1} + \mathbf{r}_{i+2})$. The centerline elements $i = (\mathbf{r}_i, \mathbf{r}_{i+1})$ are employed to compute the stretch strains and the mass matrices.

To derive the equations of motion for the reference model, the Lagrangian formalism [Gol81] is employed. The Lagrangian formalism is a variational formulation of the dynamic equilibrium of the rod that is governed by the differential equation

$$\underbrace{\frac{d}{dt} \frac{\partial T}{\partial \dot{g}_i}}_{\rightarrow \mathbf{M}, \mathbf{b}} + \underbrace{\frac{\partial V}{\partial g_i}}_{\rightarrow \mathbf{k}} = \mathbf{f} \quad (3.30)$$

$T = \sum_{i=1}^{N-1} T[i]$ is the total kinetic energy of the rod and $V = \sum_{i=1}^{N-1} V_s[i] + \sum_{j=1}^{N-2} V_b[j]$ is its total potential energy. Further the $g_i \in \{\mathbf{r}_1^T, \varphi_1, \dots, \varphi_{N-1}, \mathbf{r}_N^T\}$ are the DOFs of

the rod that are collected in a vector $\mathbf{g} = (g_1 \cdots g_{4N-1})^T$. The time-differentiated contribution of the kinetic energy yields the mass matrix \mathbf{M} and the coriolis force vector \mathbf{b} . The differentiation of the potential energy with respect to the coordinates yields the stiffness function \mathbf{k} . The external and contact forces are collected in the vector \mathbf{f} .

The stretching energy $V_s[i]$ follows from (3.8). The bending energy follows from integrating the bending and twisting strain densities over the orientation element length l_j ,

$$V_b[j] = \int_0^{l_j} \mathbf{u}_j \cdot \mathbf{K} \mathbf{u}_j d\xi \quad (3.31)$$

where the strains \mathbf{u}_j are obtained from (3.5). In contrast to [CLW06], the exact formulations (3.29) instead of a first-order approximation of the directors is employed.

Following Antman [Ant95], the kinetic energy density of a rod element sums from the translational kinetic energy density of the centerline and from the rotational kinetic energy density of the centerline's cross-section. The kinetic energy is then obtained by integrating the kinetic energy densities over the element length l_i ,

$$T[i] = \frac{1}{2} \int_0^{l_i} (\rho A \dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_i + \omega_i \cdot \mathbf{J} \omega_i) d\xi \quad (3.32)$$

where the angular velocities ω_i are obtained from plugging (3.7) in (3.6). To compute the temporal derivatives $\dot{\mathbf{d}}_k$, one must consider that the directors \mathbf{d}_k are functions of the time-dependent control points $\mathbf{r}(t)$ and roll angles $\varphi(t)$.

By assembling the mass matrix from the mass matrices of each element, and collecting the force contributions per nodes \mathbf{r}_i and roll angles φ_i , the matrix-vector equation

$$\mathbf{M}(\mathbf{g}) \ddot{\mathbf{g}} - \mathbf{k}(\mathbf{g}, \dot{\mathbf{g}}) - \mathbf{b}(\mathbf{g}, \dot{\mathbf{g}}) = \mathbf{f} \quad (3.33)$$

is obtained. \mathbf{M} is a symmetric banded (but not positive definite) matrix of size $4N - 1$ with a bandwidth of 13, $\mathbf{k}(\mathbf{g}, \dot{\mathbf{g}})$ is the stiffness function computing the restitution forces, and $\mathbf{b}(\mathbf{g}, \dot{\mathbf{g}})$ are the coriolis terms. Since the mass matrix depends on the control points and roll angles, it must be re-computed in each simulation step. To accomplish this, the WildMagic library² is employed to solve for the accelerations $\ddot{\mathbf{g}}$. Then, the equations of motion are time-integrated by employing a semi-implicit Euler scheme.

3.5.3 Implementation notes

To assemble the mass matrices and the force terms, the directors and the derived quantities have to be symbolically differentiated with respect to the coordinates. Since the directors are rather complicated terms (consider that already the v_k have a square root divisor), a computer algebra software must be employed to carry out the symbolic differentiation. For example, the resulting coriolis force term for the

²www.geometrictools.com

first component of \mathbf{r}_i consists of 204K multiplications, 26K divisions and 22K calls to trigonometric functions. By employing Maple's `codegen` package with the `tryhard` option, the term can be simplified to 355 multiplications, 5 divisions, 3 calls to trigonometric functions and 289 temporary variables. However, it is needless to say that the simulation of an elastic rod discretized into more than about 50 mass points can hardly be realized at interactive rates.

3.6 Results

In this section, an extensive and careful evaluation of the proposed deformation model CORDE is given. Further, the broad applicability of the proposed model is illustrated. The section is started with a qualitative evaluation of the deformation model. The purpose is to illustrate that the behavior of real rods can be reproduced, and that the global deformation is independent from the discretization. In the second part, a comparison to a geometrically exact, constraint-free deformation model for elastic rods is presented. The goal of the comparison is to quantify the benefits and limitations of CORDE. In the third part, the broad applicability of the deformation model is illustrated. It is shown that the approach is well-suited for interactive applications. Further, examples illustrate the conceptual advantages of the Cosserat nets.

All experiments have been performed on an Intel Xeon PC with 3.8GHz. The offline rendering is accomplished with a raytracer³, the real-time rendering with OpenGL. For rendering, the rods are skinned with a tubical, B-spline interpolated surface mesh.

3.6.1 Qualitative evaluation

In this section, it is illustrated that the looping phenomena that are induced by material torsion can be reproduced. Further, it is shown that the global deformation is independent from the underlying discretization, which is enabled by employing FE methods to compute the elastic forces. In this section, the focus is on the qualitative behavior, the quantitative evaluation is performed in the subsequent section. The material parameters are summarized in Tab. 3.2.

Looping In a first experiment, the global deformation of clamped rods is studied that occurs when the end-to-end rotation is varied. These experiments are inspired by recent research in the field of nonlinear mechanics [vdHNGT03]. The experimental setup is shown in Fig. 3.1 on the left. A torque transducer exerts a continued torsional rotation on the rod, while the right end of the rod stays clamped. The rod forms a looping with an increasing number of self-contacts. The number of self-contacts is obviously limited by the number of elements of the rod. The radius and the location of the loop is influenced by the ratio of bending and torsion

³PovRay for Windows v3.6, www.povray.org

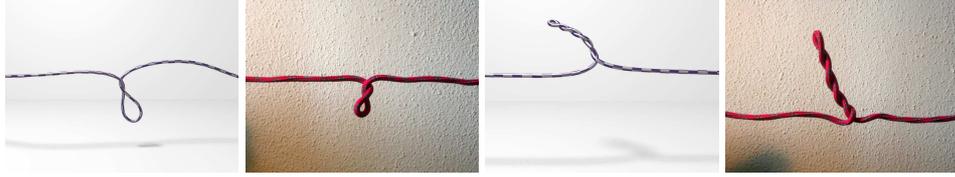


Figure 3.11: To validate our deformation model, the varied end-to-end rotation experiment is performed with a real rope. The global deformation is comparable to the simulated virtual rope.



Figure 3.12: Coiling up three clamped rods. The rods are discretized into 1000, 100 and 50 elements, illustrating that the global deformation is independent of the discretization.

stiffness [vdHNGT03]. The time step of the simulation is $h = 0.1\text{ms}$. The rod is discretized into 100 elements. To illustrate that this corresponds to the expected behavior, the same experiment is performed with a real rope. Fig. 3.11 illustrates the looping sequence of the rope under torsional strain. Although the material parameters of the real rope differ in magnitude, its global deformation is comparable to the simulated virtual rope.

Global deformation In a second experiment, the left ends of three rods are clamped to torque transducers. The transducers exert a continued rotation on the rods, thus coiling up the rods (see Fig. 3.12). The front rod is discretized into 1000 elements, the middle rod is discretized into 100 elements and the back rod is discretized into 50 elements. The time step is $h = 0.1\text{ms}$. The figures illustrate that the global deformation of the rods is independent of the discretization. Further, this experiment illustrates that the time to compute the dynamics of a CORDE rod is linear in the number of elements. Timings are summarized in Tab. 3.1.

# Elements	Force comp. [ms]	Integration [ms]
50	0.051	0.018
100	0.096	0.035
1000	0.91	0.33

Table 3.1: Timings for the force computation and integration of discretized rods. The timings indicate that the method is linear in the number of elements.

3.6.2 Quantitative evaluation

The goal of this section is to quantitatively evaluate both the static and the dynamic behavior of CORDE. To accomplish this, CORDE is compared to the geometrically exact reference deformation model presented in Sec. 3.5. The reference model bases on the FE method, is constraint-free and computes the correct dynamics in the sense of Antman [Ant95]. The expensive assembly of the mass matrix and the numerical instabilities at the poles make the reference model of little practical interest in computer graphics. However, by comparing CORDE to the reference model, it is shown how to parameterize CORDE such that the artifacts introduced by the approximations are minimized.

Static equilibrium It has been proposed to model the elastic rod from control points that define the configuration of the centerline, and from quaternions that define the material frames. The twisting and bending moments are computed by minimizing the respective energies. In turn, the twisting and bending energy results from the squared Darboux vector. Thus, by minimizing the Darboux vector, the twisting and bending moments are automatically coupled. Thereby, physical plausibility is gained, and the interesting buckling and looping phenomena subject to elastic rods can be reproduced.

To illustrate this property of the CORDE model, the experiment that has already been sketched in Sec. 3.1 is performed. Two elastic rods are considered, one is simulated with CORDE, the other with the reference model. The material parameters of the two identical rods are given in Tab. 3.2, and the time step of the simulation is 0.1ms. The two perfectly straight elastic rods are spanned between two anchors, and then the material frames are twisted around the centerlines such that the end-to-end rotation angle is $\frac{3}{2}\pi$. The rods are now in an unstable equilibrium. As long as the centerlines stay straight, the unstable equilibria are held. As soon as the centerlines are slightly disturbed, the rods start to buckle until stretch, bending and torsional strain are balanced.

Fig. 3.13 illustrates the comparison of the two deformation models in the unstable equilibrium (left) and after having reached a stable equilibrium that balances between the stretch and strain rates (right). The resting state of the CORDE model (blue texture) is less curved than the resting state of the reference model (red texture). The reason for this asymmetry is the constraint energy that absorbs some of

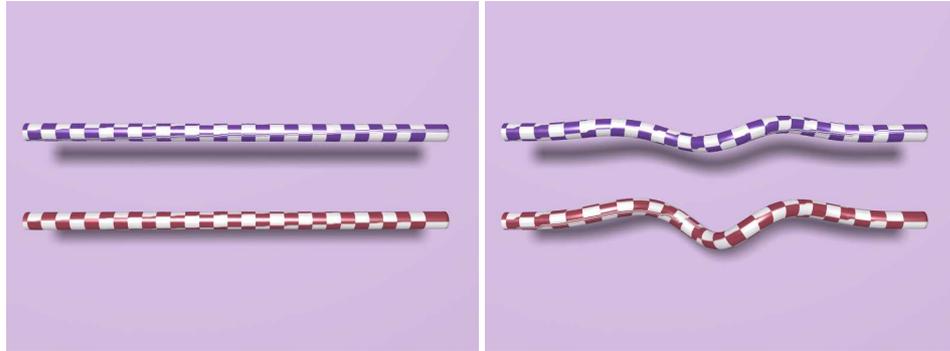


Figure 3.13: Two elastic rods under torsional load are in an unstable equilibrium (left). If the centerline is disturbed, then the rods buckle in order to balance bending, torsion and stretch strains (right). The blue rod is simulated with CORDE, the red rod with the reference model. The penalty method of CORDE absorbs energy, thus its resting state is less curved than the resting state of the reference model.

the total potential energy.

The temporal course of the simulation is illustrated in Fig. 3.14. The strain energy of the four deformation modes (first bending, second bending, torsion and stretch) plus the sum of those energies are plotted. The goal of this experiment is to illustrate the qualitative buckling behavior of CORDE. In the beginning, both rods are in the unstable equilibrium and the total energies conform to the torsion energies. If there were no external forces acting on the centerlines, they would keep the unstable equilibrium. Thus, in order to induce the buckling, a small gravitational force is exerted on the centerlines. The rods immediately pop out, striving towards stable equilibria states that balance between the four deformation modes. This is indicated in the total energies that have decreased after the rods found their equilibria. In contrast, the bending and stretching energies have increased, indicating that the centerlines are no longer straight. While the reference model accurately balances between stretch and bending modes, the CORDE deformation model has a much lower stretch energy. This comes from the penalty method that absorbs some energy. Still, it can be concluded that CORDE reproduces the buckling behavior which illustrates the physical plausibility of the proposed deformation model.

Dynamic equilibrium As illustrated in Sec. 3.3, the CORDE model simulates the elastic rod as a chain of mass points and a chain of quaternions. The coupling is realized with the penalty method. If now a rod is subject to a rigid body rotation, then it is accounted twice for the moment of inertia, once by accelerating the centerline, and once by accelerating the quaternions. This results in a visible drag or ‘ghost inertia’. To overcome this limitation, it has been proposed to scale the first two diagonal entries of the inertia tensor (that conform to the bending moments of the quaternions) by a factor $\alpha < 1$. The smaller the α -parameter, the better the

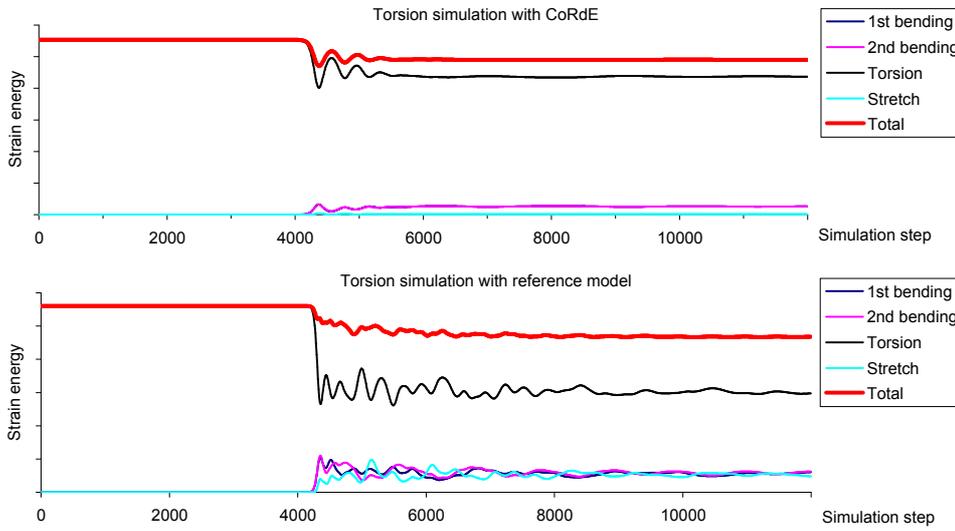


Figure 3.14: Temporal course of the torsion experiment. The goal of this experiment is to illustrate that by minimizing the Darboux vector, CORDE reproduces the buckling behavior, which underlines the physical plausibility of the deformation model. In fact, both deformation models reproduce the buckling behavior that exhibits in the sudden increase of the bending and stretch energy. The total deformation energy, however, has decreased after the buckling, indicating that the rods have reached a stable equilibrium. The reference model balances accurately between bending and stretch.

optimal solution is approximated, which conforms to quaternions that have only a moment of inertia when rotated around the third axis (conforming to torsion).

To illustrate this problem, an experiment of a simple elastic pendulum is performed: The upper end of two vertically hanging elastic rods are attached, and the lower ends are elongated (the elastic rods have the same physical properties as in the experiment described above, see Tab. 3.2). One rod is simulated with CORDE, the other with the reference deformation model. The time step of the simulation is 0.1ms. Upon releasing the lower ends, the elastic rods swing about their vertical equilibria. In this setting, the rods are subject to rigid body rotations, thus the CORDE rod reveals the ‘ghost inertia’ phenomenon. The CORDE rod is simulated with $\alpha = 1$, $\alpha = 10^{-1}$, $\alpha = 10^{-2}$ and $\alpha = 10^{-3}$. In Fig. 3.15, the elongations of the elastic rods are plotted over time. As the graphs illustrate, the smaller the α -values (i. e. the ‘lighter’ the quaternions), the better is the approximation of the dynamics of the CORDE rod. The reference model does not reveal this artifact since its dynamics are exclusively governed by its $4N$ DOFs (see Sec. 3.5). Notice that smaller α -values require more accurate numerical integration schemes, thus choosing an α -value is always a trade-off between efficiency and accuracy. A simple multi-step scheme has been implemented to time-integrate the quaternions in order to account for smaller α -values.

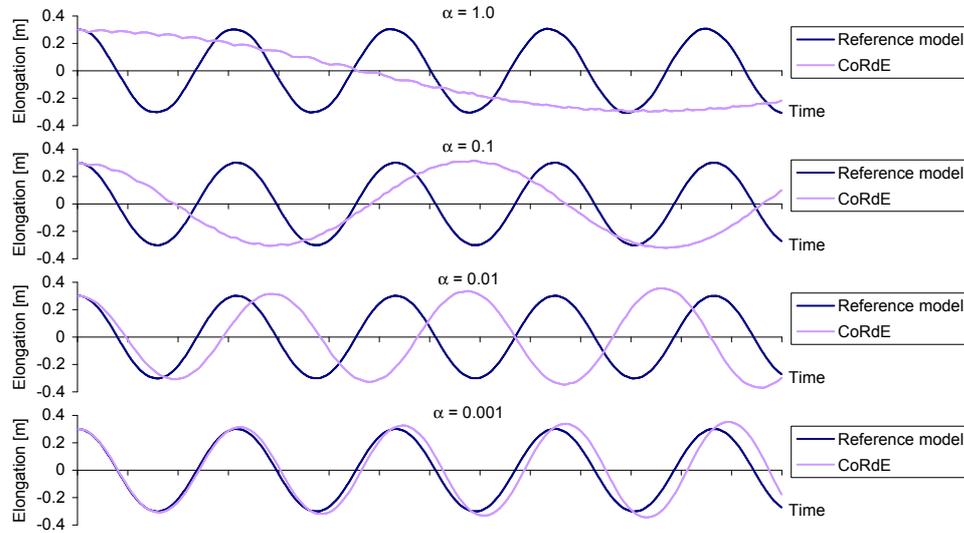


Figure 3.15: Temporal courses of the pendulum experiment. The visual drag influences the frequency of the pendulum. The smaller the α values, the better is the approximation of the correct dynamics of the reference model. In turn, small α -values require a more accurate integration scheme.

3.6.3 Application

The goal of this section is to give an idea on the wide applicability of CORDE. To accomplish this, both interactive and off-line experiments have been staged with rods that differ in their material properties. Further, experiments with Cosserat nets round out this section.

Stiff objects Due to the robustness of our deformation model, a broad variety of different materials can be simulated, i. e. highly flexible structures such as ropes as well as stiff objects such as bars. By defining an intrinsic bending and torsion, it is also possible to simulate pre-shaped objects such as springs. In these experiments, stiff wire-like objects are modeled whose material parameters are summarized in Tab. 3.2. Each rod is discretized into 50 elements. The stiff equations require a small time step $h = 0.05\text{ms}$. Fig. 3.16 illustrates the dynamic simulations.

Interactive rods Since the approach is linear in the number of elements, elastic rods can be simulated at interactive rates. Further, internal friction minimizes oscillations and makes the material look more realistic.

To illustrate that CORDE can be used in interactive applications, a simulation of a thread consisting of 100 elements is run. Its material parameters are summarized in Tab. 3.2. While one end of the thread is clamped, the user can freely interact with the centerline (Fig. 3.17). In this experiment, the user ties a knot into the thread. The time step is $h = 2\text{ms}$, and the configuration is rendered after

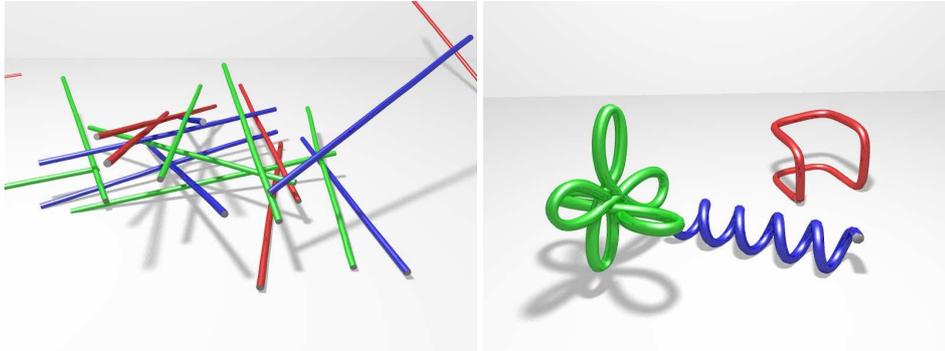


Figure 3.16: Dynamic simulation of stiff objects. Left: Stacking of rods. Right: Objects with intrinsic bending and torsion.

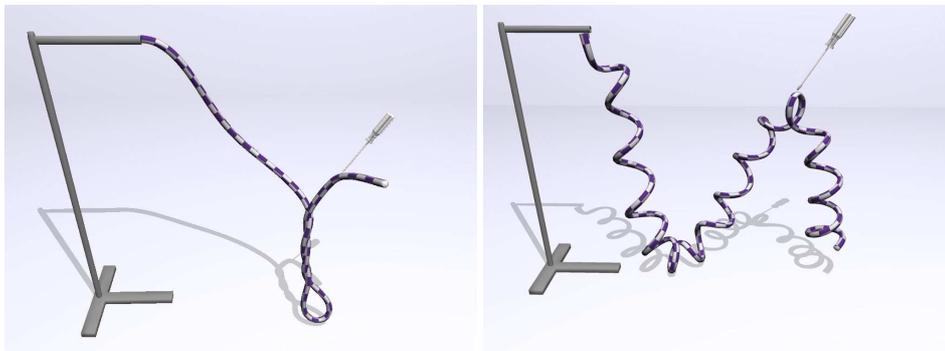


Figure 3.17: Interactive simulation of threads. The user can freely interact with the thread. Oscillations are minimized by modeling internal friction.

20 simulation steps. The simulation runs at 26 frames per second. As indicated in Fig. 3.17 on the right, it is also possible to interact with an intrinsically coiled thread.

Cosserat nets The Cosserat nets are an important extension of the CORDE deformation model to graph-like structures. The resulting networks of elastic rods linked by elastic joints allow to model a large variety of objects. To illustrate the wide applicability of the proposed deformation model, a series of experiments has been performed. It has to be underlined that the focus is more on the animation than on the mechanical simulation of the modeled objects.

Since CORDE handles both bending and twisting deformation modes, the Cosserat nets provide a high physical plausibility. To illustrate this, an experiment is performed where the end-to-end rotation of a spheroidal object depicted in Fig. 3.18 is varied. The four rods linking the two ends successively deform until an eight-shaped state is reached.

A net can be understood as a bundle of single threads, forming a planar two-

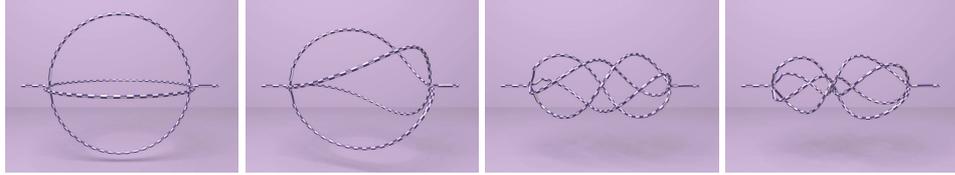


Figure 3.18: The end-to-end rotation of a spheroidal object that consists of four elastic rods is varied. This experiment illustrates the rich deformations that are achieved by employing CORDE to model such networks of joints and elastic handles.

	Ropes Fig. 3.11	Comparison Fig. 3.14	Wires Fig. 3.16	Threads Fig. 3.17
Length [m]	10	10	0.1	1
Radius r [m]	0.01	0.01	0.001	0.001
Density ρ [kg m^{-3}]	1300	1300	7860	1300
Young modulus E [MPa]	0.5	1	200	10^{-5}
Shearing modulus G [MPa]	0.5	4	100	10^{-5}
Stretch modulus E_s [MPa]	20	1000	100	0.02
Spring const. κ [10^3 kg m s^{-2}]	100	1000	300	0.1
γ_t [$10^{-6} \text{ kg m}^3 \text{ s}^{-1}$]	10	10	0.05	1
γ_r [$10^{-6} \text{ kg m}^3 \text{ s}^{-1}$]	1	1	0.01	0.1

Table 3.2: Material parameters for the objects simulated in the experiments. The parameter values have been determined by visually comparing the behavior of the simulated objects to their real counterparts.

dimensional structure. In contrast to knitted garment, the single threads are tied together in each crossing. Thus, a net can be modeled by employing CORDE, where a joint is considered in each crossing of the net. By simulating such Cosserat nets, the expected cloth-like behavior of the objects is obtained. While dense nets could also be modeled and (more efficiently) simulated by employing triangle topologies and thin shell methods, the simulation of coarse nets is enabled by employing CORDE, as depicted in Fig. 3.19. The computational effort to simulate bigger nets is remarkable, especially since the complexity of the collision detection grows quadratically with the number of segments. The coarse net in Fig. 3.19 a) is modeled from 0.8K points, 0.9K centerline elements and 1.1K orientation elements. Computing the deformation and integration takes 6.6ms, collision detection takes 62.3ms in average, and collision response is cheap with 0.3ms. The dense net in Fig. 3.19 b) is modeled from 1.2K points, 1.5K centerline elements, and 3K orientation elements. Computing the deformation and integration takes 14.5ms, collision detection takes 224ms, and collision response takes 10.6ms.

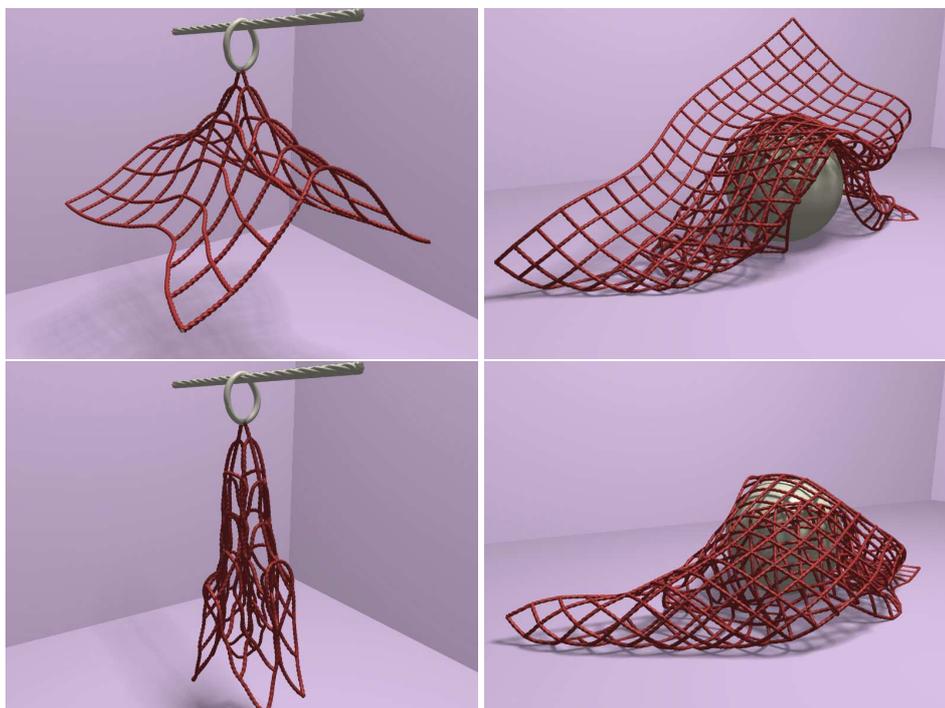


Figure 3.19: A net is modeled from a bundle of threads that are joined in each crossing. The deformation of the resulting objects share some similarities with cloth. However, coarse nets cannot be simulated with thin shell methods.

3.7 Discussion

In this section, a novel deformation model for elastic rods has been proposed. The starting point has been the observation that existing models for elastic rods are either implausible as they do not model the twisting and bending deformation in a consistent manner, or that they are inefficient and not suitable for representing complex contact configurations. Thus, the goal has been to find a way to simplify the underlying partial differential equations that govern the static and dynamic equilibrium of elastic rods. The result of these investigations is CORDE, a deformation model for elastic rods. CORDE combines the physically-based representation of a directed curve with the efficient modeling and simulation techniques that are commonly employed in computer graphics.

CORDE discretizes the centerline of the rod into a set of mass points. Further, the material direction field is discretized into a set of quaternions that conform to the material frames of the corresponding centerline segments. To compute the elastic forces, FE methods are employed in order to guarantee that the deformation is independent from the discretization. In order to adapt the material direction to the centerline, the penalty method is employed. While the penalty method lacks of physical correctness, its superior advantage is that the parallel constraint can be

maintained locally, without requiring the solution of a system of equations. Consequently, elastic rod can be directly simulated. That is, the equations of motion for the discretized centerline conform to the simple Newton's equations for a mass point, and the equations of motion for the discretized material frames conform to the angular equations of motion that are commonly employed in rigid body dynamics. Together, computing the dynamics of an elastic rod with CORDE is almost as simple as computing the dynamics of an ordinary mass-point system.

The second main benefit of CORDE is that the centerline is explicitly represented, which is in contrast to most previous works in the field of physically-based elastic rod simulation [Pai02, BAC*06, CSZ07]. This enables a straight-forward collision detection and response. For example, the Strands model of Pai reconstructs the centerline by solving a BVP, and constraints along the centerline are hard to maintain [Pai02]. In contrast, for the CORDE model, contact forces can directly act on the mass points, thereby enabling a predictor-corrector scheme to maintain the non-interference constraints. Such a response scheme is proposed in Chapter 6. The employ of an accurate response scheme allows to reproduce the looping phenomena (also known as plectonemes [BWR*08]) that characterize the deformation of elastic rods, and which have not been simulated in the field of computer graphics before.

Future work After all, CORDE does not reveal the end of the investigations in elastic rod modeling. There are a couple of issues that have not been solved convincingly, and that need future investigations. First, the employ of the penalty method influences the geometry of the static equilibrium. This comes from the fact that the penalty method 'absorbs' some of the total deformation energy of the rod. Consequently, less deformation energy is available in the bending modes of deformation, as illustrated in Fig. 3.14. Therefore, CORDE is not *geometrically exact* in the sense of Simo *et al.* [SFR90], i. e. the geometry of the rod and the kinematic equations are not treated exactly. To overcome this problem, the method of Lagrange multipliers could be employed. That means, instead of computing penalty forces, the acceleration components that violate the constraints could be directly canceled out, thereby resulting in workless forces. However, this requires in turn the solution of a banded matrix system.

The second drawback of CORDE reveals in the 'ghost inertia' problem. Rods that are subject to rigid body rotation modes suffer from a visible drag that comes from the inertia moment of the quaternions (being considered as rigid bodies). To overcome this problem, the quaternions had to be removed from the simulation, and torsion had to be modeled with one single roll parameter, in the spirit of [TGAB07, BWR*08]. This is also the motivation of the reference deformation model that has been presented in Sec. 3.5. The ghost inertia problem is independent from the penalty method, and cannot be solved by employing Lagrange multipliers alone.

Summarizing, CORDE is a simple and elegant deformation model for elastic rods that is able to reproduce the looping and buckling phenomena that are only

available if directed curves are considered. The dynamic simulation is efficient and allows for interactive applications. Still, the penalty method and the ghost inertia problem forbid applications which rely on mechanical accuracy.

Chapter 4

Adaptive variational simulation

In the previous section, a deformation model for elastic rods has been proposed. The goal has been to enable the physically plausible though efficient simulation of elastic rods. This has been enabled by simplifying the mechanical equations governing the static and dynamic equilibrium of the rod. To accomplish this, the rod has been discretized into mass points and orientation nodes, where the orientation nodes have been employed to provide the DOFs for the material directions. Then, the elastic forces have been computed locally for each element, and the force contributions are then summed per node. Finally, the equations of motion are solved per mass point and per orientation node. It follows that the complexity of the deformation model is linear in the number of DOFs.

Amongst others, elastic rods can be employed to represent threads or ropes in knot simulations. A knot is characterized by having a potentially complex contact geometry, while large parts of the elastic rod are practically undeformed. In order to represent the configuration of the knot, a large number of DOFs is necessary. This is, however, unfortunate, since a high resolution makes the deformation model less efficient.

To overcome this discrepancy, adaptive methods can be employed. That means, during the simulation, the DOFs are dynamically arranged, based on the temporally and spatially varying requirements for accuracy. In the knot, DOFs will be inserted in order to represent the complex configuration, while DOFs will be removed from the undeformed parts in order to improve the efficiency.

In this chapter, an adaptive framework for elastic rods is presented. The goal is to enable the reproduction of knots, as illustrated in Fig. 4.1. The framework is generic and can be applied to any hyper-elastic deformation model for elastic rods, i. e. a deformation that defines a necessarily convex strain energy function from which a stress-strain relationship can be derived. Both geometric and mechanical adaptation criteria govern the refinement of the rod. The *geometric adaptation criterion* considers the curvature $\|\mathbf{u}\|$ of consecutive centerline elements. Then, elements are bisected if the curvature is above a certain threshold u_{\max} , or alternatively, elements are merged if the curvature is below a smaller threshold u_{\min} . The

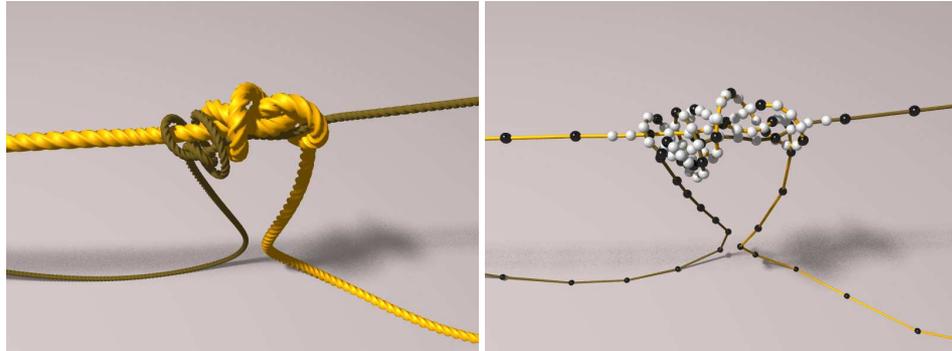


Figure 4.1: In knot-tying simulations, large parts of the elastic rod are undeformed while a high mechanical accuracy is required to simulate the knot, which calls for adaptive methods. The double Fisherman's knot is used to tie together two ropes of unequal radii (left). In the discretization, the black bullets correspond to the initial DOFs, and the white bullets correspond to the adaptively inserted DOFs (right). The collision handling has been realized with the method presented in Chapter 6.

goal of the geometric adaptation is to represent sharp curvatures more smoothly. The *mechanical adaptation criterion* considers the contact configuration of the rod. Then, elements are bisected if additional DOFs are required to reproduce the desired deformation. Both adaptation criteria are mandatory in the simulation of knots.

To subdivide the rod, previous approaches [DDCB01, TWS06] have proposed to insert the new control points at the barycenter of the old control points. While this technique works well for damped material such as clothing, the spatially discontinuously changing curvature will invoke ghost forces that instabilize the simulation. Having stiff differential equations, the elastic rods are particularly sensitive to this problem. To address this problem, a variational subdivision approach is employed. This technique is inspired by the previous work of Kobbelt and Schömer in the field of geometric modeling [KS98]: Instead of inserting a new control point at the barycenter, its energy-minimizing position is found by solving a system of equations. By applying this technique, any post-stabilization schemes based on damping [PLK02] are avoided.

The most important terms in the context of the adaptive simulation have been introduced in Sec. 2.2.3 in the related work chapter. The approach is detailed in Sec. 4.1. In Sec. 4.2, implementation details of the adaptive CORDE model are provided. The approach is evaluated in Sec. 4.3, and a discussion is given in Sec. 4.4. In addition, details on the non-linear conjugate gradient method are found in Appendix D.

4.1 Approach

The simulation of knots requires a large number of DOFs in a small spatial region in order to meet the necessary mechanical accuracy. Still, large parts of the elastic rod stay undeformed and can be represented by employing only a few DOFs. That is why adaptivity is attractive in the simulation of elastic rods.

In this section, the adaptive framework for the simulation of elastic rods is described. First, the imposed conditions on the underlying deformation model are spelled out. Then, the adaptation criteria that govern the (un)refinement of the elements are explained. The main part of this section is dedicated to the variational subdivision procedure.

4.1.1 Deformation model

It is assumed that the deformation model for the rod is hyper-elastic. That means, there exists a scalar-valued strain energy density function $V(\mathbf{g} - \hat{\mathbf{g}})$, where $\mathbf{g}(\sigma, t) : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^N$ is the configuration of the rod at time t defined on a domain $\Omega \subset \mathbb{R}$, and $\hat{\mathbf{g}}$ is the stress free resting configuration of the rod. The static equilibrium configuration is then characterized as a critical point of the functional

$$\int_{\Omega} V(\mathbf{g}(s) - \hat{\mathbf{g}}(s)) ds - \int_{\Omega} \mathbf{g}(s) \mathbf{f} ds \quad (4.1)$$

where \mathbf{f} are generalized external forces. The dynamic equilibrium is obtained by considering the mass matrix and dissipation potentials. V is required to be convex, with a minimum at $V(\mathbf{0})$. Examples for such deformation models include the deformation models introduced by Phillips *et al.* [PLK02], Theetten *et al.* [TGAB07], or the CORDE deformation model that is discussed in Chapter 3.

To numerically simulate the rod, a finite element method usually solves the weak form of the governing differential equation. The domain Ω is discretized into a disjoint union of $N - 1$ elements. The nodes or *control points* $\mathbf{g}_i \in \mathbb{R}^M$ then comprise the DOFs of the elastic rod. For example, for the CORDE deformation model, one has control points $\mathbf{g}_i = (\mathbf{r}_i^T, \mathbf{q}_i^T)^T \in \mathbb{R}^7$, with spatial control points $\mathbf{r}_i \in \mathbb{R}^3$ and quaternions $\mathbf{q}_i \in \mathbb{R}^4$. The control points \mathbf{g}_i are collected in a control point vector $\mathbf{g} = (\mathbf{g}_1^T \cdots \mathbf{g}_M^T)^T \in \mathbb{R}^{NM}$, replacing the previously defined continuous configuration function $\mathbf{g}(s)$. The resulting equations of motion are then

$$\mathbf{M}(\dot{\mathbf{g}}, \mathbf{g}) \ddot{\mathbf{g}} + \mathbf{k}(\dot{\mathbf{g}}, \mathbf{g}) - \mathbf{f} = 0 \quad (4.2)$$

with mass-matrix \mathbf{M} and stiffness function \mathbf{k} .

4.1.2 Adaptation criteria

Most existing adaptive schemes define adaptation criteria (sometimes also denoted as quality criteria or error criteria) solely on geometric properties, i. e. how well the basis functions approximate the current deformation state [DDCB01, CGC*02a].

In multigrid approaches, one usually compares the solutions at different hierarchy levels [GTS02, OGRG07]. However, as Lenoir *et al.* [LGCM05] recognized, a pure geometric criterion is not sufficient for elastic rods. Instead, colliding elements have to be refined in order to meet the required DOFs. While Lenoir *et al.* only consider user interactions with the rod, the approach follows Gayle *et al.* [GLM06] who incorporate the contact configuration. More precisely, an element i is bisected if there is at least one contact on the left and on the right subelement. With this adaptation criterion, the instabilities that are related to cascaded refinements can be avoided. Two elements are merged if there has not been a contact for a user-defined time span t_{merge} .

Further, a geometric adaptation criterion is imposed that considers the local curvature of the rod, i. e. the rod is refined if the curvature $\|\mathbf{u}\|$ between two consecutive elements exceeds a user-defined threshold value u_{max} . Similar to [DDCB01, LGCM05], a smaller threshold value u_{min} is used to unrefine an element. In the experiments, a threshold value of $u_{\text{max}} = 0.6$ is used to refine and a threshold value of $u_{\text{min}} = 0.2$ is used to unrefine elements.

4.1.3 Element refinement

If the adaptation criteria for an element i are met, the element is bisected into two subelements i_1 and i_2 , and a new control point \mathbf{g}_i^+ is introduced. Likewise, the element domain Ω_i is partitioned into disjoint subelement domains Ω_{i_1} and Ω_{i_2} with $\Omega_i = \Omega_{i_1} \cup \Omega_{i_2}$ in order to preserve the resting length of the rod.

Previous approaches propose to place the introduced control point \mathbf{g}_i^+ at the barycenter of the old control points \mathbf{g}_{i-1}^- and \mathbf{g}_i^- . However, since the curvature in the new control point \mathbf{g}_i^+ will be 0, the curvature energies are not distributed uniformly over the elements, which in turn causes instabilities in the underlying differential equations. While previous approaches recommend to add artificial damping springs to stabilize the simulation over time [PLK02], a procedure is proposed that computes valid, i. e. energy-minimizing control points instantaneously.

Since a new control point cannot be inserted without either the stretch or the curvature varying discontinuously, the left or the right control point has to be displaced accordingly. For efficiency purposes, the method always chooses the neighboring control point whose adjacent line segments are involved in less collisions. The goal is to find new control point positions \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ such that the strain energy over the new elements is minimized (see Fig. 4.2). More precisely, the new positions \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ constitute the solution of the non-linear constrained minimization problem

$$\sum_{j=i-1}^{i+1} \int_{\Omega_j} V(\mathbf{g} - \hat{\mathbf{g}}) ds - \sum_{j=i}^{i+1} \mathbf{g}_j^T \mathbf{f}_j \rightarrow \min \quad (4.3)$$

subject to the holonomic boundary conditions

$$\mathbf{g}_{i-1}^+ - \mathbf{g}_{i-1}^- = 0 \quad (4.4)$$

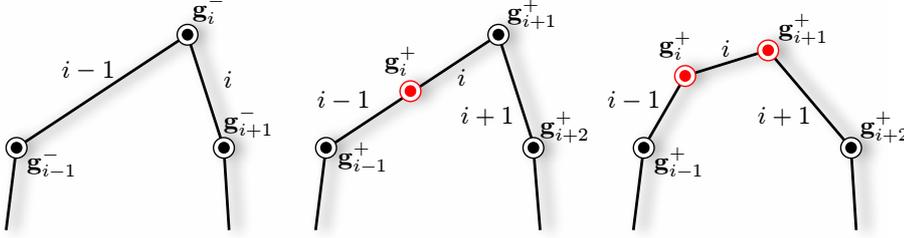


Figure 4.2: By placing the new control point \mathbf{g}_i^+ at the barycenter of \mathbf{g}_{i-1}^- and \mathbf{g}_i^- , the curvature changes discontinuously. In contrast, here, energy-minimizing control points \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ are computed.

$$\mathbf{g}_{i+2}^+ - \mathbf{g}_{i+1}^- = 0 \quad (4.5)$$

where the \mathbf{f}_j are external forces such as gravity or contact forces acting on the control points. The incorporation of contact forces is important since displacing control points eventually result in interferences (see Sec. 6.1 for details). Intuitively, the control points \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ are searched that minimize the strain energy while preserving all other control points in the mesh (notice that \mathbf{g}_{i+1}^- becomes \mathbf{g}_{i+2}^+ after refinement, as indicated in Fig. 4.2). Since the lengths of the new elements sum to the length of the old element, and since all other control points are fixed during refinement, both the current and the resting length of the rod are preserved.

The solution of this system of equations is computed by a non-linear conjugate gradient (CG) method. The necessary analytic differentiation and integration of V is done with a symbolic computer algebra software. Notice that the gradient of V conforms to the restitution force vector. If analytic integration is not possible, the expressions can be obtained by a quadrature method. The step length for the line search in the CG algorithm is determined by employing the Wolfe conditions [NW99], with the initial guess depending on the employed deformation model. Details, including the algorithms, are found in Appendix D. By only updating the coordinates of the control points \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ during minimization, the boundary conditions are enforced. Since V is restricted to be convex, the computed constrained minimum is guaranteed to be global. The necessary number of iterations depends on the deformation model. For moderately damped elastic rods simulated with CORDE, usually 5 to 10 iterations suffice to stabilize the simulation.

To preserve the dynamics of the rod, the mass-matrix \mathbf{M} in (4.2) has to be updated. For approaches that assume mass-lumping, this conforms to recomputing the point-masses, which can be done efficiently. Further, the velocities of the new control points are linearly interpolated from the old control points.

4.1.4 Element unrefinement

To unrefine the rod, two neighboring elements are merged into one. Again, it is not possible to just remove a control point \mathbf{g}_i^- . Instead, either the left or the right control point has to be displaced to the energy-minimizing position. The solution of this non-linear minimization problem is similar to the refinement-case.

4.1.5 Remarks

The scheme in its present form always bisects elements in the middle. It is, however, easy to extend the scheme to general bisection rules. Moreover, since remeshing is trivial for one-dimensional structures, the scheme allows for both structured and unstructured refinement. Depending on the deformation model, it is necessary to enforce upper and lower bounds on the domains Ω_i , i. e. $\Omega_i \in [\Omega^{\min}, \Omega^{\max}]$. A mathematical analysis on these bounds is, however, beyond the scope of this thesis.

The rods are skinned with a spline-interpolated tubical mesh. To avoid visible popping artifacts of the skin during refinement, the old skin is smoothly blended with the new skin. Thus, visual interferences might occur, especially in coarsely sampled regions with high curvature. It has to be underlined that this does not harm the mechanical accuracy since all physical operations are performed on the simulated mesh \mathbf{g} . Still, skinning is part of ongoing work.

4.2 Adaptive CORDE

In Chapter 3, CORDE, a deformation model for elastic rods, has been presented. CORDE is classified as being a discrete hyper-elastic deformation model, and thus it is well-suited for the adaptive framework discussed in this chapter.

As illustrated in Sec. 3.2, the centerline of the rod is discretized into N mass points $\mathbf{r}_i \in \mathbb{R}^3$. Further, the $N - 1$ material frames are represented by quaternions $\mathbf{q}_i \in \mathbb{R}^4$. The penalty method has been employed to adapt the material frames to the centerline. The stretch energy V_s is obtained by integrating the stretch energy $V_s[i]$ per centerline element i , and summing the resulting elemental stretch energies. Likewise, the bend and twist energy V_b is obtained by integrating the bend energy $V_b[i]$ per orientation element i , and summing the resulting elemental bend energies. Further, the constraint energy E_p resulting from the penalty method is obtained by integrating the constraint energy $E_p[i]$ per centerline element i , and again summing the resulting constraint energies. Notice that the $E_p[i]$ are functions in $\mathbf{r}_i, \mathbf{r}_{i+1}$ and \mathbf{q}_i . The total potential energy is then the sum of the stretch energy V_s , the bend energy V_b , and the constraint energy E_s .

To simulate an elastic rod with CORDE, its initial configuration has to be defined. The initial configuration is user-defined and conforms to a set of N spatial coordinates that correspond to the mass points \mathbf{r}_i . These initial mass points constitute the DOFs of the initial rod, and the initial configuration will not be unrefined

further, even if it is completely undeformed. Notice that it is possible to further define intrinsic curvatures $\hat{\mathbf{u}}_i$ for each orientation element.

If either the geometric or the mechanical adaptation criterion evaluates to true for a centerline element $i - 1$, then it is refined. To accomplish this, the element $i - 1$ with endpoints \mathbf{r}_{i-1}^- and \mathbf{r}_i^- is bisected and a new node \mathbf{r}_i^+ is inserted. Further, \mathbf{r}_i^- becomes \mathbf{r}_{i+1}^+ , and \mathbf{r}_{i+1}^- becomes \mathbf{r}_{i+2}^+ , as illustrated in Fig. 4.2. In addition, the quaternion \mathbf{q}_{i-1}^- is replaced with two new quaternions \mathbf{q}_{i-1}^+ and \mathbf{q}_i^+ , and the quaternion \mathbf{q}_i^- becomes \mathbf{q}_{i+1}^+ after the subdivision, thereby considering that the quaternions ‘sit’ on the centerline elements.

When assembling the system of equations governing the minimization problem, one has to consider that by updating the positions \mathbf{r}_i^+ and \mathbf{r}_{i+1}^+ , the orientations of the new centerline elements $i - 1$, i and $i + 1$ change. Therefore, all three quaternions \mathbf{q}_{i-1}^+ , \mathbf{q}_i^+ and \mathbf{q}_{i+1}^+ must be considered in the iterative optimization. The points \mathbf{r}_{i-1}^+ and \mathbf{r}_{i+2}^+ , and the quaternions \mathbf{q}_{i-2}^+ and \mathbf{q}_{i+2}^+ constitute the boundaries that keep the solution local. Consequently, the number of unknowns in (4.3) is 18. Notice that the minimum element length Ω^{\min} depends on both the stiffness and the simulation time step.

4.3 Results

A series of experiments has been staged in order to evaluate the adaptive framework. The method is exemplified by employing the CORDE deformation model as described in Chapter 3. In a first experiment, it is illustrated that the dynamic behavior of the simulation is not affected by the adaptive refinement. The performance gain of the adaptive model is investigated in a second experiment. The experiments have been performed on an Intel Xeon PC, 3.8Ghz.

Dynamic behavior One of the main benefits of the proposed adaptive model is that it determines the exact control point positions instantaneously. It does not assume time-coherence, by damping the control points over time as in previous approaches. Fig. 4.3 illustrates a collision of two angular accelerated rods with a rigid bar. While one rod has a fixed resolution, the other rod is refined to respond more accurately to the collision. As the images illustrate, the dynamic behavior of the refined rod does not significantly differ from the dynamics of the unrefined rod. In average, 10-20 CG iterations are necessary to obtain stable control point positions. The average time to insert or remove a control point is 1.5ms, which includes the time to update the data structures.

Performance To investigate the gain in efficiency when employing an adaptive model, two ropes have been simulated that are tied together with the double Fisherman’s knot (see Fig. 4.4). The behavior of an adaptive rope is compared versus a uniformly sampled high-resolution rope whose sampling density conforms to the



Figure 4.3: Two rods with equal material properties are colliding with a rigid obstacle. One rod is dynamically refined while the other rod has a fixed resolution. The four images illustrate that the dynamic behavior is not affected by the refinement.

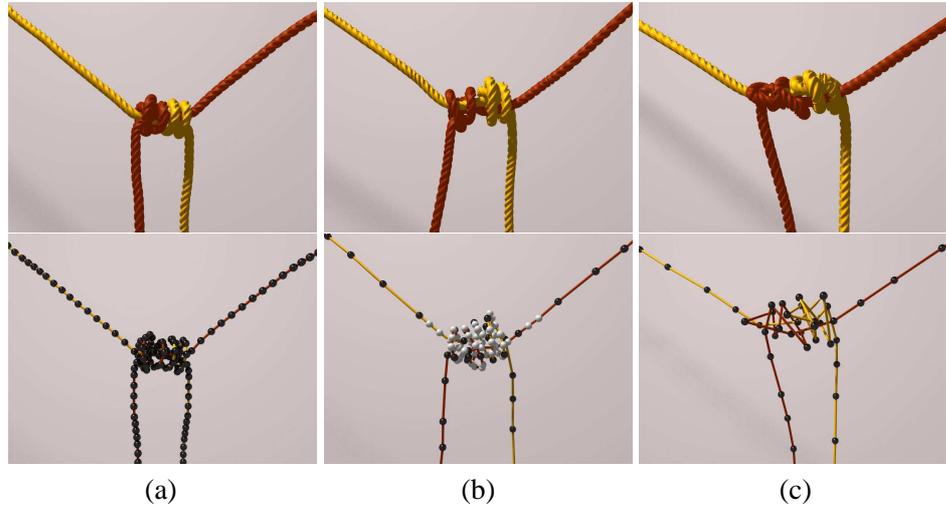


Figure 4.4: Simulation of two ropes tied together with the double Fisherman's knot. (a) A high-resolution rope with 252 nodes. (b) An adaptive rope with 116 nodes and the same maximum resolution as the high-resolution rope. (c) A low-resolution rope with 63 nodes.

maximum resolution of the adaptive rod, and versus a uniformly sampled low-resolution rope whose sampling density conforms to the minimum resolution of the adaptive rope. For the adaptive rope, the allowed minimum element length is $\Omega_i^{\min} = \frac{1}{4}\Omega_i^0$ for all elements i . As a consequence, the maximum number of nodes for the adaptive rope (conforming to the number of nodes for the high-resolution rope) is four times the number of nodes of the low-resolution rope. As Tab. 4.1 indicates, simulating an adaptive rope is almost twice as efficient as the high-resolution rope. The number of collisions for both cases do not vary significantly, indicating that the adaptive rope does not result in a loss of accuracy. On the other hand, simulating the knot with a low-resolution rope results in physical and visual artifacts exhibited in Fig. 4.4. These artifacts are induced by the insufficient control point sampling.

For these moderately damped ropes, 10 iterations are sufficient to stabilize the simulation. If fewer iterations are performed, then the simulation becomes insta-

	High res.	Adaptive	Low res.
Nodes (Fig. 4.4)	252	(avg.) 116	63
Avg. collisions	70	69	22
Simulation step [ms]	13.3	7.3	2.6
Nodes (Fig. 6.4)	2K	(avg.) 1.9K	0.5K
Avg. collisions	10K	9.7K	2.6K
Simulation step [ms]	710	694	77

Table 4.1: Statistics on the adaptive refinements. The upper three rows show the measurements of the simulation of the double Fisherman’s knot (Fig. 4.4). As the measurements indicate, simulating the knot with adaptive ropes is almost twice as efficient and requires less than half the number of nodes than uniformly sampled ropes at the same maximum resolution. Still, the number of collisions does not vary significantly, illustrating that the knot is simulated with similar accuracy. In contrast, the low-resolution ropes cannot reproduce the knot configuration accurately enough. The lower three rows show the measurements of a worst-case scenario for adaptive ropes (Fig. 6.4), illustrating that the overhead for the adaptivity is negligible.

ble. The average time to (un-)refine the adaptive rope is 1.35ms. Still, since the adaptation criteria are only met in about 5% of all simulation steps, only 0.41% of the total simulation time is spent for node insertions and deletions.

Employing an adaptive model always pays off in interactive knot-tying simulation when the location of the knot is not known in advance. Currently, the limiting factor for the refinement is the enforced minimum element length Ω^{\min} , which depends on the deformation model and on the simulation parameters. A worst-case scenario for an adaptive rope is illustrated in Fig. 6.4 in Sec. 6.2, where most elements of the rope are involved in collisions. Thus the number of nodes of the adaptive rope conforms to the number of nodes of the high-resolution rope. Still, the simulation step timings for the adaptive and the high-resolution rope are about the same, illustrating that the overhead of the adaptivity is negligible.

4.4 Discussion

In this section, an adaptive framework for elastic rods has been presented. Motivated by the observation that a high spatial resolution is necessary to represent knots while a few DOFs are often sufficient to represent the undeformed parts of the rod, two adaptation criteria are imposed: First, DOFs are inserted if the curvature between two consecutive segments is above a certain threshold. This criterion is a geometric adaptation criterion. Second, DOFs are inserted if two rods are in contact. In doing so, it is accounted for the fact that rigid segments are employed to model a continuously deforming rod: In order to reproduce a local deformation

induced by a collision, the rigid segment has to be bisected and a new DOF has to be inserted. This adaptation criterion is called a mechanical adaptation criterion. Both criteria are necessary to arrange the DOFs in order to simulate knots.

If the adaptation criteria induce a refinement, then the rod segments are bisected and new control points are inserted. Previous approaches have usually inserted the new control points at the barycenter of the old control points. While this works well for two- and three-dimensional objects, it exhibits a problem in the dynamic simulation of one-dimensional objects. Inserting a control point at the barycenter makes that the curvature change discontinuously. In turn, this induces ghost forces that try to balance the curvature. If the refinement operations are cascaded, then these ghost forces can quickly cause the simulation to crash.

Interestingly, this problem has rarely been addressed in the past. Spline-based methods such as [LGCM05] do not exhibit this problem since the determination of the control points is easy for B-spline curves. Phillips *et al.* have proposed to employ damping springs to stabilize the simulation [PLK02], but this affects the dynamics of the rod. In contrast, the solution that has been presented in this chapter determines those control point positions that minimize the deformation energy over the rod segments. This technique is termed “variational subdivision” and is well-investigated in the field of geometric modeling [KS98]. The computations of those positions requires the solution of a constrained non-linear system of equations, which can be done efficiently with a non-linear conjugate gradient method. By employing this technique, any post-stabilization schemes such as damping are avoided, and the simulation is stabilized.

Future work While the method works well in practice, there exists little theory about the minimum element lengths Ω^{\min} that guarantee a stable simulation. Investigating into this topic would provide further insights into the method.

One of the main drawbacks, however, is the skinning of the rod. While the skinning technique works well for static discretizations, it exhibits visual popping artifacts if the underlying resolution changes. This comes from the fact that the skin conforms to a tubical B-spline volume, with the centerline control points of the rod being the control points of the spline. Thus, if the resolution changes, so does the spline.

Chapter 5

Contact handling for volumetric deformable objects

In the previous chapters, the deformation modeling for elastic rods has been discussed. The temporal evolution of the rod has considered to be unconstrained. However, this is not true if either self-interactions or multiple interacting rods are considered. In this case, non-penetration constraints are imposed that prevent parts of the rods passing through each other. Based on the non-penetration constraints, contact forces are computed that maintain these constraints throughout the dynamic evolution. In this thesis, it is looked for a simple, elegant and physically plausible way to enforce the non-penetration constraints. This is accomplished by employing a predictor-corrector methodology, together with a position-based optimization scheme.

In this chapter, a novel contact model for interacting volumetric deformable objects is presented. This comes with an immersed discussion on the general contact mechanics. The goal is to get acquainted with the terms and concepts of the predictor-corrector methodology in contact handling. Equipped with this knowledge, an understanding of the contact handling for elastic rods as discussed in Chapter 6 is made easier.

The contact handling can be regarded as a special case of the constrained simulation. Consequently, when addressing the contact handling, one can stick to the wealth of literature in constrained simulation. What makes contact handling a particular challenging field in the constrained simulation is that the non-penetration constraints are formulated as inequalities. This contrasts the articulated body simulation where the joint constraints are in general formulated as equalities. To illustrate this point, a prismatic joint can be considered. At time t , the configuration of the joint satisfies the constraint, and the relative velocity will be 0. External forces then result in an acceleration that eventually violates the constraint. By employing the method of Lagrange multipliers, constraint forces can be computed that cancel the acceleration contributions that violate the constraint, thereby maintaining a valid configuration in the subsequent time step. Still, the situation becomes more

challenging if inequality constraints are considered. If a collision is reported at time t , then neither the position will satisfy the constraint, nor the relative velocity will be 0. Consequently, it is not possible to compute a constraint force that makes the configuration satisfying the non-penetration constraint in the subsequent time step. This is because constraint forces influence the position only indirectly.

Instead, the exact point in time of the contact has to be backed up. Then, for collisions with a relative velocity larger than 0, a contact impulse causes a discontinuous change in the velocities of the incorporated objects, thereby avoiding an interpenetration in the subsequent time step. For collisions with a relative velocity of 0 (i. e. resting contacts), a contact force cancels the relative acceleration and maintains the collision-free configuration. The decision whether to use contact forces or impulses is NP-hard [Bar91].

In contrast to the rigid body simulation, collisions between deformable objects take a non-zero amount of time during which the incorporated objects deform. Following Baraff and Witkin, a contact impulse has to be computed in the first instant of the collision that discontinuously sets the relative velocity to 0 and thereby prevents a subsequent interpenetration. During the collision process, contact forces maintain the collision-free configuration and induce a deformation of the object surfaces. In turn, the internal elastic forces then induce the release of the objects at a later point in time [BW92]. The magnitude of the contact forces is governed by the momenta and the elasticities of the interacting objects.

The analytical computation of the contact impulses and contact forces in the spirit of Baraff and Witkin has turned out to be a challenging problem. Previous analytical approaches mostly employ Signorini's problem which considers the static equilibrium between deformable objects. By assuming a linear elasticity model, the resulting contact forces can be computed by employing an LCP [PPG04, DDKA06]. However, these techniques are complex from an algorithmic point of view and therefore difficult to implement.

Consequently, most approaches employ the penalty technique that approximates the constraint forces by computing a penalty force whose magnitude depends on an interpenetration measure [MW88, HFS03, HTK*04]. This technique is particularly popular in the context of interacting deformable objects since interpenetrations are usually not visible as the surfaces deform during the collision and thereby 'hiding' these overlaps. Still, they exist, and may produce undesired results if the impact becomes larger. The main drawback, however, is that the plausibility of the technique depends largely on the choice of the spring constant k . If k is large, then the discrete time stepping is prone to overshooting. In contrast, if k is small, then the interpenetrations might be arbitrary large. Consequently, the choice of the spring constant k and the related time step h depends on the simulation context.

In this chapter, an alternative technique is proposed that overcomes the problems that are inherent to penalty-based techniques. Inspired by the rigid body simulation, a predictor-corrector methodology is employed. By predicting the configuration and velocities in the subsequent time step, contact forces can be computed such that the imposed non-penetration constraints are met in the fu-

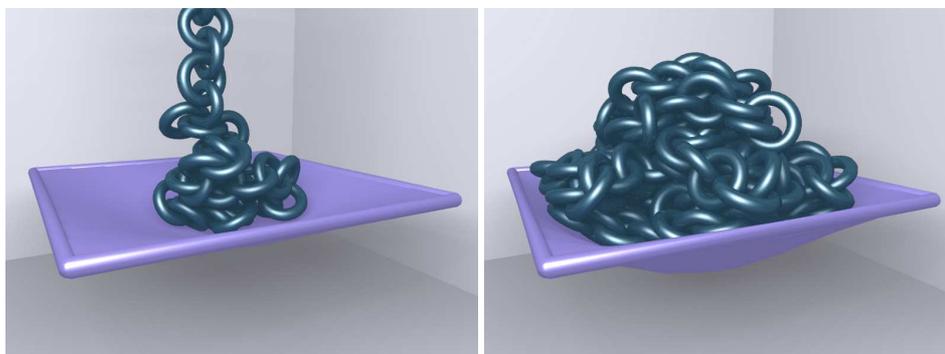


Figure 5.1: Simulation of 200 chained torii falling onto an elastic membrane. The membrane deforms and buckles under the weight of the chain. Nevertheless, a stable resting state is reached using the proposed collision response scheme.

ture [GBF03, WTF06]. Since an explicit integration method is chosen, these contact forces can be non-linear. Thereby, post-stabilization techniques are avoided, and the contact simulation proves to be stable, even for large time steps. Thus, both dynamic collisions and challenging resting states can be efficiently simulated in a physically plausible way, as illustrated in Fig. 5.1. A brief introduction into rigid and deformable contact mechanics is given in Sec. 5.1. The approach is presented in Sec. 5.2 and evaluated in Sec. 5.3. A discussion is given in Sec. 5.4.

5.1 Contact mechanics

In this section, the fundamental concepts of contact mechanics are briefly explained. In the first part, the general principles that apply for all kinds of objects are addressed. Later, the focus is on deformable object contact mechanics.

5.1.1 Principles

As already mentioned in the introduction of this chapter, contact mechanics is related to constrained simulation. If two objects interfere, then a contact force counteracts the interpenetration and maintains the non-penetration constraint. According to the principle of least work [Gol81], the constraint forces are always perpendicular to the constraint manifold. Consequently, the contact forces live in the normal space of the surface of the interacting objects. In this context, the normal space \mathbf{n} is also denoted as the *contact space* [DDKA06]. If the contact force is considered in the contact space, it reduces to a scalar value f . In contrast to contact forces, the frictional forces live in the *tangent space*. First, only frictionless contacts are considered. Afterwards, the discussion is extended to contacts with friction.

Consider two objects that are in contact, but not colliding, i. e. their distance

is 0. If the relative velocity is < 0 , then the two objects will collide subsequently, thereby violating the constraint. Consequently, the velocity of the objects must be changed immediately (i. e. in a discontinuous manner), which is achieved by applying a contact impulse having the units of mass times velocity. If, in contrast, the relative velocity is 0 but the relative acceleration < 0 , then a contact force counteracts the acceleration. The contact force has the units of mass times acceleration. In the following discussion, the term “contact force” refers to impulsive as well as to continuous contact forces.

In rigid object collisions, the objects always touch in a finite set of *contact points* i . The computation of the contact forces results in scalars f_i . Then, contact forces $\mathbf{f}_i = f_i \mathbf{n}_i$ act on these contact points. According to Newton’s third law, if a contact force \mathbf{f}_i acts on the first object, then a contact force $-\mathbf{f}_i$ acts on the second object. Valid contact forces are characterized by three necessary conditions:

1. Contact forces must maintain the collision-free constraint and follow the principles of least action.
2. Contact forces can ‘push’, but they cannot ‘pull’.
3. Contact forces occur only if the objects are in contact, as soon as they separate, the contact forces vanish.

Condition (1) basically states that contact forces are considered as constrained forces. Condition (2) restricts the contact forces to the positive half-space, they are always pointing inwards. Condition (3) gives rise to the the statement that either the contact force or the separating distance must be 0. This condition makes the computation of the contact forces a linear complementary problem. Details are found in [Bar89].

5.1.2 Frictional contacts

The consideration of friction greatly complicates the problem. The reason is two-fold. First, in contrast to the contact force, the friction force lives in the tangent space. Thus, for concave surfaces, the application of a friction force can immediately result in secondary collisions that have to be handled accordingly. Second, there are two kinds of friction, notably static and dynamic friction. Especially the transitions in between are hard to capture analytically.

Friction is commonly modeled with the Coulomb friction model,

$$\mathbf{f}^f = -\mu f \frac{\mathbf{v}^t}{\|\mathbf{v}^t\|} \quad : \quad \mathbf{v}^t \neq \mathbf{0} \quad (5.1)$$

$$\|\mathbf{f}^f\| \leq \mu f \quad : \quad \mathbf{v}^t = \mathbf{0} \quad (5.2)$$

where \mathbf{f}^f is the friction force, \mathbf{v}^t is the tangential component of the relative velocity, f is the normal force on the contact point i (corresponding to the contact force),

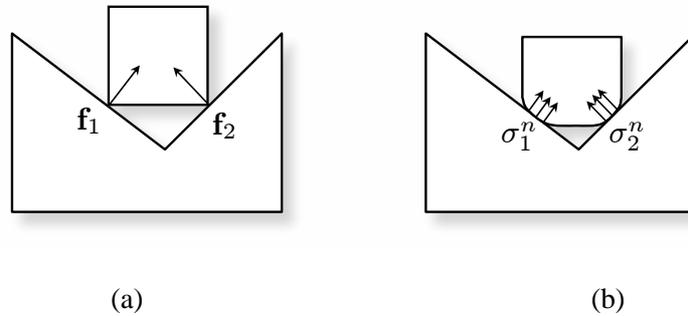


Figure 5.2: (a) In rigid object collisions, the interacting objects touch in contact points. (b) In deformable object collisions, the interacting objects locally deform. The spots where the objects touch are denoted as contact surfaces. The resulting contact forces have the units of pressure.

and μ is the friction coefficient. From (5.1), it follows that the net contact force in a contact point is

$$\mathbf{f} = f(\mathbf{n} - \mu \frac{\mathbf{v}^t}{\|\mathbf{v}^t\|}) \quad (5.3)$$

for dynamic friction [Bar91].

Since the dynamic friction is non-linear, the friction cannot be expressed directly in an LCP form. Therefore, analytical approaches usually approximate the Coulomb friction cone with a k -sided pyramid [DDKA06]. However, it has become widely accepted that secondary collisions due to the tangential friction force are neglected. In turn, this allows to decouple the contact force computation from the friction force computation. The resulting force components are then simply added [PPG04]. In the approach being presented, this view is shared.

5.1.3 Deformable contact mechanics

In contrast to rigid object collisions, a collision between deformable objects always takes a non-zero amount of time during which the objects compress and release. Since the surfaces of the objects deform, the contact forces act on contact surfaces, which correspond to the contact points in the case of rigid object collisions (see Fig. 5.2). The relative velocity of the contact surfaces is necessarily 0 in the course of the collision, which implies that the contact surfaces do not interfere. In contrast, the relative velocity of the centers of mass of the objects is only 0 in the case of resting contacts. If two deformable objects collide, then a contact impulse sets the relative velocity of the contact surface to 0. During the subsequent collision course, contact forces prevent a further interpenetration. In the following discussion, it is assumed that the relative velocity is 0, and the focus is on the computation of the

contact forces. First, the case of the static equilibrium between deformable objects is addressed. Later, the discussion is extended to the dynamic case.

The static case is characterized in the contact force \mathbf{f} on the contact surface (therefore having the units of pressure) being equal to the normal stress σ^n :

$$\mathbf{f} = \sigma^n := \mathbf{n}(\boldsymbol{\sigma} \cdot \mathbf{n}) \quad (5.4)$$

In the static case, the velocity of the contact surface is 0 in the local frame of the colliding objects. If condition (3) is combined with Eq. 5.4, then the complementary relation

$$0 \leq \mathbf{d} \perp \sigma^n \geq 0 \quad (5.5)$$

is obtained. This relation is known as Signorini's formulation [DAK04] and states that either the separating distance \mathbf{d} between the objects is 0, and there may be a stress on the deformed surfaces, or alternatively the separating distance is > 0 , which implies that the stresses are 0 since the objects are undeformed. Again, this formulation does not consider dynamic effects such as vibrations. The relation (5.5) can be combined with a FE method in order to compute the stresses, which results in an LCP formulation for the contact forces. Such static formulations are employed by e. g. [DDKA06] or [PPG04].

The formulation of the contact forces based on Signorini's problem is particularly well-suited for approaches which employ an implicit time-integration method such as the implicit Euler method. The reason is that if a linear elasticity model is assumed, the contact forces become linear. Consequently, the system of equations that is imposed by the implicit integration method becomes linear, which in turn enables the efficient solve. Details are found in [DDKA06].

If the dynamic case is considered, then the velocity of the contact surface is in general not 0. Instead, it will move during the collision course, depending on the relative elasticity of the two objects. To illustrate this point, consider a deformable object colliding with a rigid obstacle. In the local frame of the deformable object, the contact surface will accelerate towards the center of mass of the deformable object until the point of maximum compression is reached. Afterwards, the contact surface will depart again, until the two objects separate. Since a formulation of the stresses in the spirit of the Signorini's formulation is not at hand, an LCP formulation for the dynamic case is hard to formulate. In turn, the simulation of dynamic collisions, along with an implicit time-integrator, is hard to realize.

However, if an explicit time-integration method is employed, then the contact forces may be non-linear. In this case, the computation of the contact forces is governed by the condition that *the contact forces must sum to 0*. This follows directly from Newton's third law, and it will govern the computation of the contact forces in the approach being presented in this chapter. The condition conforms to the *conservation of momentum* during the collision.

5.2 Approach

In this section, an approach is presented that computes contact forces for interacting deformable objects. The approach follows the predictor-corrector methodology. In contrast to previous penalty-based approaches, interpenetrations are avoided. In contrast to previous analytical methods, the contact force computation is decoupled and an explicit time-integration method is employed. Therefore, the simulation of dynamic collisions is enabled.

5.2.1 Object representation

In the following discussion, it is assumed that the deformable objects are discretized into tetrahedral meshes. To produce these meshes, a novel mesh generator has been employed. The mesh generator is detailed in Appendix A. As a consequence, the boundaries of the objects are closed triangulated manifolds. Further, it is assumed that the mass of the object is lumped in the nodes of the mesh that are here consistently denoted as mass points. Mass lumping makes that the mass of an element is equally distributed within the elements, resulting in a discontinuous mass ‘jump’ at the element boundaries [WDGT01]. The underlying deformation model is treated as a black box. That is, the approach does not make any assumptions on the way the internal forces acting on the mass points are computed. In the experiments, both a mass-spring deformation model [THMG04] and a co-rotational FE method [MG04] have been employed.

5.2.2 Predictor-corrector simulation

The approach follows a variant of the predictor-corrector methodology. First, the deformation model computes the elastic forces $\sigma(t)$, based on the deformed configuration $\mathbf{x}(t)$ at time t . Then the objects are numerically evolved in time, resulting in unconstrained future positions $\tilde{\mathbf{x}}(t+h)$. These unconstrained positions might be colliding, therefore a collision detection is carried out. The collision detection results in the set C of colliding points i . For each point i , the contact space $\mathbf{d}_i(t+h)$ is computed. At this point, the approach being presented steps in and computes contact forces $\mathbf{f}(t)$ that serve as additional external forces. Then the configuration $\mathbf{x}(t)$ is restored and the numerical re-integration of the points $\mathbf{x}(t)$ results in positions $\mathbf{x}(t+h)$ that satisfy the constraints. Algorithm 1 sketches the simulation loop.

The function `ElasticForces` depends on the deformation model. The resulting elastic force vector σ can result from any black box process, such as a FE method, a modal analysis, or a mass-spring method. The function `ExternalForces` returns the external force vector \mathbf{f}^e by adding gravity and user-interaction forces. Likewise, the functions `UnconstrainedMotion` and `ConstrainedMotion` that both numerically integrate the equations of motion are considered as black boxes, as long as they perform an explicit integration. In case of a multistep integration scheme such as Runge-Kutta, the simulation loop has to be adapted accordingly. The experi-

SimulationLoop**repeat** $\boldsymbol{\sigma}(t) \leftarrow \text{ElasticForces}(\mathbf{x}(t));$ $\mathbf{f}^e(t) \leftarrow \text{ExternalForces};$ $\tilde{\mathbf{x}}(t+h) \leftarrow \text{UnconstrainedMotion}(\mathbf{x}(t), \mathbf{v}(t), \boldsymbol{\sigma}(t));$ $C \leftarrow \text{DetectCollisions}(\tilde{\mathbf{x}}(t+h));$ $\mathbf{d}(t+h) \leftarrow \text{ComputeContactSpace}(C, \tilde{\mathbf{x}}(t+h));$ $\mathbf{f}(t) \leftarrow \text{ComputeContactForces}(C, \mathbf{x}(t), \tilde{\mathbf{x}}(t+h), \mathbf{d}(t+h), \boldsymbol{\sigma}(t), \mathbf{f}^e(t));$ $\mathbf{x}(t+h) \leftarrow \text{ConstrainedMotion}(C, \mathbf{x}(t), \mathbf{v}(t), \boldsymbol{\sigma}(t) + \mathbf{f}^e(t) + \mathbf{f}(t));$ $t \leftarrow t+h;$ **until** stop;

Algorithm 1: The main loop of a predictor-corrector based simulation approach first computes the elastic forces and then temporally evolves the configuration. The contact handling is then performed on the unconstrained configuration, and the re-integration results in the constrained configuration for the next time step.

ments have been carried out with the Verlet integration scheme. Of note is that the same integration scheme is applied in both the unconstrained and the constrained integration. Only those mass points that are subject to a constrained motion are then re-integrated, where the constraint forces basically serve as additional force components. To detect the collisions, a spatial hashing approach [THM*03] is employed. Further, the contact space is computed with the approach described in [HTK*04]. This approach guarantees that the computed penetration directions are consistent, even for deep penetrations that occur in violent collisions.

The computation of the contact forces is done in the function `ComputeContactForces` that is focused in the subsequent sections. The proceeding is as follows: First, the necessary terms are introduced. Then, the analytical expressions for the contact forces are provided and motivated, which also includes a mathematical justification.

5.2.3 Terms

A *collision* is a pair (i, T_i) , where $i \in C$ is a point that has interpenetrated a volume under unconstrained motion. $T_i = \{j, k, l\}$ is the triangle on the surface of the collided discretized object that has been penetrated by i . i is consistently denoted as *contact point*, and T_i as the corresponding *contact triangle*.

From the assumption of triangulated surfaces, it follows that a point of one object always penetrates a surface triangle of the other object. Consequently, for each collided point, a corresponding contact triangle can be identified. For degenerated contact points [Bar89], a random neighboring triangle is chosen. Moreover, edge collisions are neglected.

In order to compute a smooth answer to a collision, the coordinates of a collided

point with respect to its contact triangle must be determined. These coordinates are captured in the *mapping matrix* $\mathbf{H} = (h_{ij}) \in \mathbb{R}^{N \times N}$ that provides the barycentric coordinates $\omega_{i,j}$ of the point i projected onto T_i with respect to $j \in T_i$ (see Fig. 5.3):

$$h_{ij} = \begin{cases} \omega_{i,j} & j \in T_i \\ 0 & \text{else} \end{cases} \quad (5.6)$$

Since a contact point is colliding with exactly one contact triangle, each row in \mathbf{H} contains exactly three non-zero entries. And since the barycentric coordinates partition unity, the sum of each row is exactly one, $\sum_{j=1}^n h_{ij} = 1$ for all i . The size N of \mathbf{H} conforms to the sum of the nodes of the two discretized objects whose collision is considered. However, notice that the concept of the mapping matrix \mathbf{H} allows for a consistent derivation of the contact forces, but it is not explicitly represented in the actual implementation. Instead, the barycentric coordinates are stored per contact point, thereby saving memory. The h_{ij} can easily be obtained from the penetration directions \mathbf{d}_i .

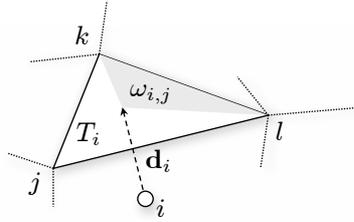


Figure 5.3: The element h_{ij} provides the barycentric coordinate $\omega_{i,j}$ of the contact point i projected onto the penetrated surface triangle T_i with respect to $j \in T_i$.

An important quantity is the sum $\sum_{j=1}^N h_{ji}$ of the mapping matrix entries per column i . This sum considers the barycentric coordinates of all points that have been collided with a surface triangle in the neighborhood of a node j . This sum is in general not 1. However, since the row sums are all 1, it holds that the average of the column sums is as well 1. Later, this property will turn out to be important.

5.2.4 Overview

The approach computes contact forces that correct the unconstrained motion, which results in a collision-free configuration in the subsequent time step. To accomplish this, the *collisions* (the pairs of contact points and contact triangles) are first considered in isolation. For each collision, displacements are computed such that the resulting point/triangle configuration is collision-free, under the condition that the displacements conserve the momentum. Since the collisions are spatially coupled (the nodes constituting the contact triangle might themselves be contact points), the displacements are summed up per contact point, thereby arriving at the final displacements. The contact forces are then chosen such that after the numerical time-integration, the computed goal positions are met exactly.

The layout of the explanation is as follows: First, the resulting expression for the contact forces is given and intuitively motivated in Sec. 5.2.5. Staffed with this knowledge, a implementation of the collision response is already enabled. To justify the contact forces mathematically, the expressions are proofed by back-substitution in Sec. 5.2.6. Frictional contacts are briefly explained in Sec. 5.2.7. The explanation concludes by proposing a correction scheme that inhibits ghost forces, as detailed in Sec. 5.2.8.

5.2.5 Contact forces

The direct computation of the contact forces is enabled by the fundamental observation that in order to conserve the momentum during the collision, the contact forces must sum to 0. First, it is shown that the contact forces correct the unconstrained motion instantaneously. Second, a mathematical justification shows that both the the momentum is conserved and the non-interference condition is satisfied up to a discretization error.

Claim (Time-integration) By applying a contact force

$$\mathbf{f}_i(t) = \frac{m_i}{h^2} \mathbf{d}_i(t+h) \alpha_i \quad (5.7)$$

to the mass points at time t , the unconstrained motion is corrected and the resulting positions $\mathbf{x}_i(t+h)$ constitute a collision-free configuration. Further, the contact forces sum to 0 and thus conserve the momentum.

In the formula for the contact forces (5.7), m_i is the mass of the point i , and t is the time step of the numerical integration. \mathbf{d}_i is the contact space of i . Further, α_i is a dimensionless scalar that ensures that the resulting goal position of i does not interfere with the corresponding contact triangle T_i . Since m_i has the units of weight, t and h have the units of time, \mathbf{d}_i has the units of distance, and α_i is dimensionless, the resulting force \mathbf{f}_i has the units of $[\text{kg m s}^{-2}]$, which corresponds to Newton, the units of force.

Proof (Time-integration) To understand the contact force expression (5.7), it is worth to carry out the numerical time-integration of the force. Consider the (collision-free) position $\mathbf{x}_i(t)$ of a contact point i at time t . If the sum of the internal force $\boldsymbol{\sigma}_i(t)$, the external force $\mathbf{f}_i^e(t)$ and the contact force $\mathbf{f}_i(t)$ is time-integrated with the Euler scheme, then

$$\mathbf{x}_i(t+h) = \mathbf{x}_i(t) + (\mathbf{v}_i(t) + \frac{1}{m_i} (\boldsymbol{\sigma}_i(t) + \mathbf{f}_i^e(t) + \mathbf{f}_i(t)) h) h \quad (5.8)$$

$$= \underbrace{\mathbf{x}_i(t) + (\mathbf{v}_i(t) + \frac{1}{m_i} (\boldsymbol{\sigma}_i(t) + \mathbf{f}_i^e(t)) h) h}_{=\tilde{\mathbf{x}}_i(t+h)} + \frac{1}{m_i} \mathbf{f}_i(t) h^2 \quad (5.9)$$

$$= \tilde{\mathbf{x}}_i(t+h) + \frac{h^2}{m_i} \mathbf{f}_i(t) \quad (5.10)$$

$$= \tilde{\mathbf{x}}_i(t+h) + \mathbf{d}_i(t+h)\alpha_i \quad (5.11)$$

is obtained. Interestingly, by time-integrate (5.7), the unconstrained position $\tilde{\mathbf{x}}_i(t+h)$ is corrected, and the subsequent constrained position $\mathbf{x}_i(t+h)$ is exactly $\tilde{\mathbf{x}}_i(t+h) + \mathbf{d}_i(t+h)\alpha_i$. Therefore, if the *displacements* $\mathbf{d}_i(t+h)\alpha_i$ are chosen such that the positions $\tilde{\mathbf{x}}_i(t+h) + \mathbf{d}_i(t+h)\alpha_i$ constitute a collision-free configuration, then the corresponding contact forces $\mathbf{f}_i(t) = \frac{m_i}{h^2} \mathbf{d}_i(t+h)\alpha_i$ are guaranteed to result in the collision-free positions $\mathbf{x}_i(t+h)$ after time-integration.

□

Similar techniques have been employed by Choe *et al.* in the context of hair simulation [CCK05] and Gissler *et al.* for maintaining geometric constraints in deformable object modeling [GBT06]. In this thesis, this technique is also employed in the computation of contact forces for elastic rods, as discussed in Sec. 6.1.

Collision-free displacements The collision-free positions $\mathbf{x}_i(t+h)$ live in the contact spaces \mathbf{d}_i of the contact points. Since the contact spaces have dimension one, the problem reduces to finding the feasible scalars α_i .

The α_i depend on the relative masses of the collisions and therefore govern the resulting position of the points and the corresponding contact triangles. They are obtained as

$$\alpha_i = \frac{\sum_{j=1}^n c_j h_{ij} m_j}{c_i m_i + \sum_{j=1}^n c_j h_{ij} m_j} \quad (5.12)$$

where the c_j are normalization factors

$$c_i = \frac{1}{1 + \sum_{j=1}^n h_{ji}} \quad (5.13)$$

To understand this formula, one has to reconsider the two conditions that the contact forces must satisfy, notably (a) that the resulting constrained positions after the numerical time-integration are collision-free, and (b) that the contact forces sum to 0. The second condition corresponds to the conservation of momentum in the collision.

The first thing to understand is that the row sum $\sum_{j=1}^n c_j h_{ij} m_j$ in (5.12) adds the weighted masses of the contact triangle of i , because $h_{ij} = 0$ for all j that do not constitute the contact triangle of i . Consequently, the scalar α_i is understood as a mass ratio between the contact point i and its corresponding contact triangle T_i .

However, by computing the mass ratio between a contact point and its contact triangle, it has to be considered that the nodes constituting the contact triangle might serve themselves as contact points. This results in a coupling of the contacts, as already earlier mentioned by e. g. Volino *et al.* [VMT00]. This fact has to be

considered in the computation of the mass ratios that finally governs the conservation of momentum. And therein lies the motivation of the normalization factors c_j : They account for the fact that the mass contribution of a single contact point is considered several times in the computation of the contact forces, notably exactly one time as contact point, and k times as node constituting a contact triangle. Here, k is the number of contact points having a corresponding contact triangle in the neighborhood of the considered contact point. The column sum $\sum_{j=1}^n h_{ji}$ computes k , i. e. how many times the point i is considered as a corner of a contact triangle. By weighting the barycentric coordinates with c_j , it is guaranteed that each mass is considered exactly once. An amortized analysis shows that the weights c_i can be computed per contact point at constant costs. Thus the costs to compute \mathbf{f}_i per point are constant.

While this justification is kept informal in order to support the intuitive understanding, a mathematical justification is given in the subsequent section.

5.2.6 Mathematical justification

In this section, the correctness of the proposed contact force expression (5.7) is proved. The focus is on showing that the positions $\mathbf{x}_i(t+h) = \tilde{\mathbf{x}}_i(t+h) + \mathbf{d}_i\alpha_i$ constitute a collision-free configuration, and that the resulting displacements make that the contact forces sum to 0. By plugging the displacements into (5.7), it has been shown in (5.8) that the collision-free positions are immediately met after the numerical time-integration.

Claim (Non-penetration) The positions $\mathbf{x}_i(t+h) = \tilde{\mathbf{x}}_i(t+h) + \mathbf{d}_i\alpha_i$ constitute a collision-free configuration (up to a small discretization error).

Proof (Non-penetration) In order to accomplish the proof, a simplifying assumption has to be taken, notably that *the contact space of a collision has dimension one*. Mathematically,

$$\mathbf{d}_j = \mathbf{d}_k = \mathbf{d}_l = -\mathbf{d}_i \quad (5.14)$$

where i is the contact point, and j , k and l constitute the corresponding contact triangle. This assumption holds only for perfectly flat contact surfaces. Otherwise, an error that depends on the curvature of the contact surface is introduced. Still, the resulting overlaps are acceptable.

By assuming that the contact space per collision has dimension one, the non-interference property of a collision implies that $\alpha_i + \alpha_j = 1$ for $j \in T_i$. Now consider the sum $\sum_{j=1}^n c_j h_{ij} m_j$ in the numerator of the expression (5.12) for α_i . This sum is interpreted as a mass averaging operation over vertices j on the contact triangle. Thus, the average mass \bar{m}_j of vertices j is defined as

$$\sum_{j=1}^n c_j h_{ij} m_j = \bar{m}_j \sum_{j=1}^n h_{ij} c_j \quad (5.15)$$

The row sums $\sum_{j=1}^n h_{ij}$ of the mapping matrix \mathbf{H} equal 1 per definition. Thus, the average of the column sums $\sum_{j=1}^n h_{ji}$ of \mathbf{H} is also 1, and therefore allows to approximate $c_j \approx \frac{1}{2}$. It has to be noted that the approximation $c_j \approx \frac{1}{2}$ is only acceptable in the context of the proof for non-penetration, because here, small overlaps are tolerated. However, in the subsequent proof for momentum conservation, the exact expressions for c_j have to be considered. The approximation of c_j allows to rewrite α_i of the contact point as

$$\alpha_i = \frac{\bar{m}_j/2}{m_i/2 + \bar{m}_j/2} \quad (5.16)$$

Now consider a point $j \in T_i$. The α_j of this point is

$$\alpha_j = \frac{\bar{m}_k/2}{m_j/2 + \bar{m}_k/2} \quad (5.17)$$

with \bar{m}_k the average mass of points in j 's corresponding contact triangle T_j . Thus, the nodes constituting T_j are in the neighborhood of i . By assuming that the differences in the masses of neighboring points are small, \bar{m}_k and \bar{m}_j can be approximated as $\bar{m}_k \approx m_i$ and $\bar{m}_j \approx m_j$. Thus the sum of α_i and α_j is

$$\alpha_i + \alpha_j \approx \frac{m_j/2}{m_i/2 + m_j/2} + \frac{m_i/2}{m_j/2 + m_i/2} = 1 \quad (5.18)$$

□

Notice that this does not restrict the masses of the colliding objects to be equal. In fact, the expression for α_i ensures that collisions of objects at differing weights are handled correctly, which is in contrast to penalty methods. Further, the objects may have spatially differing masses, as long as the mass field variations are kept within reasonable limits.

Claim (Global conservation of momentum) The contact forces \mathbf{f}_i sum to 0.

To illustrate that the proposed contact forces \mathbf{f}_i sum to 0 and thus ensure the conservation of momentum, the concept of *local contact forces* has to be introduced. The local contact forces are the forces that arise due to a single collision considered in isolation. That is, for a collision, one local contact force acts on the contact point, and three local contact forces on the nodes of the corresponding contact triangle. The contact force on a contact point i is then obtained by summing the local contact forces from all collisions where i is involved into.

The proceeding is as follows: First, it is shown that the local contact forces per collision sum to 0, which corresponds to the conservation of momentum for a single collision. Then, the sum of all local contact forces per contact point is formed, which turns out to be the proposed contact force (5.7). This result enables the conclusion that the contact forces sum to 0. Thus, before proving the claimed global conservation of momentum, the local conservation of momentum is proofed.

Claim (Local conservation of momentum) Having a collision (i, T_i) , the local force

$$\tilde{\mathbf{f}}_i = c_i \frac{m_i}{h^2} \mathbf{d}_i \alpha_i \quad (5.19)$$

on the point i and the local force

$$\tilde{\mathbf{f}}_{T_i} = \sum_{j=1}^n c_j h_{ij} \frac{m_j}{h^2} (-\mathbf{d}_i) (1 - \alpha_i). \quad (5.20)$$

on the triangle T_i ensure a local force equilibrium. Further, the sum of the local forces per point yields the proposed contact force \mathbf{f}_i on that point.

Proof (Local conservation of momentum) For the local force equilibrium,

$$\begin{aligned} \tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i} &= \frac{1}{h^2} (c_i m_i \mathbf{d}_i \alpha_i + \sum_{j=1}^n c_j h_{ij} m_j (-\mathbf{d}_i) (1 - \alpha_i)) \\ &= \frac{1}{h^2} \mathbf{d}_i (c_i m_i \alpha_i - \sum_{j=1}^n c_j h_{ij} m_j (1 - \alpha_i)) \end{aligned} \quad (5.21)$$

The definition of α_i (5.12) results in

$$\tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i} = 0 \quad (5.22)$$

□

Proof (Global conservation of momentum): Since it has been shown that for a collision (i, T_i) , the local contact forces sum to zero, it follows that the sum of local forces over all collisions is also zero. By summing up all local forces per point, it is shown that the contact forces \mathbf{f}_i are obtained. It follows that the contact forces sum to zero.

To calculate the contact force \mathbf{f}_i per point, the coupling of contacts has to be considered. Thus, the sum over all collisions C is obtained by summing over all n rows of \mathbf{H} ,

$$\sum_{i \in C} (\tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i}) = \frac{1}{h^2} \sum_{i=1}^n \left(c_i m_i \alpha_i \mathbf{d}_i + \sum_{j=1}^n c_j h_{ij} m_j (-\mathbf{d}_i) (1 - \alpha_i) \right) \quad (5.23)$$

Now the assumption (5.14) is employed, notably that $-\mathbf{d}_i = \mathbf{d}_j$ for each node j belonging to the contact triangle of the contact point i . Moreover, $\alpha_j = 1 - \alpha_i$ for each node j belonging to i 's contact triangle, which is a direct result from (5.18). Thus

$$\sum_{i \in C} (\tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i}) = \frac{1}{h^2} \sum_{i=1}^n \left(c_i m_i \alpha_i \mathbf{d}_i + \sum_{j=1}^n c_j h_{ij} m_j \mathbf{d}_j \alpha_j \right) \quad (5.24)$$

$$= \frac{1}{h^2} \left(\sum_{i=1}^n c_i m_i \alpha_i \mathbf{d}_i + \sum_{i=1}^n \sum_{j=1}^n c_j h_{ij} m_j \mathbf{d}_j \alpha_j \right) \quad (5.25)$$

Now a little trick can be applied to the second summation in (5.24): Since this double sum basically iterates over all entries of \mathbf{H} , the order of summation can be exchanged. That means, instead of summing over all rows and then over the columns, it is equally possible to sum over all columns, and then over the rows,

$$= \frac{1}{h^2} \left(\sum_{i=1}^n c_i m_i \alpha_i \mathbf{d}_i + \sum_{i=1}^n \sum_{j=1}^n c_i h_{ji} m_i \mathbf{d}_i \alpha_i \right) \quad (5.26)$$

What is gained with this transformation? The answer reveals if (5.26) is rewritten as

$$= \frac{1}{h^2} \sum_{i=1}^n \left(c_i m_i \alpha_i \mathbf{d}_i + \sum_{j=1}^n c_i h_{ji} m_i \mathbf{d}_i \alpha_i \right) \quad (5.27)$$

Obviously, within the brackets, all quantities appear with the index i . In other words, the quantities within the bracket all correspond to the same contact point i . Apparently, this is the result that was asked for, notably the sum of all local contact force contributions arising from all collisions where i is involved into. Thus,

$$\sum_{i \in C} (\tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i}) = \frac{1}{h^2} \sum_{i=1}^n c_i m_i \alpha_i \mathbf{d}_i \left(1 + \sum_{j=1}^n h_{ji} \right) \quad (5.28)$$

And by using the definition of c_i

$$\sum_{i \in C} (\tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_{T_i}) = \sum_{i=1}^n \underbrace{\frac{m_i}{h^2} \alpha_i \mathbf{d}_i}_{=\mathbf{f}_i} = \sum_{i=1}^n \mathbf{f}_i = 0 \quad (5.29)$$

It has been shown that by reordering the terms in the sum of local forces over all collisions (left hand side of (5.29)), the sum of contact forces over all collisions (right hand side of (5.29)) is obtained, which conforms to the sum of contact forces over all contact points. Thus the sum of local forces acting on a single contact point is

$$\mathbf{f}_i = \frac{m_i}{h^2} \alpha_i \mathbf{d}_i \quad (5.30)$$

as claimed in (5.7). As a consequence, the proposed contact forces $\sum_{i \in C} \mathbf{f}_i$ sum to zero and thus conserve the momentum of the collision.

□

In (5.28), the fact that points $j \in T_i$ do not necessarily collide themselves and thus cannot be reordered for an $i \in C$ has been neglected. The case of the boundary points is discussed below.

5.2.7 Frictional contacts

To implement frictional contacts, a force component orthogonal to \mathbf{f}_i is computed using Coulomb's law, as discussed in Sec. 5.1.2. The friction is computed per

collision, i. e. between each contact point and its corresponding contact triangle. The friction force is approximated in the sense that the current contact force is neglected. Instead, the internal force is assumed as normal force, which is a fair assumption in most contexts.

The force component is then added to the contact force \mathbf{f}_i . Similar to [PPG04], it is assumed that frictional forces do not result in secondary collisions. This allows to decouple friction from the contact force computation.

5.2.8 Force post-correction

In the previous section, it has been shown that the proposed contact forces result in a collision-free configuration, and that the momentum is conserved during the collision. Thereby, simplifying assumptions have been made at several places. To summarize, it has been assumed that (a) the contact space is one-dimensional for each collision, (b) the points of a contact triangle do all themselves collide.

However, both assumptions obviously introduce errors in the force computation. Assumption (a) is only true if the contact surface is a planar, and therefore the computed contact spaces are all parallel. Assumption (b) is only true in the interior of the contact surface. However, at the boundary, there are always contact triangles having at least one point that does not collide. Consequently, for such a point, the contact space is 0, which furthermore contradicts the assumption (a).

The introduced errors reveal in two artifacts: First, the constrained configuration that results from the numerical time-integration is not entirely collision-free. Second, the computed contact forces do not conserve the momentum entirely. Still, these two artifacts do differ in their impact on the simulation. Having in mind that the surfaces of interacting elastic objects do deform and therefore ‘hide’ the contact points, the small overlaps can be tolerated. In contrast, if the momentum is not conserved, then the resulting ghost forces act as additional external forces on the two objects, making them drifting in an implausible way. This behavior is particularly bothersome in the simulation of resting contacts such as depicted in Fig. 5.5. Therefore, a way has to be found that alleviates the ghost force problem.

An obvious solution was to employ viscous damping that works against the ghost forces. However, viscous damping in turn limits the visual plausibility of the simulation. Here, an alternative way is proposed that guarantees the conservation of momentum. After having computed the contact forces \mathbf{f}_i for each contact point i , the contact forces are corrected as

$$\forall i \in C : \quad \mathbf{f}_j \leftarrow \frac{1}{2}\mathbf{f}_j - \frac{1}{2}h_{ij}\mathbf{f}_i \quad j \in T_i \quad (5.31)$$

In other words, the contact force of each point of all contact triangles is divided by two (nothing is done if j is not colliding). Moreover, the contact force is weighted by a factor h_{ij} and distributed onto the three points of the corresponding contact triangle. Notice that the set $j \in T_i : i \in C$ contains contact points and border points, i. e. points that do not collide themselves, but belong to a contact triangle.

Claim The corrected contact forces conserve the momentum exactly.

Proof The corrected contact forces are summed per point over all collisions:

$$\sum_{i=1}^n \left(\frac{1}{2} \mathbf{f}_i - \sum_{j=1}^n \frac{1}{2} h_{ij} \mathbf{f}_i \right) = \quad (5.32)$$

$$\sum_{i=1}^n \left(\frac{1}{2} \frac{m_i}{h^2} \mathbf{d}_i \alpha_i + \sum_{j=1}^n -\frac{1}{2} h_{ij} \frac{m_i}{h^2} \mathbf{d}_i \alpha_i \right) = \quad (5.33)$$

$$\sum_{i=1}^n \left(\frac{1}{2} \frac{m_i}{h^2} \mathbf{d}_i \alpha_i - \frac{1}{2} \frac{m_i}{h^2} \mathbf{d}_i \alpha_i \underbrace{\sum_{j=1}^n h_{ij}}_{=1} \right) = 0 \quad (5.34)$$

□

The motivation of this correction is two-fold: First, the momentum is now conserved exactly, and the corresponding ghost forces are inhibited. Second, the border points, i. e. those points that have not been colliding, are now as well subject to a contact force. This corresponds nicely to the fact that these border points must as well be constrained, if edge collisions would have been considered. On the other hand, the drawback of this post-correction is that it somehow contradicts the mathematical justification made before. Consequently, no statement about the non-penetration property of the constrained configuration can be made. Still, experiments indicate that the resulting overlaps are tolerable.

5.3 Results

The implementation has been tested on various problems, ranging from complex off-line computations to interactive animations. The goal of this discussion is to illustrate two key benefits of the proposed approach, notably that the non-penetration constraints are maintained, even under high external load, and that resting states can be simulated plausibly. Further, the performance of the approach is evaluated in a massive n-body simulation. All experiments have been performed on an Intel Xeon PC, 3.8 GHz using an nVidia Quadro FX 4500 graphics card. The meshes have been generated with the approach presented in Appendix A. The deformations have been computed with an efficient co-rotational FE method [MG04].

Non-penetration constrain maintenance The employ of a predictor-corrector methodology enables the computation of contact forces that correct interfering primitives. Thereby, a collision-free configuration is maintained throughout the simulation. In addition, the method comes with no user-defined parameters, therefore collision configurations with temporally and spatially varying stress can be consistently handled. In Fig. 5.4, elastic bars are squeezed between two fixed anchors and thus they are deformed. Due to the low elasticities of the bars (Young



Figure 5.4: Elastic bars are squeezed between anchors. A collision-free state is maintained even if the stresses on the contact regions are enormous high. The bars release and return to their resting shape if a heavy object is disturbing the equilibrium.

modulus 100kN/m^2), the stress on the contact regions is enormous high. Nevertheless, a collision-free state is maintained. To show that no artificial gluing has been applied in this simulation, a heavy object is falling onto the bars. The violent push destroys the equilibrium, the bars slip off each other and return to their straight resting shape.

Frictional contacts The ‘archway’ (Fig. 5.5) is a classical problem in architecture. In fact, this construction can be built without employing any gluing substance. Its stability relies on a combination of pressure and static friction that prevents the elements from sliding off each other. By modeling Coulomb friction (Sec. 5.1.2), the simulation of a stable resting state of the ‘archway’ is enabled, even if its equilibrium is disturbed by objects falling onto it. Each wedge-shaped element consists of 375 tetrahedra, has a Young modulus of 20kN/m^2 , a Poisson ratio $\nu = 0.35$, and a static friction coefficient of 0.95. Computing the deformation forces takes 2.1ms, detecting the collisions and computing the contact space takes 17.2ms, computing the contact forces takes 2.5ms, and the time-integration takes 0.4ms. Together, computing one simulation pass takes 22.2ms.

Performance The complexity of the approach is linear in the number of contact points. To illustrate this, an experiment with 500 objects being dropped into a container is performed (see Fig. 5.6). Each object has a Young modulus of 10kN/m^2 . The total number of mass points is 50.4K, and the total number of tetrahedrons is 88.2K. Fig. 5.7 indicates that the time to respond to collisions depends linearly on the number of collisions.

Fig. 5.1 illustrates the simulation of 200 chained torii falling onto an elastic membrane. The membrane deforms and buckles under the weight of the chain, but nevertheless a stable resting state is reached. The overall number of tetrahedrons is 80K, and there are 2500 contacts in average. The computation of one simulation pass takes 470ms in total, and the collision response alone takes 41ms.

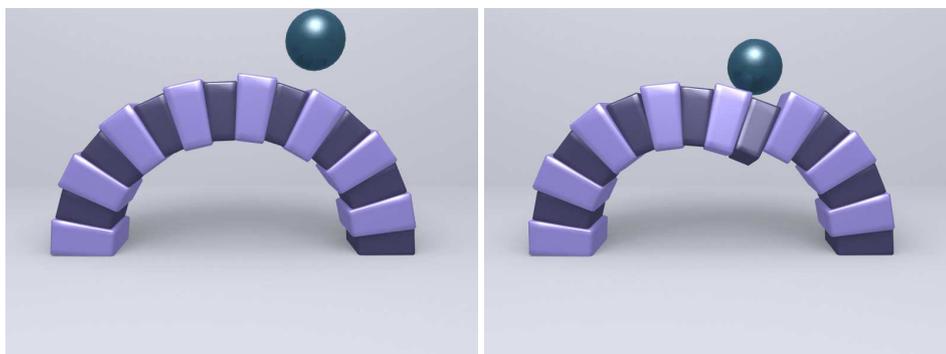


Figure 5.5: Simulation of a classical problem in architecture. The sensitive equilibrium of static friction and pressure makes this configuration a challenge for the collision response. Each element consists of 375 tetrahedra. Computing one simulation step (including deformation, collision detection and response, and time-integration) takes 22.2ms, and there are 400 contact points in average.



Figure 5.6: To test the performance of the approach, 500 deformable chess figures are filled into a container. Thus the number of collisions grows linearly in time.

5.4 Discussion

Up to the time of publishing this approach, there has been surprisingly few activity in the field of deformable collision response. Most existing schemes have employed simple penalty approaches to accomplish the response [FL01, HFS03, KMH*04]. However, since penalty approaches relate the contact force to an interpenetration measure, they are either inefficient or inaccurate. Further, the choice of the spring constant and the related time step can be difficult. A couple of works have proposed to compute contact forces in an analytic way, by imposing non-penetration constraints and solving the resulting LCP [PPG04, DDKA06]. The analytical schemes are accurate, but they address static equilibria in order to apply Signorini's law. Further, the resulting LCPs either require effortful implementations or iterative scheme.

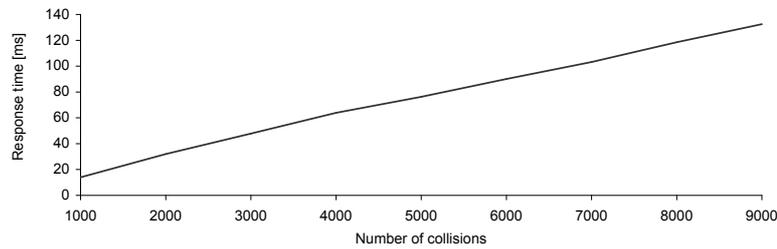


Figure 5.7: The measurements indicate that the performance of the proposed approach is linear in the number of collisions.

In this chapter, an approach has been presented that employs the predictor-corrector methodology to solve contact problems of interacting deformable objects. In contrast to previous analytical methods, the approach addresses static contacts as well as dynamic collisions. For dynamic collisions, there is no pressure equilibrium at the contact surface. Instead, the computation of the contact surface, i. e. the non-interfering configuration of the contact points is governed by Newton's third law which demands that forces must sum to 0 (which conforms to conserving the momentum). The idea behind the approach is to compute for each colliding primitive a non-interfering goal position. Then, a contact force is computed that, if numerically time-integrated, makes the primitive meet its goal position immediately. This technique is also called *direct forcing*.

While the derivation of the contact forces from the goal positions is straightforward, the computation of these non-interfering goal positions is a challenging problem. This is motivated by the observation that contact points cannot be considered in isolation: In order to guarantee the conservation of momentum, both the contact point and the corresponding contact triangle on the discretized surface of the opposite object must be displaced. Still, the corners of the contact triangle might themselves be contact points. The result is a densely coupled network of contact points and contact triangles that forbids local computations.

The presented approach faces this problem by taking some simplifying assumptions, based on the observation that small overlaps are tolerable while the conservation of momentum must be strictly guaranteed. For each pair of contact point and corresponding contact triangle, the contact space is assumed to have dimension one. Thus, the problem reduces to finding scalar displacements that resolve the interference while conserving the momentum. After having solved for the local displacements, these displacements are summed per contact point (knowing that for each contact point, multiple displacements have been computed). The resulting global displacement then conserves the momentum up to a discretization error.

However, since in the discretized setting, these errors might become large and result in visual artifacts, a force post-correction scheme has been proposed. The corrected contact force will conserve the momentum exactly (independent from

the discretization). Further, a side-effect of the post-correction is that border points, i. e. those points being adjacent to a colliding edge but not colliding themselves, are now displaced as well. Thus, the edge collision that are not handled directly can nevertheless be minimized. In contrast, the resulting configuration might no longer be non-interfering. While experiments indicate that the resulting overlaps are hardly noticeable and do not disturb the plausibility of the simulation, the lack of a concise error analysis is a limitation of the approach.

Summarizing, the predictor-corrector methodology allows to compute contact forces that maintain non-interfering configuration throughout the simulation. In order to conserve the momentum exactly, a compromise in the accuracy of the non-penetration constraints has to be accepted. The compelling simplicity and efficiency of the approach is a striking benefit.

Future work In order to increase the theoretical soundness of the approach, a thorough error analysis should be made. By considering the underlying discretization, this analysis could give an idea on the magnitude of the resulting overlaps. Further, by employing an iterative scheme, these resulting overlaps could eventually be minimized, resulting in a scheme that both conserves the momentum and maintains an interference-free state.

Chapter 6

A global approach for the contact handling of elastic rods

In the previous chapter, an approach has been presented to handle contacts between interacting deformable objects. The approach has been motivated by the observation that the deforming surfaces of the objects must be non-overlapping after the collision response, and that the contact forces must sum to 0 in order to preserve the momentum. Further, the precondition has been imposed that the objects are not overlapping at the begin of the time-step. The approach has then temporally evolved the mass points in time, and then it has detected collisions of those unconstrained positions. Based on this information, contact forces have been computed that correct the unconstrained positions such that after re-integration, the non-penetration constraints are maintained. Therefore, the approach has been classified as a predictor-corrector scheme.

This chapter illustrates that a similar strategy can as well be employed to handle contact problems of interacting elastic rods. The goal is to simulate knots, which is a particularly difficult problem. In the context of the contact handling, elastic rods share similarities with the contact handling of thin shells or clothing. To see why, one has to consider that thin shells are commonly modeled with surfaces, without explicitly representing the volume. In contrast to interacting volumetric objects, interpenetrations do therefore instantaneously result in an inconsistent state, with a region of the cloth being on the ‘wrong’ side of the other cloth. Therefore, the penalty method that relates the contact force to a measure of the violation of the constraint will not work for such objects. Consequently, in the past, approaches have been proposed that compute repulsion forces that prevent collisions rather than trying to resolve them [BW98, BFA02]. Another strategy has been proposed by Volino and Magnenat-Thalmann: If an interference between two pieces of cloth has been detected, then the approach minimizes the intersection contour, thereby resolving the collision [VMT06].

In Chapter 3, elastic rods have been modeled with simple chains of mass points that represent the centerline of the elastic rod. Similar to thin shells, the volume

is thereby not explicitly represented, and consequently, interpenetrations must be avoided. In the contact handling of elastic rods, it has to be considered that an interpenetration cannot be determined with local tests, which is in contrast to thin shell contact handling. This comes from the fact that an elastic rod does not even cover a surface. Consequently, each configuration is valid unless the temporal course of the simulation is considered. This observation forbids an approach in the spirit of [VMT06]. Instead, interpenetrations have to be strictly avoided.

Having this in mind, one is tempted to employ strong repulsion springs that exert forces on the rod segments in order to prevent an interpenetration. However, repulsion springs basically suffer from the same drawbacks as penalty forces do, notably that they must be stiff in order to be physically plausible, but in turn require small steps in order to not to overshoot. And considering that in knot-tying simulations, the resulting elastic forces can be very strong, the repulsion method is obviously not a clever choice.

Instead, in this chapter, the inspiration comes from the rigid object contact methods. There, interpenetrations are usually strictly avoided by formulating non-penetration constraints. And since in stacking problems, these non-penetration constraints are conflicting, the approaches usually assemble all constraints in one system of equations, which is then solved globally [Bar89, GBF03]. This then allows to simulate famous problems such as the Newton's pendulum, where impulses are traveling immediately through stacked objects.

In this chapter, a method is presented that combines the global non-penetration approach in rigid object contact handling with the predictor-corrector approach that has been discussed in the previous chapter. First, the unconstrained positions of the elastic rods are predicted by time-integration. Then, collisions are detected and non-penetration constraints are formulated per pair of interacting rod segments. Based on these constraints, feasible configurations are computed locally that preserve the momentum. By iteratively repeating this procedure, positions are obtained such that the non-penetration constraints are satisfied globally. Finally, contact forces can be computed such that those constrained positions are met after time-integration, which is similar to the method presented in the previous chapter. The proposed approach allows to simulate challenging knot configurations, such as depicted in Fig. 6.1.

An introduction into contact mechanics has earlier been given in Sec. 5.1. In Sec. 6.1, the approach to handle collisions between elastic rods is presented. Results and experiments are found in Sec. 6.2, and a discussion is given in Sec. 6.3.

6.1 Approach

In this section, an approach is presented to handle collisions and self-collisions of interacting elastic rods. The scheme follows the predictor-corrector methodology and is therein similar in spirit to the approach presented in Sec. 5.2. Since the cross-section of elastic rods is assumed to be rigid, the approach computes contact

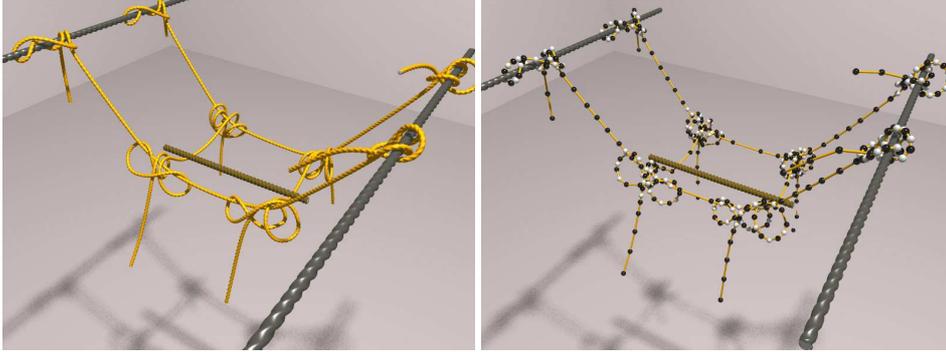


Figure 6.1: The contact handling of elastic rods is similar in spirit to rigid body collision handling, since the cross-section of elastic rods is rigid. The proposed approach can be employed to simulate knots. Here, a network of four Prusik knots carries an object.

forces that maintain the non-penetration constraints globally. To accomplish this, an iterative scheme is employed.

6.1.1 Object representation

In the following discussion, it is assumed that the centerline of the elastic rod is discretized into spatial control points $\mathbf{r}_i \in \mathbb{R}^3, i = 1..N$. The contact handling is then performed on these control points by enforcing a minimum distance between the segments $S_i = (\mathbf{r}_i, \mathbf{r}_{i+1})$. This results in contact forces \mathbf{f}_i that act as external forces on the points. No further assumption is made about the deformation model or the representation of the orientation. Consequently, the contact handling works with any deformation model which explicitly represents the centerline, e. g. the discrete model of Bergou *et al.* [BWR*08], the geometrically exact spline model of Theetten *et al.* [TGAB07], or the CORDE model presented in Chapter 3.

6.1.2 Predictor-corrector simulation

The approach follows the predictor-corrector methodology that has already been presented in Sec. 5.2. First, the rods are evolved in time by employing the Verlet-scheme, resulting in unconstrained future positions. Then, the collisions are detected in the (possibly) interfering state. To accomplish this, hierarchies of axis-aligned bounding boxes are employed to detect collisions and self-collisions. Details are found in [TKH*05]. Similar to Müller [MHHR07], non-penetration constraints are imposed on the primitives. Then, the minimum displacements are computed that satisfy the constraints. By applying constraint forces, the rods meet the non-penetration constraints after evolution. Notice that since collisions are detected at discrete times, collisions can be missed in case of large relative velocities. The simulation loop is shown in Algorithm 2. The algorithm is similar to Algo-

rithm 1, with the only difference that here, the computation of the contact spaces is carried out in each iteration of the contact force computation.

```

repeat
   $\boldsymbol{\sigma}(t) \leftarrow \text{ElasticForces}(\mathbf{x}(t), \dot{\mathbf{x}}(t));$ 
   $\mathbf{f}^e(t) \leftarrow \text{ExternalForces};$ 
   $\tilde{\mathbf{x}}(t+h) \leftarrow \text{UnconstrainedMotion}(\mathbf{x}(t), \mathbf{v}(t), \boldsymbol{\sigma}(t));$ 
   $C \leftarrow \text{DetectCollisions}(\tilde{\mathbf{x}}(t+h));$ 
   $\mathbf{f}(t) \leftarrow \text{ComputeContactForces}(C, \mathbf{x}(t), \tilde{\mathbf{x}}(t+h), \mathbf{d}(t+h), \boldsymbol{\sigma}(t), \mathbf{f}^e(t));$ 
   $\mathbf{x}(t+h) \leftarrow \text{ConstrainedMotion}(C, \mathbf{x}(t), \mathbf{v}(t), \boldsymbol{\sigma}(t) + \mathbf{f}^e(t) + \mathbf{f}(t));$ 
   $t \leftarrow t+h;$ 
until stop;

```

Algorithm 2: The main loop of a predictor-corrector based simulation approach first computes the elastic forces and then temporally evolves the configuration. The contact handling is then performed on the unconstrained configuration, and the re-integration results in the constrained configuration for the next time step.

6.1.3 Constraint-based contact forces

In this section, the system of inequations that governs the collision-free configuration is derived. Afterwards, an iterative solution of this system of equations is provided in the subsequent section. Let

$$\text{md}(i, j) = \text{md}(\mathbf{r}_i, \mathbf{r}_{i+1}, \mathbf{r}_j, \mathbf{r}_{j+1}) > 0$$

be the minimum Euclidean distance between two rod segments S_i and S_j . Then the collision detection procedure provides a set of *collisions* (S_i, S_j) , i. e. segments S_i and S_j for which $\text{md}(i, j) < d$. Here, $d = r_i + r_j$ is the enforced minimum distance between the rod segments, with r_i and r_j the radii of the rods. The penetration depth $\varepsilon_{ij} > 0$ of a collision (S_i, S_j) is then $d - \text{md}(i, j)$.

The contact space of a collision (S_i, S_j) is defined to be spanned by the minimum distance vector \mathbf{n}_{ij} between S_i and S_j . It is then looked for displacements $\Delta \mathbf{r}_i$ such that the coordinates $\mathbf{r}_i + \Delta \mathbf{r}_i$ constitute an interference-free configuration. The displacement $\Delta \mathbf{r}_i$ is a linear combination

$$\Delta \mathbf{r}_i = \sum_{j \in \mathcal{C}(i)} \mathbf{n}_{ij} \chi_{ij} \quad (6.1)$$

with $\mathcal{C}(i)$ the set of all pairs of interfering segments that \mathbf{r}_i is adjacent to, and unknown scalars χ_{ij} . The non-interference conditions for the set of collisions are written as

$$\begin{aligned} \text{md}(\mathbf{r}_i + \Delta \mathbf{r}_i, \mathbf{r}_{i+1} + \Delta \mathbf{r}_{i+1}, \mathbf{r}_j + \Delta \mathbf{r}_j, \mathbf{r}_{j+1} + \Delta \mathbf{r}_{j+1}) &\geq d \\ \text{md}(\mathbf{r}_k + \Delta \mathbf{r}_k, \mathbf{r}_{k+1} + \Delta \mathbf{r}_{k+1}, \mathbf{r}_l + \Delta \mathbf{r}_l, \mathbf{r}_{l+1} + \Delta \mathbf{r}_{l+1}) &\geq d \\ &\vdots \end{aligned} \quad (6.2)$$

As a second condition, the momentum must be conserved,

$$\sum_i \Delta \mathbf{r}_i m_i = 0 \quad (6.3)$$

According to Gauss' principle of least work (see e. g. [RKC02]), it is looked for the minimum displacements that result in an interference-free configuration, i. e. it is required that

$$\sum_i \|\Delta \mathbf{r}_i\|_2 \rightarrow \min \quad (6.4)$$

Together, these equations constitute a non-linear optimization problem. This formulation is similar to the optimization-based algorithm in [MS01] with the difference that here, the conservation of momentum is explicitly enforced.

6.1.4 Iterative solution

Computing the exact solution of the system of equations (6.2-6.4) is hardly feasible at interactive rates. Instead, an iterative solution in the spirit of [GBF03, MHR07] is proposed to obtain interference-free positions. First, interacting collisions are grouped to *impact zones* (according to the definition of Provot [Pro97]). In contrast to [HMB01, BFA02], these impact zones are not treated as rigid. Then, it is solved for feasible positions for each collision individually: Let

$$\xi = \frac{m_j w_j + m_{j+1} (1 - w_j)}{m_i w_i + m_{i+1} (1 - w_i) + m_j w_j + m_{j+1} (1 - w_j)} \quad (6.5)$$

be the barycentrically weighted ratio of masses that accounts for conservation of momentum of the collision (S_i, S_j) , with w_i being the barycentric coordinates of the contacts on the line segments. The m_i are the lumped masses in the points i . In case of a continuous mass distribution, the values m_i and m_{i+1} should reflect the weight of the segment i , and can be computed by assuming an arbitrary shape function. Notice that $\xi = \frac{1}{2}$ if the rod segments have equal weights. Then the collision displacements

$$\begin{aligned} \Delta \mathbf{r}_i &= \mathbf{n}_{ij} \xi (\text{md}(i, j) - d) w_i \\ \Delta \mathbf{r}_{i+1} &= \mathbf{n}_{ij} \xi (\text{md}(i, j) - d) (1 - w_i) \\ \Delta \mathbf{r}_j &= \mathbf{n}_{ij} (1 - \xi) (d - \text{md}(i, j)) w_j \\ \Delta \mathbf{r}_{j+1} &= \mathbf{n}_{ij} (1 - \xi) (d - \text{md}(i, j)) (1 - w_j) \end{aligned} \quad (6.6)$$

effect that the penetration depth ε_{ij} for this collision is at least halved, if the collision is considered in isolation. It is accounted for spatial continuity by weighting the displacements with the barycentric coordinates of the collision. By processing all collisions per impact zone in a sequential manner, and summing the displacements per point \mathbf{r}_i , displacements are obtained that conserve the momentum (6.3). This process constitutes one step of the iterative scheme.

By repeating this process, displacements quickly propagate through the impact zone and secondary collisions are resolved (impact zones are merged if segments from different impact zones become colliding). The iterative search is stopped if $\varepsilon_{ij} < \varepsilon_{\max}$ for all collisions (S_i, S_j) in the impact zone, where ε_{\max} is the error tolerance. Experiments indicate that the number of iterations depends linearly on the number of contacts, as expected. The number of iterations could be reduced by employing a contact graph, as proposed in [GBF03]. However, building a contact graph for knots is not straight-forward.

To guarantee a certain frame rate, the process is interrupted after a pre-defined number of iterations. It is then hoped that the remaining collisions can be resolved in the subsequent simulation step. In the case of elastic rod contacts, this situation occurs very rarely. Still, the robustness of the method could be augmented by employing the fail-safe that has recently been proposed in [HVTG08].

6.1.5 Friction

The incorporation of frictional forces complicates the problem. Still, friction is crucial for simulating knots. Analytical solutions approximate the friction by a friction cone [Bar91, DDKA06]. In contrast, here, a position-based approximative solution for frictional effects is proposed that is consistent with the iterative method. For each collision (S_i, S_j) , a Coulomb friction force component \mathbf{f}^{fric} is computed that is orthogonal to the contact space. Then, the friction displacement $\Delta \mathbf{r}_i^{\text{fric}}$ for the point \mathbf{r}_i is approximated as $\Delta \mathbf{r}_i^{\text{fric}} = \mathbf{f}^{\text{fric}} w_i \frac{h^2}{m_i}$, where the mass m_i is assumed to be lumped in the points \mathbf{r}_i . Further, h is the time step. In each iteration and for each collision, the friction displacements is added to the collision displacements. To account for the exponential decrease of the collision displacements in the course of the iterative computation, the friction displacements $\Delta \mathbf{r}_i^{\text{fric}}$ are weighted with a factor $2^{-\text{iter}}$, where iter denotes the current iteration. While this approximative friction produces plausible results at minor computational overhead, more elaborated models could eventually improve the accuracy and support perfect sticking.

6.1.6 Contact forces

Having computed the displacements $\Delta \mathbf{r}_i$ that yield feasible positions $\mathbf{r}_i + \Delta \mathbf{r}_i$, a contact force is computed that accelerates the mass points towards the feasible position. More precisely, contact forces \mathbf{f}_i are sought that, if time-integrated numerically at time t , result in the feasible positions $\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + \Delta \mathbf{r}_i$. Algebraic transformations lead to $\mathbf{f}_i = \frac{1}{h^2} \Delta \mathbf{r}_i m_i$. Similar formulations have been successfully applied for geometric constraint maintenance [GBT06] and hair dynamics [CCK05]. Since this formulation considers the simulation time step, instabilities due to large contact forces are largely inhibited. Details are found in Chapter 5.

The computed displacements and corresponding contact forces result in entirely inelastic collisions. This corresponds nicely to the fact that the textile material of ropes and threads is best modeled with inelastic collisions.

6.1.7 Contact handling in the adaptive simulation

If the contact method is employed together with an adaptive model, then it has to be considered that the contact space might change discontinuously if the rod is refined. Moreover, if a variational control placement (see Sec. 4.1) is employed, then secondary collisions might occur during the displacement.

More precisely, when the rod is refined, then control points \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ are displaced in order to minimize the elastic energy over the new elements (see Sec. 4.1.3). The iterative displacement of the control points in the CG method eventually results in secondary interferences, which in turn causes instabilities. These interferences are avoided by detecting collisions of the bisected elements in each CG iteration (which can be done efficiently since the potentially colliding elements are already known in advance). By considering the resulting contact force in the static equilibrium functional (4.3), interference-free control points are guaranteed.

6.2 Results

In this section, an evaluation of the proposed contact handling scheme is given. The goal is to experimentally show the quadratic complexity of the scheme, and to illustrate that the simulation of knots is enabled by modeling the Coulomb friction. The experiments have been performed on an Intel Xeon PC, 3.8Ghz. The deformations of the elastic rods have been computed with the CORDE deformation model presented in Chapter 3.

Complexity The physically motivated collision resolving scheme iteratively computes feasible collision-free positions of interfering rod elements. In each iteration, all contacts are processed to determine the locally feasible positions. Since displacing one rod element produces secondary collisions, the number of iterations depends linearly on the number of colliding elements. As a consequence, the complexity of the scheme is in $\mathcal{O}(C^2)$ with C the number of contacts. Still, since computing locally feasible positions is a cheap operation, the running times are comparable to state-of-the-art response schemes. Fig. 6.2 relates the time to compute feasible positions to the number of contacts and underlines the theoretical result.

Application In the Coulomb friction model, the magnitude of the friction force is governed by the magnitude of the contact force. Therefore, the employ of a Coulomb friction model is mandatory in the knot simulation in order to guarantee the stability of tight knots. Fig. 6.3 illustrates a simulation of the shoelace knot. Here, a thread is tied around four poles and linked with the shoelace knot. The stretched thread exerts large contact forces on the poles, which in turn prevents that the thread slips down. In this challenging simulation, the user can interact

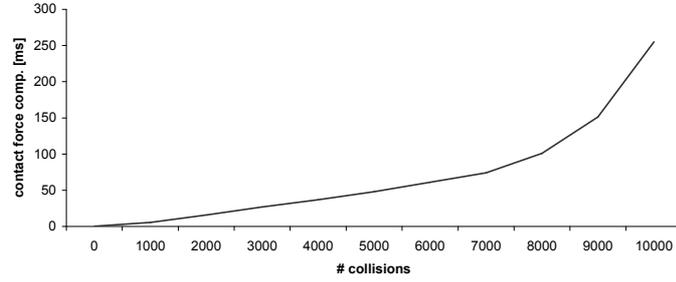


Figure 6.2: The complexity of the global response scheme is in $\mathcal{O}(C^2)$. However, usually fewer than C iterations are necessary to resolve C collisions with sufficient accuracy. In the experiment, an error tolerance $\epsilon_{\max} = 10^{-3}r$ has been considered, with r the radius of the rod.

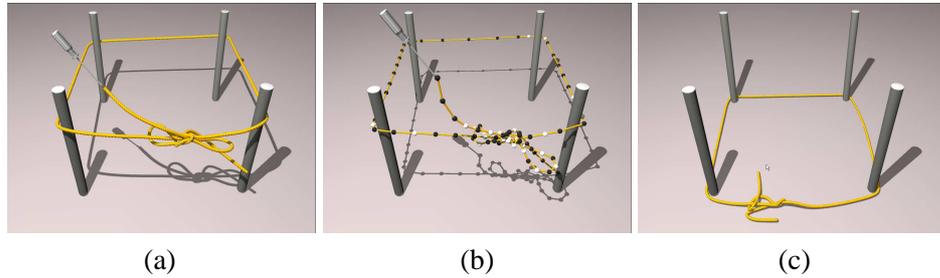


Figure 6.3: Interactive simulation of adaptive knot-tying. (a) A thread is tied around four poles, and a shoelace knot prevents it from slipping down. (b) To accurately represent the knot, an adaptive technique as presented in Chapter 4 is employed. (c) As the user unties the knot, the contact forces at the poles are reduced and the thread slips to the ground. This simulation illustrates the effect of Coulomb friction.

with the thread. When the user unties the shoelace knot, the contact forces in the poles are reduced, and the thread slips down.

To show that complex self-contact configurations can be resolved, a simulation of a falling coiled rope with 2K control points is run, as illustrated in Fig. 6.4. The simulation time step is 0.1ms. Collision detection of 10K collisions takes 355ms and computing feasible positions takes 255ms. The average initial penetration depth is $\epsilon = 1.8 \cdot 10^{-3}r$, with r the radius of the rod. 34 iterations are necessary to reduce the penetration depth to $\epsilon < \epsilon_{\max} = 10^{-3}r$. Force computation and integration takes 100ms, the simulation runs at 1.5 frames per second.

6.3 Discussion

In this chapter, an approach to handle collisions between elastic rods has been presented. Since elastic rods have a negligible volume compared to their longitudinal

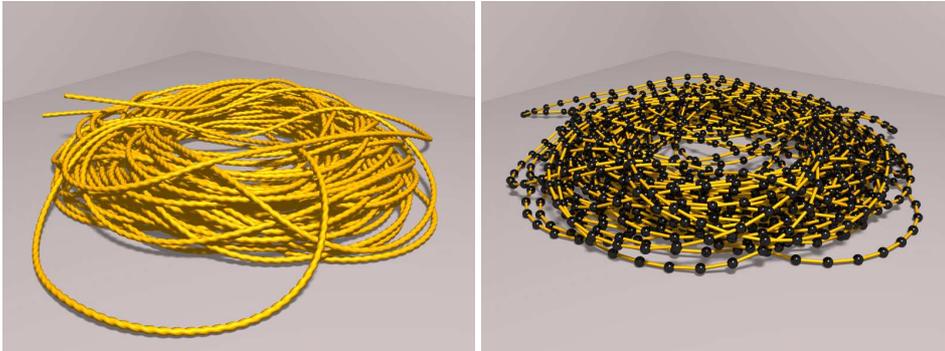


Figure 6.4: Simulation of a coiled rope. The rope consists of 2K control points, with 10K collisions in average. This experiment illustrates global collision response in the context of massive self-collisions. Since most elements are involved in collisions, this simulation represents the worst-case scenario for adaptivity, as the statistics in Tab. 4.1 in Sec. 4.3 indicate.

extent, their cross-section is assumed to be rigid. If this assumption was dropped, then their volumes could quickly be compressed to 0, resulting in interferences. Still, detecting interferences cannot be done with local tests as for cloth collisions, but requires global tests. As a consequence, interferences must be strictly prevented.

If, in turn, the rod's cross-section is assumed to be rigid, then the contact handling cannot be done locally. This is because multiple rods can be interfering instantaneously, and resolving the collision between any two rod segments cannot be done without inducing secondary collisions. Thus, the contact handling of elastic rods shares similarities with the contact handling for rigid bodies, which in turn served as a source of inspiration for the proposed scheme. Consequently, the proposed approach is similar in spirit to the approach of Milenkovic and Schmidl [MS01], and Müller *et al.* [MHR07]: First, optimal constrained positions are computed, thereby preserving the linear momentum. Second, impulsive contact forces correct the unconstrained positions to meet the constrained positions after the numerical time-integration. This technique is also known as predictor-corrector simulation.

To resolve the collisions globally, an iterative scheme has been proposed. The collisions are resolved locally in a sequential way. The proposed computation guarantees that the linear momentum is preserved. By iteratively repeating this process, the displacements are quickly propagated through the stack of rods. Having computed these optimal collision-free positions, contact forces can be derived that make the mass points meeting those constrained positions. This derivation is similar in spirit to the method presented in Chapter 5. To model the Coulomb friction, an approximative model has been proposed which assumes that the contact and the tangent spaces are not coupled. This allows to simply add the Coulomb friction component to the contact force component.

The benefit of the proposed approach is its stability: Since the constrained positions in the subsequent time step are already known before the time-integration, the overshooting problem is avoided. The contact simulation remains stable, even for large time steps, which is in contrast to penalty approaches. The drawback of the approach is that the non-penetration constraints are only maintained up to a given tolerance, but never exactly. While this does not disturb the quality of the knot simulation, it makes the scheme inappropriate for mechanical simulations requiring a high accuracy.

Future work The approach in its previous form cannot simulate perfect sticking. That is, the “box on an inclined plane” test [GBF03] would fail if the box was simulated with the proposed approach. This comes from the fact that the prediction of the future positions is governed by the external forces, thereby accelerating the box vertically downwards. The contact handling would then compute the optimal position as being perpendicular to the plane. As a result, the box would slowly slip down the plane. To overcome this problem, the idea of Guendelman *et al.* could be adopted, notably a separate update of the positions and velocities [GBF03], with the contact handling carried out in between.

Another problem is that by adding the friction force component to the contact force, the approach cannot guarantee that the computed constrained positions are met exactly. In turn, this can result in secondary collisions. To overcome this problem, the friction force component is not added at the end of the iterative process, but a weighted friction component is added in *each* iteration. This inhibits the problem of secondary collisions, but does not completely avoid it. Still, a more elaborated friction model would be desirable.

Chapter 7

Conclusion

The physically-based simulation of one-dimensional deformable objects – also termed *elastic rods* – is a particularly challenging problem. In this thesis, approaches have been proposed to compute the deformations and to handle collisions of elastic rods. In contrast to previous approaches in these fields, the focus has been on simple and efficient algorithms that nevertheless provide the necessary physical plausibility. The goal has been to reproduce the characteristic deformation phenomena of elastic rods, and to simulate complex knots at interactive rates. Since elastic rods share similarities with both rigid and volumetric deformable objects, an investigation into this research field has not been possible without considering other simulation issues that are not directly related to the simulation of elastic rods. Consequently, additional contributions have been proposed in the appendix that round out the thesis.

Deformation modeling Based on the observation that no dynamic deformation model for one-dimensional deformable objects existed that considers torsional shearing strain at interactive rates, Chapter 3 has investigated into a novel deformation model for elastic rods. The centerline of the rod is discretized into mass points, and the material direction is discretized into orientation nodes. Further, quaternions are employed to represent the orientations. To adapt the material frames to the centerline, the penalty method is employed. Thus, the mass points and the quaternions are time-integrated independently, thereby avoiding the solve of a system of equations and enabling the interactive animation.

The proposed deformation model CORDE effectively constitutes the first deformation model for elastic rods which allows for the dynamic simulation of twisting rods at interactive rates. Previous approaches have either been static, or did not treat the bending and torsion strains in a consistent manner. Moreover, since the centerline is explicitly represented, the contact handling is eased, as later illustrated in Chapter 6. To evaluate the deformation model, a comparison with a mechanically accurate reference deformation model has been made. Further, many examples have illustrated the wide applicability of the proposed model.

Adaptive simulation In order to improve the efficiency of one-dimensional deformation models, an adaptive technique has been considered in Chapter 4. It bases on the observation that a high resolution is mandatory to represent the complex contact configurations that arise in e. g. the simulation of knots. In contrast, a low resolution is usually sufficient to represent the undeformed parts of the rod.

In the adaptive refinement, a variational principle is employed that places the new control points such that the elastic energy over the refined region is minimized. Thus, post-stabilization schemes that damp the motion of the rod are avoided. Experiments have been presented that evaluate the gain of efficiency over non-adaptive models.

Contact handling of volumetric deformable objects The contact handling of interacting deformable objects can be considered as a special case of constrained simulation. It is particularly challenging since the non-penetration constraints are temporally varying. If friction is modeled, then the resulting equations are non-linear and forbid analytical solves at interactive rates. In order to support the understanding of the contact handling of elastic rods, the thesis has first proposed a novel approach to handle contacts between interacting volumetric objects. This has enabled the understanding of the contact handling approach for elastic rods that has been presented subsequently.

To model the interaction between volumetric deformable objects, the penalty-method has usually been employed. However, stiff penalty forces require small time-steps. This problem has been circumvented by a predictor-corrector approach presented in Chapter 5. The approach predicts the unconstrained positions that are governed by the momenta of the interacting objects. Then, impulsive contact forces are computed that make the collided points meet their collision-free positions. Further, these contact forces preserve the linear momentum.

The main benefit of the proposed method is that it computes the contact forces analytically, without requiring iterative solvers. Further, the computation is locally, without requiring the solve of a system of equations. In addition, the approach comes without user-defined stiffness parameters, which is in contrast to existing penalty methods. Therefore, the non-penetration constraints can be maintained efficiently. Experiments of both challenging resting states and dynamic collisions have been performed.

Contact handling of elastic rods In contrast to volumetric deformable objects, elastic rods have a rigid cross-section. As a consequence, collisions cannot be resolved locally, but must be considered globally. Therefore, an iterative scheme for the contact handling of elastic rods has been presented in Chapter 6. Similar to previous rigid body collision schemes, the approach computes optimal collision-free positions. Then, impulsive contact forces are computed that make the collided rod segments meet their collision-free positions while preserving the linear momentum.

The proposed approach maintains a collision-free configuration throughout the simulation, even for large time-steps. This contrasts the repulsion methods that fight interferences with stiff forces. In turn, small time-steps are mandatory to avoid the overshooting problem. To illustrate the conceptual advantage of the method, experiments of complex knots have been presented.

Additional contributions The deformation modeling and the handling of constraints are identified to be the main challenges in the physically-based simulation. Still, there are other important research areas that need to be addressed accordingly. In order to simulate objects, their domains must be spatially discretized with an appropriate mesh generator. However, the existing meshing approaches usually require manifold and orientable input surfaces, which is often not satisfied by surfaces resulting from laser scans or CAD programs. Therefore, in Appendix A, an approach has been presented to produce tetrahedral meshes from arbitrary input surfaces. By circumventing any pre-processing, the mesh generation process is eased. The resulting meshes are suitable for the interactive simulation, as illustrated in Chapter 5.

Moreover, the collision detection is a mandatory task, which is often identified as the bottleneck in the physical simulation. However, an output-sensitive collision detection for deformable objects can be realized if the deformation model is considered. Thus, a novel approach to detect collisions of geometric deformable objects has been proposed in Appendix B. By updating a bounding sphere hierarchy in an output-sensitive way, the complexity of the collision query becomes independent from the number of simulated points. In contrast, it depends on the number of colliding points. Although the approach has no apparent relation to the simulation of elastic rods, it has provided an important step in the realization of the simulation framework.

7.1 Outlook

Although the proposed deformation model CORDE, along with the proposed contact handling method, allows for the efficient and robust simulation of interacting elastic rods, it is but a little piece in this puzzle. There is still plenty of room for future investigations. Here, some issues are emphasized.

In the context of deformation modeling for elastic rods, the correct simulation of the dynamics remains an open problem. Due to the ghost inertia problem, CORDE fails to simulate rods that are subject to large rigid body rotations, such as swung lassos.

A lot of work can also be done in the field of contact handling. Both approaches in Chapter 5 and 6 are not able to simulate sticking contacts, which is crucial for e. g. the accurate simulation of knots. From the literature, it is known that in contact mechanics, it is distinguished between impulsive contact forces and continuous contact forces [Bar89]. By investigating into efficient methods to compute both

kinds of contact forces, the contact handling could be made even more realistic. Further, in the context of contact handling for elastic rods, continuous collision techniques would prevent that rods pass through each other in the case of large time steps.

Appendix A

Robust spatial discretizations of triangle soups

In order to simulate the dynamics of a deformable object, its domain has to be discretized into disjoint volumetric elements. The governing equations of motion are then solved on the nodes of the discretized object, and the stresses acting on the nodes result from the strain rates that are computed per deformed volumetric element. Thus, the discretization or *meshing* of the domain occupied by an object has a great influence on both the efficiency and plausibility of the simulation.

Given is a domain that is bounded by a triangulated surface. The goal is to generate a volumetric data structure that covers the domain. Since most deformation models assume tetrahedral volume elements, the discussion is limited to tetrahedral meshes. In the literature, there exist many different meshing approaches. In order to answer the question about the motivation of the approach being proposed, it is necessary to consider the whole pipeline of mesh generation. If the deformation of a ‘real’ object shall be simulated on the computer, then the first step is to obtain the digitalized surface of the object. This is accomplished with e. g. a 3D laser scan of the real object. Alternatively, the surface can e. g. be constructed by employing a CAD software. These two cases have in common that the resulting triangulated surface is neither guaranteed to be ‘water-tight’, nor guaranteed to be manifold and intersection-free. Consequently, for a given surface, it is in many cases not possible to analytically compute the domain that is bounded by the surface. Even worse, it is sometimes even hard to ‘guess’ the domain that is bounded by the surface. Some examples are given in Fig. A.1.

However, many mesh generators require a manifold and intersection-free surface, e. g. the Delaunay-approaches of Alliez *et al.* [ACYD05] and Cheng *et al.* [CDRR04]. Consequently, the first step in meshing usually consists of repairing the input surface [TL94, Ju04, BPK05]. Moreover, the resulting triangle surfaces might still suffer from badly-shaped triangles, which in turn result in badly-shaped tetrahedra, and the number of triangles might be significantly higher than the desired number of tetrahedra. Thus, the consecutive stage involves to optimize and

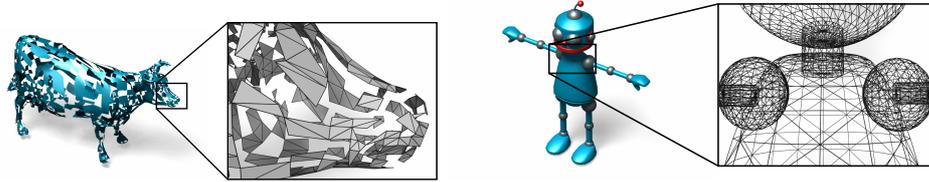


Figure A.1: Left: Surfaces obtained by laser scans can contain holes and cracks. In this example, 50% of the triangles have been removed. Most schemes will fail to tetrahedralize the volume that is bounded by the surface. Right: Models that have been constructed using a CAD software usually consist of interpenetrating subparts. Traditional mesh generators cannot handle such models properly.

simplify the triangle surface. Both steps require effortful implementations of complicated algorithms. Further, these algorithms must be carefully parameterized in order to produce valid results. In addition, these schemes do in general not operate on totally unconnected triangle soups such as depicted in Fig. A.2, which implies that such objects are fixed in a preliminary step.

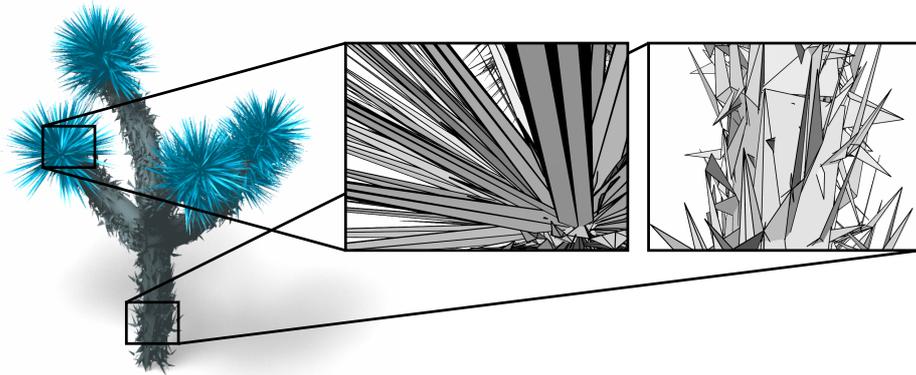


Figure A.2: This object has been modeled from unconnected triangles. For such surfaces, it is even hard to ‘guess’ the domain the bounded by the surface. An analytical computation of the volume is impossible.

In this appendix, an approach is presented that addresses the aforementioned shortcomings. It bundles the surface repair, the surface simplification, and the mesh generation into one tool with an intuitive parametrization. Instead of trying to reconstruct a closed surface which serves as input for the mesh generator, it computes an implicit representation of the *pseudo volume*, which corresponds to the volume that is intuitively occupied by the surface. The tetrahedra are then generated from the voxels that constitute the pseudo volume. A facultative post-processing stage smooths the meshes in order to make them more suitable for animation purposes. The related work in the field of tetrahedral meshing is discussed in Sec. A.1. The

algorithm is presented in Sec. A.2, and examples are given in Sec. A.3. The approach is concluded in Sec. A.4.

A.1 Related work

In order to solve the Lagrange equation of motion for a deformable object, the domain of the object must be discretized. This is usually accomplished by decomposing the domain that is occupied by the object into a disjoint set of volumetric elements. The union of those elements is then denoted as the *mesh*. The nodes of the mesh are commonly treated as the DOFs of the discretized object. In accordance, such a mesh is also denoted as a Lagrangian mesh, and the accompanying solution methods are denoted as Lagrangian approaches. This contrasts Eulerian approaches that employ a grid that is aligned to the reference frame, and where the mass of the simulated object is not explicitly carried. Eulerian approaches are widely used in fluid dynamics.

The discretization of the domain, i. e. the mesh generation, is a non-trivial task that has been extensively researched in the past years. This thesis concentrates on approaches that compute tetrahedral meshes, since tetrahedra are the most widely employed volumetric primitive. As pointed out by Shewchuk [She02], the conditioning of the stiffness matrices in a finite element method depends largely on the quality of the tetrahedra. The quality of a tetrahedron is often measured in terms of its dihedral angles, or of its radius ratio. The radius ratio is the ratio between the inscribed and the circumscribed sphere of the tetrahedron. Consequently, any meshing approach must aim at the best possible tetrahedron quality.

Tetrahedral meshing approaches differ in the boundary requirements. The strategy the mesh is computed and in Three types of boundary requirements are distinguished: *Conforming* meshes match the triangulated boundary exactly, i. e. boundary faces of the mesh correspond to the triangles of the boundary surface. Consequently, the resulting tetrahedra tend to suffer from a bad quality which is a consequence of the reduced DOFs in the node placement. *Interpolating* meshes require that the vertices of the mesh are located on the surface, and *approximating* meshes have the boundary vertices close the boundary. Approximating meshes usually allow for better tetrahedron quality, and the corresponding methods can also process non-manifold boundaries.

Three different strategies can roughly be identified. *Advancing front methods* start with the triangulated boundary and iteratively insert new interior nodes, thereby forming a front advancing towards the interior of the domain. An example of an advancing front method is the software Netgen that bases on a work by Schöberl [Sch97a]. Problems arise when two fronts have to be merged upon collision. In this case, badly shaped tetrahedra are almost unavoidable. Several schemes focus on this problem, e. g. by node removal and smoothing techniques, as described by Ito *et al.* [ISS04].

Delaunay methods start with a set of sample nodes covering the domain. They

then compute the Delaunay tetrahedralization, having the property of minimizing the maximum radius of the circumscribed spheres. In the approximation theory, this corresponds to the interpolation error inside the domain. Pioneering work in 3D-Delaunay meshing has been done by Dey *et al.* [DBS91] where meshes are generated from unconstrained sets of nodes. Their approach is extended by Shewchuk [She98] by allowing constrained input surfaces. Drawing on his work, many approaches have been proposed that aim at solving the inherent sliver problem (a sliver is a synonym for a badly-shaped tetrahedron) [CDRR04, CX04]. A scheme that combines the advantages of those publications has been proposed by Alliez *et al.* [ACYD05] that is based on a variational energy minimization during the Delaunay refinement. Since this scheme is remarkably more efficient than previous schemes, it is suitable for applications where the domain has to be remeshed frequently, as exemplified by Klingner *et al.* [KFCO06] in the context of fluid simulation. However, Alliez *et al.*'s approach does still not guarantee to completely avoid badly-shaped tetrahedra, which is a major drawback since even one single sliver can make a simulation crashing.

In contrast to Delaunay methods, *octree-based methods* start with a cubic lattice usually defined on an octree. The lattice cells are then split into tetrahedra according to some splitting rule, resulting in face-centered-cubic meshes [RO00], or body-centered-cubic meshes [MBTF03, WT08]. These strategies avoid slivers in a natural way. However, their drawbacks are two-fold: First, the resulting meshes suffer from a preferred edge direction, which can eventually distort the simulation. Second, the meshes do in general not interpolate the boundary. To address the second problem, these methods propose ways to post-process their meshes. For example, Molino *et al.* [MBTF03] simulate the mesh as a mass-spring system, thereby moving the nodes such that the exterior nodes interpolate the boundary while the interior tetrahedra preserve their good quality. A couple of schemes cut the tetrahedra that intersect the boundary. For example, Cutler *et al.* [CDM*02, CDM04] produce meshes for layered solids. To conform the tetrahedra to the external and internal boundaries, they are cut, and the resulting slivers are removed by a series of edge swaps and collapses. Similar in spirit is the approach of Müller and Teschner [MT03] that focuses on coarse meshes that are suitable for interactive simulations. Again, a post-processing of the mesh tries to maximize the quality of the tetrahedra. In contrast, Wojtan *et al.* argue that the mesh boundary does not need to be conforming if a high-resolution surface is embedded into the tetrahedral mesh. In their case, the contact handling is carried out on the surface, while the tetrahedral mesh is only employed to compute the deformations [WT08]. The research on octree-based methods cumulates with a recent publication of Labelle and Shewchuk [LS07]. They start with a signed distance field defining the domain. To cut the lattice cells along the iso-surface, they propose a 'marching cubes'-like scheme: The cells are tetrahedralized according to a small set of rules, based on the signs of the nodes of the cell. Their major contribution over all previous approaches is that they guarantee minimum dihedral angles bounds, thereby completely avoiding slivers.

Despite of their diversity in the employed strategies, the discussed methods share one similarity: They mainly aim at generating dense meshes from well-formed, manifold boundaries. The resulting tetrahedron meshes are then employed by numerical solvers for partial differential equations that govern e. g. fluid flow or mechanical deformation. Here, the accuracy requirements dominate the efficiency, which is also exhibited in the size of the generated meshes, notably in the order of 10K to 200K in Alliez *et al.*'s approach [ACYD05], or 100K to 560K in Cutler *et al.*'s approach [CDM*02]. Moreover, most schemes focus on graded meshes where the size of the tetrahedra differ in size. Instead, for the interactive simulation of deformable objects, an approach that approximates a domain with only a few hundred well-shaped tetrahedra is favorable. Moreover, the surfaces that are obtained from 3D laser scans or constructed using CAD software are in general not manifold, and consequently any approach that considers the original surface will fail. This is alleviated by generating the mesh from either a distance field that covers the domain [LS07], or by a implicit surface that is computed from the input non-manifold surface [SOS05]. The approach that is presented in this appendix generates as well a distance field. In contrast to the other approaches, it additionally computes the pseudo volume, which conforms to the space that is intuitively covered by the surface. This enables a more concise discretization of the domain.

A.2 Approach

In this section, the approach to generate tetrahedral meshes from arbitrary triangle soups is presented. Fig. A.3 illustrates the process.

Input surface As input, the approach assumes a set of triangles, commonly known as a *triangle soup* (Fig. A.3 a). No assumptions are made about the orientation or the connectivity of the triangles.

Volume representation As the first step, the approach computes a distance field from the surface. The voxels with negative sign constitute the pseudo volume that is occupied by the surface (Fig. A.3 b).

Mesh generation As the second step, the distance field is overlaid with a rectangular lattice. The cells are tetrahedralized, and the tetrahedra that are outside the pseudo volume are removed (Fig. A.3 c).

Mesh optimization As the third step, the sharp features are removed from the resulting mesh, making it more suitable for the animation (Fig. A.3 d).

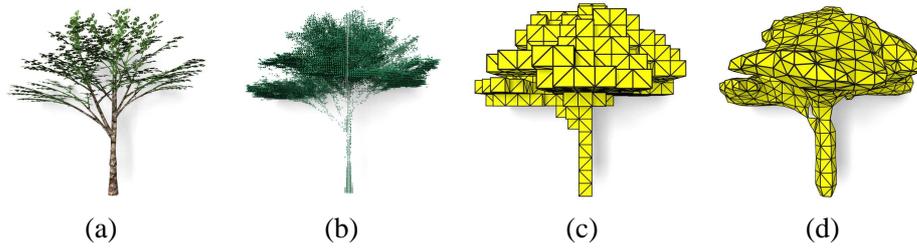


Figure A.3: Generating a tetrahedral mesh from a triangle soup (a). First, a distance field is computed. A novel method generates the signs of the distance field values. Negative values represent the space that is intuitively occupied by the surface, i. e. the pseudo volume (b). A tetrahedral lattice is laid onto the pseudo volume (c) and a smoothing filter is applied to obtain a mesh (d) that is appropriate for interactive simulations.

Volume representation

Computing a tetrahedral mesh from a surface corresponds to computing a domain Ω in \mathbb{R}^3 from a boundary $\partial\Omega$. The domain is well-defined if $\partial\Omega$ is a closed and orientable manifold. It has to be underlined that both criteria are necessary: Although a Klein-bottle is a closed manifold, it cannot be oriented and has an infinite volume.

Mesh generators commonly assume that the above criteria hold for the input surface. In contrast, the presented approach addresses surfaces that do not meet these criteria. Since the domain might be undefined, approaches that analytically determine the volume cannot be considered. Instead, an approximation of the space that is occupied by the surface has to be computed, which is here termed pseudo volume. In order to represent the pseudo volume, a signed distance field is employed, where the signs of the distance field indicate whether the voxel is inside or outside the boundary. The boundary is implicitly given by the zero-isocontour of the distance field. The set of the voxels with negative signs constitutes the pseudo volume of the object.

Distance field generation In order to generate the distance field, the approach of Felzenszwalb and Huttenlocher [FH04] is employed. This approach computes the distance transform of a sampled function. Thus, as a preliminary step, the individual triangles constituting the input surface have to be sampled into the voxel grid. If a triangle intersects a voxel, then the distance value of the voxel is 0, otherwise it is infinity. In order to determine whether the triangle intersects the voxel, an approximative test is done by considering the minimum distance between the triangle and the center of the voxel. The expected complexity of the approach is $\mathcal{O}(ST)$, where T is the number of triangles, and S is the expected number of voxels that are intersected by a triangle. Thus, S depends on the resolution of the distance field. The required resolution of the voxel grid is independent of the input surface,

but it must be significantly higher than the desired resolution of the tetrahedron mesh. For meshes with a resolution in the order of 500 elements, distance fields with a resolution of 100^3 voxels provide sufficient accuracy.

Sign generation In the preceding step, the distance transform has been applied to the sampled surface triangles. The crucial step is now to compute the pseudo volume, i. e. to classify the voxels in whether they contribute to the pseudo volume or not.

This classification is done by employing an adaption of the parity count scheme proposed by Nooruddin and Turk. They cast rays in a given direction through the surface and count the number surface intersections. Voxels with an odd number of intersections are then classified as interior. Since holes can lead to a misclassification of a span, rays are shot from 13 different directions, and the majority vote decides on the sign of a voxel [NT03]. This method works well for surfaces that contain a small number of holes, but are closed elsewhere. However, it cannot be applied to unconstrained surfaces as depicted in Fig. A.2 since the parity only changes if a ray actually interferes with the surface.

In the following discussion, a parity change is denoted as a *volume event*. A parity change from even to odd is denoted as an *out-in* volume event, and the parity change from odd to even as an *in-out* volume event. Then, the normalized distance $|d_v|$ is the distance value d_v from the center of the voxel v to the surface, divided by the maximum distance within the voxel grid. The normalization makes that all distance values are between 0 and 1, with 0 if a voxel is intersected by a surface triangle, and close to 1 if the voxel is far away from the input surface.

The normalized distance $|d_v|$ can now be considered in the decision whether a volume event is invoked, notably if $d_v < d_{\min}$. Here, $d_{\min} \in [0, 1]$ is a global threshold value that governs the shape of the resulting pseudo volume. A subsequent volume event is only reported if $|d_v|$ has been larger than d_{\min} in the meanwhile. This criterion makes that the convex hull of the pseudo volume is not larger than the convex hull of the sampled surface. Experiments indicate that $d_{\min} = 0.2$ provides good results.

Together with the fact that the rays are shot from 13 different directions, the resulting pseudo volume becomes plausible. The consideration of the normalized distance values results in superior robustness, compared to the approach of Nooruddin and Turk [NT03]. The complexity of the sign generation approach is linear in the number of voxels.

The resulting pseudo volume does not conform to the analytical volume since such a volume does not exist for unconstrained surfaces. Further, the pseudo volume is not unique for a given surface but depends on the choice of the threshold value d_{\min} . If $d_{\min} = 0$, then the pseudo volume corresponds to [NT03]. For $d_{\min} > 0$, the effect is similar to applying a morphological closure operator to the negative signed voxels produced by [NT03]. In contrast to [NT03], all kinds of degeneracies are handled in a unified scheme. Examples of the computed pseudo volumes

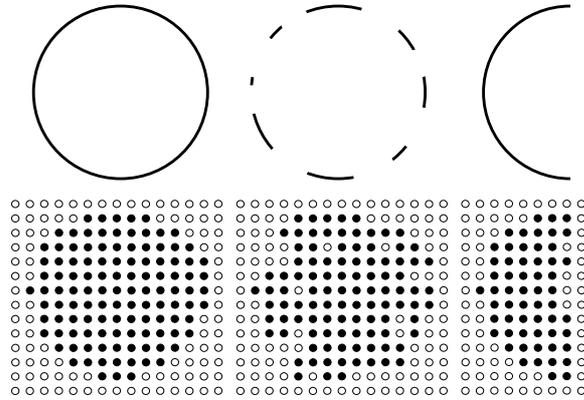


Figure A.4: Generation of the signs of a 2D distance field for a damaged circle. Negative values represent the pseudo volume bounded by the surface. Left: Pseudo volume of the unmodified circle. In this case, the pseudo volume conforms to the actual voxelized volume bounded by the circle. Middle: Pseudo volume of the circle with 50% of its segments missing. Right: Pseudo volume of one half of the circle.

are given in Fig. A.4.

Mesh generation

In the previous step, the pseudo volume has been computed, which is defined by the set of voxels with negative sign. Although the pseudo volume does not conform to the analytical volume bounded by the surface, it nevertheless provides a description of what is intuitively thought as the space occupied by the object. Moreover, it now enables the meshing of the surface.

To mesh the pseudo volume, a structured method based on an axis-aligned grid [Blo94, CDM*02] is employed. First, a uniform lattice is laid onto the distance field. The resolution of the lattice is user-defined. Each cubical cell is split into five tetrahedra. To ensure that neighboring cells match each other, the orientations of the tetrahedra alternate [CDM*02].

Second, those elements are discarded that lie outside the volume. In contrast to [MT03, VTG97], neither the input surface nor the zero isocontour are considered, since the input surface does not necessarily provide any ‘useful’ information about the volume. As a consequence, it is not possible to test an element against a surface normal to decide whether it is inside or outside the volume. Instead, the pseudo volume density $\rho(V)$ within an arbitrary tetrahedron element V is computed as

$$\rho(V) = \frac{m(V)}{M(V)}$$

where $m(V)$ is the number of voxels with negative sign within the element V , and $M(V)$ is the total number of voxels within the element V . $\rho(V)$ gives a measure

of the fill-degree of an element. By discarding all elements with $\rho(V) < \rho_{\min}$, only those elements are preserved that cover the pseudo volume. ρ_{\min} is a user-defined value that governs the shape of the final tetrahedral mesh.

Since the element discarding scheme does neither consider the orientation of the surface nor its zero isocontour, it is robust for unoriented and scattered input surfaces. Moreover, the user can control the shape of the resulting tetrahedral mesh. By adjusting the value ρ_{\min} , an intuitive way is provided to control how much space around a scattered surface is covered by the tetrahedral mesh. In the experiments, $\rho_{\min} = 0.5$. If ρ_{\min} is decreased, more tetrahedrons are added at the boundary of the mesh. Transformations of the lattice provide additional degrees of freedom. However, so far the user interface is restricted to translations.

Mesh optimization

The tetrahedra that have been generated in the previous step are of excellent quality. However, the discarding of tetrahedra that lie outside the volume produces sharp creases and corners and a dynamic simulation of such meshes with collision handling would result in a contra-intuitive behavior. Thus, a smoothing filter is applied in a postprocessing step to produce well-tempered mesh borders.

Tetrahedral mesh generators commonly postprocess the mesh boundary to improve the mesh quality and to conform the boundary to the input surface. In [VTG97], springs connect the tetrahedral mesh and the boundary surface to move mesh vertices towards the object surface, while [ACYD05] performs an iterative optimization procedure to align the vertices directly to the boundary surface. Other approaches use the isocontours of the distance field instead: [CDM*02] subdivides the elements that lie onto the considered isocontour, and [MBTF03] describes a FE or mass-spring simulation to conform the vertices to the zero isocontour. However, these approaches assume that either the boundary surface or the isocontours of the distance field are well-shaped, which is not the case for unconstrained and scattered surfaces.

Instead, a global mesh improvement procedure is performed that is independent of the input surface and its distance field. In order to get rid of sharp creases and corners, a scale-dependent umbrella operator that is a discrete approximation of a Laplacian filter [DMSB99] is applied. However, since pure diffusion induces shrinkage, the volume of the tetrahedral mesh has to be restored after each smoothing step. A slight adaption of the procedure described in [DMSB99] is employed, because scaling back the whole mesh to its original volume conforms to a morphological closure operation, which is undesirable for concave objects. Instead, a local volume difference is computed. Then, the local mesh boundary is moved, depending on the sign of this difference. This prevents the closure of concave regions. Refer to Fig. A.5 for an example.

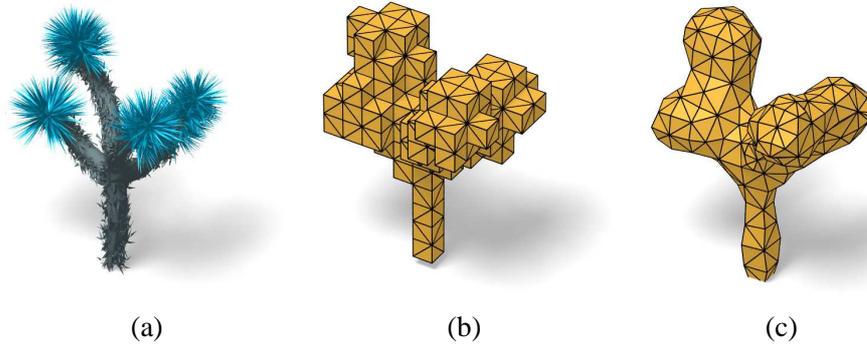


Figure A.5: Surface smoothing. (a) The input surface. (b) The tetrahedral lattice. (c) A smoothing filter is applied to the lattice in order to remove creases and corners. Basically, the volume of the mesh is preserved.

A.3 Results

In this section, an evaluation of the produced meshes is presented. The focus is on three aspects, namely the scalability of the approach, the ability to generate meshes from damaged and unconnected triangle soups, and the performance of the deformable object simulation, where the deformations are computed on the generated meshes. All tests have been performed on an Intel Xeon PC, 3.8 GHz.

Scalability

The resolution of the tetrahedron lattice provides an intuitive degree-of-freedom that governs the size of the resulting meshes. To illustrate this, a torus surface is meshed. The input surface is a closed and oriented manifold. The corresponding voxel grid has a resolution of $150 \times 150 \times 37$ voxels. Fig. A.6 illustrates that the approach can generate meshes at different resolutions. The left mesh consists of 139 elements. Due to its coarseness, it is well-suited for interactive simulations. The right mesh consists of 8K elements. The cutaway view exhibits the regular structure of the interior elements.

Applicability

The main strength of the proposed approach is its ability to generate meshes from arbitrary input surfaces. To illustrate this broad applicability, meshes have been generated from both intersecting and unconnected surfaces.

Intersecting surfaces The model depicted in Fig. A.7 is composed from interpenetrating basic primitives, with a total of 16K triangles. While a human observer can intuitively identify the volume, algorithms might have problems to consider the union of two intersecting subparts as interior. A traditional approach

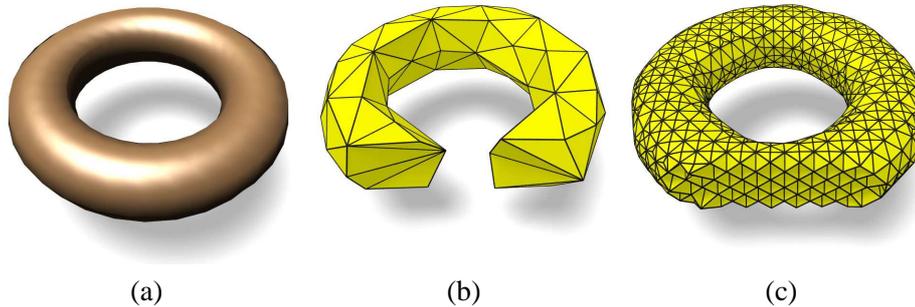


Figure A.6: Meshing the torus. (a) The input surface. The proposed approach can generate tetrahedral meshes of varying resolutions. The mesh (b) consists of 139 elements and the mesh (c) consists of 8K elements.

such as [ACYD05] would fail to generate the mesh for such a surface. In contrast, the proposed approach can generate a plausible tetrahedron mesh without pre-processing the input.

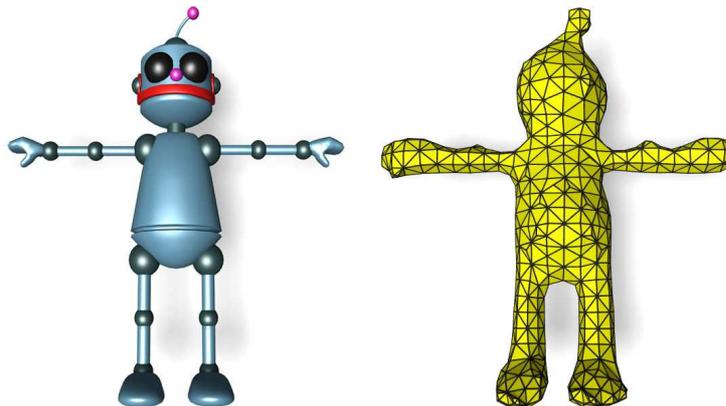


Figure A.7: The input model is composed of interpenetrating subparts. However, the approach still computes a plausible tetrahedral mesh for this object. The mesh consists of 2K elements.

Damaged surfaces To illustrate that the approach can handle even badly damaged models, 50% of the triangles have been removed from a surface as shown in Fig. A.8. Still, the generated mesh contains no holes or artifacts on the surface. Further, the elements of the generated tetrahedral mesh do not significantly vary in size, which is an important criterion for the stability of many deformation models [THMG04, MG04].

Triangle soups There exist surfaces that are composed of a set of unconnected triangles and many existing techniques are not able to tetrahedralize such models.

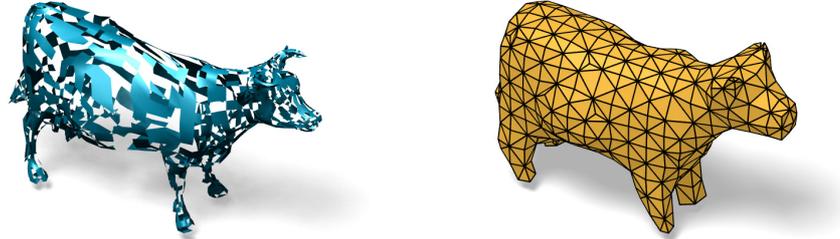


Figure A.8: The approach can be used to build tetrahedral meshes from damaged surfaces. 50% of the faces have been removed from the surface (left). However, a plausible tetrahedral mesh can still be produced for this surface (right). The mesh consists of 1158 elements.

In contrast, the proposed approach generates the tetrahedron mesh from the pseudo volume, which conforms to the space that is intuitively occupied by the surface. Fig. A.5 shows the tetrahedral mesh generated for a palm model that consists of 520 tetrahedra.

Performance The generation of a well-shaped mesh from an arbitrary input surface requires only a couple of minutes. To compute the distance field, the approach of Felzenszwalb and Huttenlocher [FH04] is employed. For example, the rasterization of the cow-model (Fig. A.8) consisting of 2.8K triangles into a grid of $150 \times 150 \times 150$ voxels takes 0.1s. Computing the distance transform takes 1.1s, and computing the pseudo volume (i. e. assigning the signs) takes 28s. The generation and the optimization of the mesh is governed by the user and is done at interactive rates.

Deformation Since the approach can produce coarse and nevertheless plausible tetrahedral meshes, they are well-suited for interactive simulations and animations. To illustrate this, the resulting meshes are employed to compute the dynamics of deformable objects, where the mass-spring deformation model of Teschner *et al.* [THMG04] is employed.

Fig. A.9 shows an animation of four deformable robot models falling onto a deformable tree model. The total number of elements is 3K. The simulation runs at about 20 frames per second including visualization and collision handling. A massive scenario is depicted in the bottom row of Fig. A.9 where 125 deformable robots are falling onto a deformable tree model. The total number of elements is 58K and the simulation runs at 0.3 frames per second including the rendering of two million surface triangles.

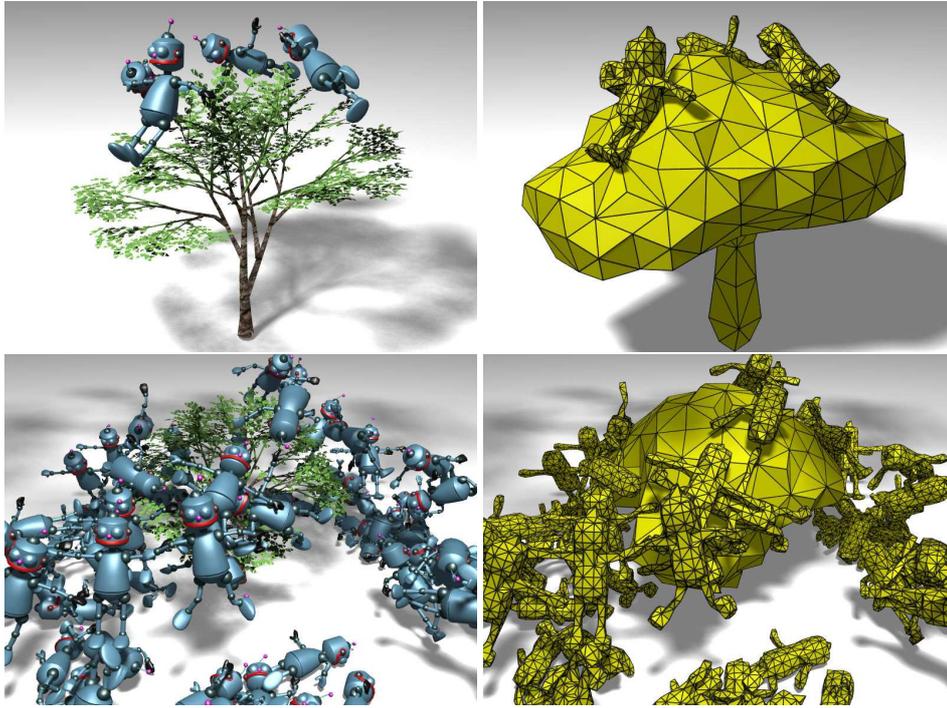


Figure A.9: Simulation of deformable objects. The meshes have been generated using the proposed approach. The scenario in the top row can be simulated at 20 frames per second using [THMG04]. A massive scenario, consisting of 125 objects and a total of 58K tetrahedra is shown in the bottom row.

A.4 Discussion

In this appendix, an approach has been presented that generates tetrahedral meshes from arbitrary input surfaces. Motivated by the observation that existing schemes impose strong constraints on the input surfaces, and that existing schemes often produce high-resolution meshes which are of little practical interest in the interactive simulation, the goal has been to provide a simple and robust scheme for the generation of coarse meshes.

Instead of trying to repair the input surface (which is hardly possible for many examples shown in this appendix), the approach performs a distance transform on the sampled input surface. Then, the pseudo volume is computed, which conforms to the volume that is intuitively occupied by the surface. A tetrahedron mesh is then generated which covers the pseudo volume. As a post-processing step, the mesh is optimized by removing the sharp creases and corners, thereby making the mesh looking smoother and making it more suitable for the animation. The resulting meshes are widely employed in a framework for deformable objects, and both mass-spring and FE deformation models perform well with the generated meshes.

The approach falls into the category of *approximating meshing schemes*. As

a consequence, the boundary nodes of the resulting meshes are not guaranteed to lie on the surface. This fact is underlined by the mesh post-processing which does not consider the input surface during node displacement. Instead, the optimization is entirely local, trying to smooth out creases and corners without significantly changing the volume of the mesh. The mesh optimization is a user-assisted process. Still, the impossibility of making qualitative statements about the approximation of the mesh is one of the main drawbacks of the approach. It is also impossible to preserve sharp creases wherever it is favorable, since a localized mesh optimization is not supported.

Another problem is the generation of sliver tetrahedra. The unoptimized meshes have an excellent quality since the tetrahedra all have the same shape. However, during the mesh smoothing, the tetrahedra will shrink, thereby eventually degenerate to slivers. Consequently, the quality of the mesh degrades with the smoothing intensity. However, experiments indicate that moderately smoothed meshes have still a better quality than meshes that are produced with other mesh generators, e. g. [MT03].

Future work Further investigations are obvious in order to give certain guarantees on the approximation quality of the resulting meshes. This problem has recently been addressed in a student project (unpublished), where the scheme has been combined with the iso-surface stuffing approach of Labelle and Shewchuk [LS07]. The resulting meshes do then interpolate the iso-surface of the pseudo volume. In doing so, the mesh optimization stage is avoided and the elements have a better quality.

Appendix B

An output-sensitive collision detection approach

One of the most expensive tasks in the dynamic simulation of deformable objects is the detection of the object interferences. Still, it is mandatory in order to compute the contact forces. The collision detection is mainly a geometric problem, and in order to quickly determine the interferences, model partition methods are widely used. Those methods construct a hierarchical data structure of bounding volumes that enclose the object. However, in the context of deformable objects, the bounding volume hierarchies must be refitted frequently in order to continuously encapsulate the deforming geometries. To accomplish this, it is in general necessary to visit each deformed primitive, thereby making the complexity of the interference test linear in the number of primitives.

In this appendix, a method is investigated that reduces the complexity of the intersection test of two bounding sphere hierarchies from $\mathcal{O}(N)$ to $\mathcal{O}(C)$, where N is the total number of primitives, and C is the number of potentially colliding primitives. Due to spatial coherence, C is usually much smaller than N . This reduction is achieved by avoiding looking at each primitive during the refit of the bounding volumes. Instead, it is assumed that the deformation of the object can be described by a single transformation matrix. Such deformation models are commonly denoted as *geometric deformation models*. Then, the bounding spheres can be updated by employing the transformation matrix, thereby making the update of a bounding sphere independent from the number of enclosed primitives.

Consequently, during the recursive interference test, a bounding sphere at a higher level in the hierarchy can be updated in constant time. Thus, if two bounding spheres at this level do intersect, it is known implicitly that the enclosed primitives do not intersect neither. The hierarchies are only traversed down to the leaves for those primitives that do collide or that are in close proximity. Since C leaves have to be tested, about C bounding spheres have to be updated, and since these updates are constant, the complexity of the intersection test is in $\mathcal{O}(C)$. A similar result has been shown by James and Pai in the context of interference queries between

reduced deformable models [JP04].

Since the update rules for the bounding spheres depend on the geometric deformation model, the necessary preliminary theory is shortly summarized in Sec. B.2. The update algorithm is detailed in Sec. B.3. The algorithm is evaluated in Sec. B.4, and a discussion on the limitations and possible future directions is given in Sec. B.5.

B.1 Related work

Collision detection is a well-investigated research area, and many different strategies exist to solve this problem. A thorough introduction to general collision detection is given in e. g. the books of Van den Bergen [Van04]. Further, the state-of-the-art report of Teschner *et al.* addresses collision detection techniques for deformable objects [TKH*05].

The collision detection strategies can broadly be separated into two classes, notably the *model partitioning schemes*, and the *space partitioning schemes*. Model partitioning schemes partition the geometry of the object into spatially coherent subsets [LLC*07] and compute a bounding volume for each subset. Usually, the spatial coherence is captured in a hierarchical data structure. During the interference test, the bounding volumes constituting the nodes in the hierarchies can be recursively tested against each other. If two bounding volumes do not overlap, the enclosed geometries can be quickly rejected.

In contrast, space partitioning schemes partition the bounded space \mathbb{R}^3 into cells. Then, the objects are rasterized into the grid. To test for collision, it then suffices for one primitive to find out the cells it belongs to, and to test all other primitives that have been rasterized into these cells. Space partition schemes are particularly popular in the field of molecular dynamics [Lev66, Tur90] and can be combined with hashing techniques to save memory and improve the efficiency [THM*03].

There exist a large variety of approaches that do not fall into one of the above categories, such as image-space techniques [MOK95, HTG04], distance field approaches [FL01], and stochastic methods [KZ03]. However, since the proposed approach falls into the category of model partitioning schemes, the focus is on this category. The discussion is started with a general review of the model partition schemes, before venturing into output-sensitive schemes.

Model partitioning schemes In the literature, many different types of bounding volumes have been proposed. The choice of bounding volume hierarchy is a trade-off between the approximation quality of the bounding volume and the efficiency of the intersection test. Spheres [Qui94, Hub95] can be quickly tested for overlap, but do not approximate the geometry very well. The same holds for axis-aligned bounding boxes (AABBs) [Van97]. Oriented bounding boxes (OBBs) [GLM96]

and k-DOPs [KHM*96] provide a tighter fitting, but the interference test is more expensive.

In the context of deformable geometries, the bounding volumes have to be refitted frequently in order to keep enclosing the primitives. The straight-forward way to accomplish this is by starting from the root, update the bounding volumes of the leaves, and then proceed up the tree and refit the inner nodes recursively. Since each primitive must be considered at least once, $\mathcal{O}(N)$ operations are necessary. Van den Bergen has reported that refitting an AABB hierarchy is about ten times faster than rebuilding the complete hierarchy [Van97], since the AABBs at the higher levels enclose the AABBs at lower levels. In the terminology of Guibas *et al.* [GNRZ02], such hierarchies are denoted as *wrapped hierarchies*, which contrasts *layered hierarchies* where parent bounding volumes do not necessarily enclose their child volumes. Later, Larsson and Akenine-Möller have proposed a hybrid method to update the tree, by first updating the leaves, then proceeding top-down [LAM01]. They have reported a significant speed-up compared to Van den Bergen, although they do not lower the $\mathcal{O}(N)$ complexity. Later, Larsson and Akenine-Möller have proposed a further optimized update approach that exploits temporal coherence [LAM05].

Sub-linear schemes In general, it is not possible to lower the $\mathcal{O}(N)$ complexity of the hierarchy update, since each primitive must be considered at least one time. This is because the deformation is given in terms of the position of the primitives in the object frame. However, one class of collision detection schemes exploits information on the deformation in order to update the hierarchy. This is enabled since the underlying deformation models limit the deformations to certain modii. For reduced deformation models, the position of a primitive is a convex combination of a set of deformation fields. In this context, James and Pai have shown that a sphere enclosing those primitives can be updated based on the deformation fields, thereby avoiding to visit each enclosed primitive [JP04]. As a consequent, they have obtained a sub-linear complexity of the hierarchy update.

Similar in spirit is the approach earlier proposed by Larsson and Akenine-Möller that considers morphing geometries. Since the geometry is always a blend between the start- and the target-configuration, the AABB hierarchy can be blended likewise, thereby avoiding the traversal of the leaves [LAM03]. Kavan and Zára have addressed skinned skeletally deformable objects, where the positions of the vertices of the skin are obtained as a convex combination of the adjacent joint transforms [KZ05]. Later, Steinemann *et al.* have extended Kazan and Zára's work in order to efficiently update the skin of a point cloud [SOG07].

The consideration of adaptive simulation opens a further way to enhance the collision detection, e. g. by employing the octree-representation in a hierarchical FE simulation [DMG05]. Alternatively, the multigrid methodology allows for an efficient refit of the bounding volumes, as illustrated by Otaduy *et al.* They employ the correspondence across the levels in the multigrid hierarchy [OGRG07].

In this thesis, an approach is proposed that refits a bounding sphere hierarchy enclosing a geometrically deformable object [MHTG05]. A sub-linear complexity is achieved by employing information on the geometric deformation.

B.2 Preliminary theory

In order to update a bounding sphere, information on the current deformation of the enclosed primitives is employed. Therefore, it is assumed that the underlying deformation model is geometric. That means, the deformation is governed by a geometric transformation of the undeformed geometry. In this appendix, the meshless deformation model of Müller *et al.* [MHTG05] is employed, but adaptations of the presented approach could be used along with any other geometric deformation model such as [RJ07, SOG08].

The motivation for the geometric deformation model as proposed in [MHTG05] comes from the observation that implicit integration schemes are stable, but require the solution of a linear system, which can be expensive. Explicit integration schemes are more efficient but prone to instabilities, which comes from the fact that they integrate blindly into the future and therefore tend to overshoot. What would be favorable was an integration scheme who already ‘knows’ the goal position $\mathbf{g}_i = \mathbf{x}_i(t+h)$ in the subsequent time-step $t+h$. In this case, the future velocity $\mathbf{v}_i(t+h)$ could be updated from the difference of the current position and the goal-position, divided by the time step. What sounds like an algorithmic discrepancy constitutes the idea of the geometric deformation model: Given is a solid whose volume is discretized into a point cloud $\mathbf{p}_i, i = 1..N$, where the \mathbf{p}_i are the coordinates of the points in the local frame. If external forces act on the points, then the geometric configuration of the point cloud is deformed. Still, the undeformed reference configuration \mathbf{p}_i^0 is known. To compute the elastic response, the reference configuration \mathbf{p}_i^0 is registered with the deformed configuration \mathbf{p}_i in order to determine the transformation matrix \mathbf{A} that, if applied to the reference configuration, matches the deformed configuration optimally in the least squares sense. The goal positions then result from the transformed reference configuration, that is blended with the rigidly transformed reference configuration, $\mathbf{g}_i = (\beta\mathbf{A} + (1-\beta)\mathbf{R})\mathbf{p}_i^0$. Here, \mathbf{R} is the optimal rigid transformation, and the blending governs the elastic response of the point cloud. To see why, let $\beta = 0$: In this case, the goal positions conform to the reference positions in the object frame, and the deformed points are immediately accelerated onto their reference positions. For $\beta > 0$, the points are instead accelerated onto their reference positions, transformed by $\beta\mathbf{A}$. Fig. B.1 illustrates the case of $\beta = 1$.

Consequently, the transformation \mathbf{A} governs the elastic response of the solid. In [MHTG05], Müller *et al.* propose two deformation modii: For *linear deformations*, \mathbf{A} is a linear 3×3 deformation matrix that minimizes $\sum_i \|\mathbf{A}\mathbf{p}_i^0 - \mathbf{p}_i\|_2$. Linear transformations are able to model shear and stretch deformations.

In contrast, *quadratic deformations* extend the range of deformations by twist-

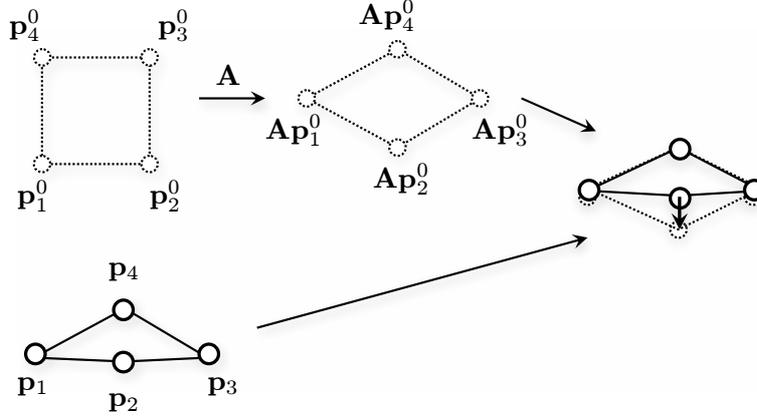


Figure B.1: The undeformed geometry $\{\mathbf{p}_i^0\}$ is registered with the deformed geometry $\{\mathbf{p}_i\}$. A transformation matrix \mathbf{A} is computed that minimizes $\sum_i \|\mathbf{A}\mathbf{p}_i^0 - \mathbf{p}_i\|_2$.

ing and bending modes. To accomplish this, the vectors

$$\tilde{\mathbf{p}}_i = (p_x, p_y, p_z, p_x^2, p_y^2, p_z^2, p_x p_y, p_y p_z, p_z p_x)$$

extend the vectors \mathbf{p}_i by the quadratic and mixed modes. Then, matrices \mathbf{A}_L , \mathbf{A}_Q and \mathbf{A}_M are computed that minimize $\sum_i \|[\mathbf{A}_L \ \mathbf{A}_Q \ \mathbf{A}_M] \tilde{\mathbf{p}}_i^0 - \mathbf{p}_i\|_2$. Here, $\mathbf{A}_L \in \mathbb{R}^{3 \times 3}$ represents the linear terms, $\mathbf{A}_Q \in \mathbb{R}^{3 \times 3}$ represents the pure quadratic terms, and $\mathbf{A}_M \in \mathbb{R}^{3 \times 3}$ represents the mixed terms.

Since the whole point cloud is transformed with the same deformation matrix, a local deflection immediately results in a global deformation. To increase the plausibility of the deformation and to enable localized deformations, the original point cloud can be divided into M overlapping clusters j . In this case, the registration is performed in the local frame of each cluster, which results in M deformation matrices \mathbf{A}_j . The clustering of the solid and especially the overlap of the clusters provide further DOFs of the deformation model.

This approach shares similarities with the previously discussed reduced deformation models. However, the displacement fields are not explicitly stored but exclusively computed from the linear or quadratic transformation matrices. Another important difference is that for reduced deformation models, the deformation is exclusively governed by the displacement fields. In contrast, for the geometric deformation model, the points are basically free to move wherever they want. The elastic response serves as an additional impulse that works against the deformation. This detail will turn out to be important in the context of the collision handling of such geometric deformable solids, as described in the next section. Details on the registration process and on the unconditionally stable numerical integration scheme that comes with the deformation model are found in [MHTG05].

B.3 Approach

The approach employs bounding sphere hierarchies to accelerate the collision detection among the deformable objects. The bounding sphere hierarchy is pre-computed for each object in its initial undeformed state. If the object deforms, the hierarchy is updated in order to enclose the deformed geometry. To update the bounding spheres, the information of the geometric transformation is employed (see Fig. B.2). The update procedure is combined with the recursive interference query in the sense that only the successors of those spheres are updated that have been colliding. Therefore, non-colliding branches can be culled at an early stage, without being refitted. The approach is similar in spirit to the BD-tree scheme of James and Pai that performs an output-sensitive interference query for reduced deformation models [JP04].

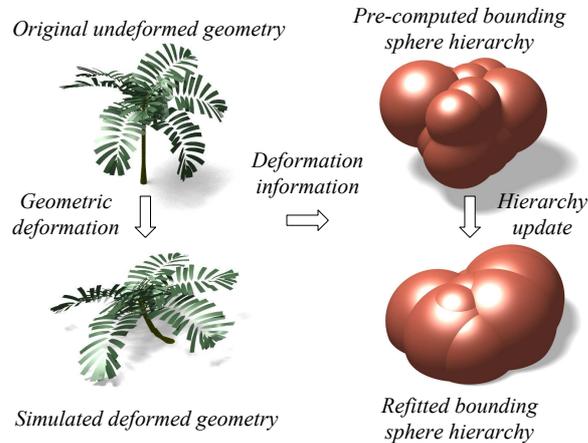


Figure B.2: Algorithm overview. A bounding sphere hierarchy is pre-computed for an undeformed object. If the object deforms, the hierarchy is updated employing information on the geometric deformation model.

A level-of-detail collision response scheme computes contact forces at arbitrary levels of the bounding volume hierarchy and propagates the result to the enclosed object primitives. Alternatively, accurate contact forces are computed for object primitives if connectivity information is available.

Hierarchy construction

The bounding sphere hierarchy is pre-computed for an object in its undeformed state by employing the technique of [OD99]. This guarantees that all successors of a bounding sphere contain the same number of points. Thus, the resulting tree is well-balanced. Following [GNRZ02], a wrapped hierarchy is used, i. e. a parent sphere does not enclose all child spheres, but only the associated geometry. This yields smaller spheres and improves the performance of the collision query. The

minimum enclosing sphere of a point set is computed using [FG03].

Hierarchy update

If an object deforms, its bounding sphere hierarchy has to be updated. Instead of using information on all points enclosed by a sphere, the transformation matrices provided by the geometric deformation model are employed to refit the spheres.

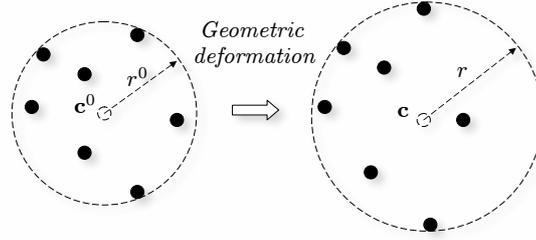


Figure B.3: Update of a bounding sphere hierarchy upon deformation. Left: In the undeformed state, a sphere given by \mathbf{c}^0 and r^0 encloses a point set \mathfrak{P}^0 . Right: After a geometric deformation of the point set, a new center \mathbf{c} and a new radius r have to be computed such that all deformed points are enclosed by the sphere.

Let \mathbf{x}_{cm}^0 be the center of mass of the undeformed object and $\mathfrak{P}^0 = \{\mathbf{x}_i^0 | i = 1, \dots, n\}$ a subset of the undeformed object. Further, $\mathbf{p}_i^0 = \mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0$ are the relative positions of the subset with respect to the center to mass. Further, \mathbf{c}^0 and r^0 are the center and radius that constitute an arbitrary sphere within the hierarchy. The sphere is constrained to enclose all points of \mathfrak{P}^0 , i. e.

$$\|\mathbf{x}_i^0 - \mathbf{c}^0\| \leq r^0 \quad \forall \mathbf{x}_i^0 \in \mathfrak{P}^0. \quad (\text{B.1})$$

Fig. B.3 illustrates the setting.

Update rule for linear deformations Now the deformed object is considered. The center of mass of the deformed object is denoted as \mathbf{x}_{cm} , and the deformed point positions \mathbf{x}_i , with the relative positions $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$. Further, $\mathbf{A}\mathbf{p}_i^0 = \mathbf{A}(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0)$ be the transformed initial positions that result from the shape matching procedure. The deviation between the transformed initial positions and the deformed positions is

$$\mathbf{d}_i := \mathbf{A}\mathbf{p}_i^0 - \mathbf{p}_i. \quad (\text{B.2})$$

For a sphere update, the 2-norm of a square matrix \mathbf{A} is considered that is defined as the square root of the absolute value of the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. In the implementation, a polar decomposition is performed on $\mathbf{A}^T\mathbf{A}$ in order to find the largest eigenvalue. Geometrically, the 2-norm indicates how much the length of a vector can change under the transformation \mathbf{A} .

Claim If the center of the bounding sphere is updated with

$$\mathbf{c} \leftarrow \mathbf{A}(\mathbf{c}^0 - \mathbf{x}_{\text{cm}}^0) + \mathbf{x}_{\text{cm}} \quad (\text{B.3})$$

and the radius with

$$r \leftarrow \|\mathbf{A}\|_2 r^0 + d \quad (\text{B.4})$$

then the updated sphere encloses all transformed points, i. e.

$$\|\mathbf{x}_i - \mathbf{c}\| \leq r \quad \forall i = 1, \dots, n \quad (\text{B.5})$$

Assuming temporal coherence, then the value $d := \max_j \|\mathbf{d}_j\|_2$ can be computed from all points \mathbf{p}_j that have been colliding in the previous simulation step. For non-colliding points \mathbf{p}_i , \mathbf{d}_i is negligible since the positions of these points coincide with the computed goal positions, and the bounding spheres are proven to enclose the latter.

Proof Applying the definitions of the transformed initial positions and \mathbf{p}_i to (B.2) results in

$$\begin{aligned} \mathbf{A}(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0) &= \mathbf{x}_i - \mathbf{x}_{\text{cm}} + \mathbf{d}_i \\ \iff \mathbf{x}_i &= \mathbf{A}\mathbf{x}_i^0 - \mathbf{A}\mathbf{x}_{\text{cm}}^0 + \mathbf{x}_{\text{cm}} - \mathbf{d}_i \end{aligned} \quad (\text{B.6})$$

Using (B.3) and (B.6) to describe the distance between a new position and the new center,

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{c}\|_2 &= \|\mathbf{A}\mathbf{x}_i^0 - \mathbf{A}\mathbf{x}_{\text{cm}}^0 + \mathbf{x}_{\text{cm}} - \mathbf{d}_i - (\mathbf{A}(\mathbf{c}^0 - \mathbf{x}_{\text{cm}}^0) + \mathbf{x}_{\text{cm}})\|_2 \\ &= \|\mathbf{A}(\mathbf{x}_i^0 - \mathbf{c}^0) - \mathbf{d}_i\|_2 \end{aligned} \quad (\text{B.7})$$

Applying the triangle inequality and the submultiplicative property of the matrix norm $\|\cdot\|_2$,

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}_i^0 - \mathbf{c}^0\|_2 + \|\mathbf{d}_i\|_2 \quad (\text{B.8})$$

Using (B.1),

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \leq \|\mathbf{A}\|_2 r^0 + \|\mathbf{d}_i\|_2 \quad (\text{B.9})$$

Using (B.4) and the definition of d results in

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \leq r \quad (\text{B.10})$$

□

Update rule for quadratic deformations As in the linear case, $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$ are the relative positions of the deformed points of \mathfrak{P}^0 . In contrast to the linear case, the relative point positions are $[\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] \tilde{\mathbf{p}}_i^0$ with $\tilde{\mathbf{p}}_i^0 := (p_{i,x}^0, p_{i,y}^0, p_{i,z}^0, (p_{i,x}^0)^2, (p_{i,y}^0)^2, (p_{i,z}^0)^2, p_{i,x}^0 p_{i,y}^0, p_{i,y}^0 p_{i,z}^0, p_{i,z}^0 p_{i,x}^0)^T$. Matrices $\mathbf{A}_{\mathcal{L}}, \mathbf{A}_{\mathcal{Q}}, \mathbf{A}_{\mathcal{M}} \in \mathbb{R}^{3 \times 3}$ are provided by the shape matching procedure. Now, the difference between a deformed point and its transformed initial position is

$$\mathbf{d}_i := [\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] \tilde{\mathbf{p}}_i^0 - \mathbf{p}_i \quad (\text{B.11})$$

As in the linear case, the center and radius of the bounding sphere are transformed such that all deformed points of the set \mathfrak{P}^0 are within the new sphere. Therefore, the quantities q_{\max} and m_{\max} are defined as

$$q_{\max} := \max_{i=1, \dots, n} \left\| \begin{pmatrix} (x_{i,x}^0)^2 - (c_x^0)^2 \\ (x_{i,y}^0)^2 - (c_y^0)^2 \\ (x_{i,z}^0)^2 - (c_z^0)^2 \end{pmatrix} \right\|_2 \quad (\text{B.12})$$

$$m_{\max} := \max_{i=1, \dots, n} \left\| \begin{pmatrix} x_{i,x}^0 x_{i,y}^0 - c_x^0 c_y^0 \\ x_{i,y}^0 x_{i,z}^0 - c_y^0 c_z^0 \\ x_{i,z}^0 x_{i,x}^0 - c_z^0 c_x^0 \end{pmatrix} \right\|_2 \quad (\text{B.13})$$

Claim If the center of the bounding sphere is updated with

$$\mathbf{c} \leftarrow [\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] \tilde{\mathbf{c}}_{\text{rel}}^0 + \mathbf{x}_{\text{cm}} \quad (\text{B.14})$$

and the radius with

$$r \leftarrow \|\mathbf{A}_{\mathcal{L}}\|_2 r^0 + \|\mathbf{A}_{\mathcal{Q}}\|_2 q_{\max} + \|\mathbf{A}_{\mathcal{M}}\|_2 m_{\max} + d \quad (\text{B.15})$$

then all deformed points are enclosed by the updated sphere, i. e.

$$\|\mathbf{x}_i - \mathbf{c}\| \leq r \quad \forall i = 1, \dots, n \quad (\text{B.16})$$

Again, $d := \max_j \|\mathbf{d}_j\|_2$ for all colliding points \mathbf{p}_j and $\tilde{\mathbf{c}}_{\text{rel}}^0 := \mathbf{c}^0 - \mathbf{x}_{\text{cm}}^0$.

Proof Using (B.14) and the definition of \mathbf{p}_i , the distance between a new point position and the new sphere center can be written as

$$\|\mathbf{x}_i - \mathbf{c}\|_2 = \|\mathbf{p}_i + \mathbf{x}_{\text{cm}} - ([\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] \tilde{\mathbf{c}}_{\text{rel}}^0 + \mathbf{x}_{\text{cm}})\|_2 \quad (\text{B.17})$$

Assume without loss of generality $\mathbf{x}_{\text{cm}}^0 = 0$. Using (B.11)

$$\|\mathbf{x}_i - \mathbf{c}\|_2 = \|[\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] (\tilde{\mathbf{p}}_i^0 - \tilde{\mathbf{c}}_{\text{rel}}^0) - \mathbf{d}_i\|_2 \quad (\text{B.18})$$

$\mathbf{x}_{\text{cm}}^0 = 0$ implies that $\mathbf{p}_i^0 = \mathbf{x}_i^0$ and $\tilde{\mathbf{c}}_{\text{rel}}^0 = \mathbf{c}^0$. Therefore, the distance is

$$\|\mathbf{x}_i - \mathbf{c}\|_2 = \|[\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}] (\tilde{\mathbf{x}}_i^0 - \tilde{\mathbf{c}}^0) - \mathbf{d}_i\|_2 \quad (\text{B.19})$$

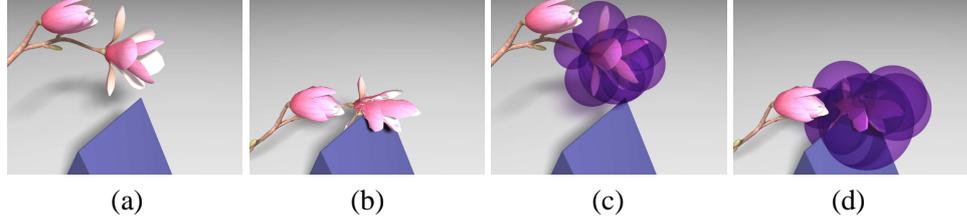


Figure B.4: A quadratically deformable flower collides with a rigid object (a)-(b). The bounding sphere hierarchy is updated according to the description in this appendix (c)-(d). For clarity, only a subset of the bounding spheres is shown.

Applying the triangle inequality,

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \leq \|[\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}](\tilde{\mathbf{x}}_i^0 - \tilde{\mathbf{c}}^0)\|_2 + \|\mathbf{d}_i\|_2 \quad (\text{B.20})$$

Decomposing the right hand side into linear, quadratic and mixed terms, using the triangle inequality and the submultiplicative property of the matrix norm $\|\cdot\|_2$ on the right hand side results in

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{c}\|_2 \leq & \|\mathbf{A}_{\mathcal{L}}\|_2 \|(\mathbf{x}_i^0 - \mathbf{c}^0)\|_2 + \|\mathbf{A}_{\mathcal{Q}}\|_2 \left\| \begin{pmatrix} (x_{i,x}^0)^2 - (c_x^0)^2 \\ (x_{i,y}^0)^2 - (c_y^0)^2 \\ (x_{i,z}^0)^2 - (c_z^0)^2 \end{pmatrix} \right\|_2 \\ & + \|\mathbf{A}_{\mathcal{M}}\|_2 \left\| \begin{pmatrix} x_{i,x}^0 x_{i,y}^0 - c_x^0 c_y^0 \\ x_{i,y}^0 x_{i,z}^0 - c_y^0 c_z^0 \\ x_{i,z}^0 x_{i,x}^0 - c_z^0 c_x^0 \end{pmatrix} \right\|_2 + \|\mathbf{d}_i\|_2 \end{aligned} \quad (\text{B.21})$$

Using (B.1), (B.12) and (B.13),

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{c}\|_2 \leq & \|\mathbf{A}_{\mathcal{L}}\|_2 r^0 + \|\mathbf{A}_{\mathcal{Q}}\|_2 q_{\max} + \\ & + \|\mathbf{A}_{\mathcal{M}}\|_2 m_{\max} + \|\mathbf{d}_i\|_2 \end{aligned} \quad (\text{B.22})$$

Using the definition of d and the updated radius r results in

$$\|\mathbf{x}_i - \mathbf{c}\|_2 \leq r \quad (\text{B.23})$$

□

Fig. B.4 illustrates the update of the sphere hierarchy for a deforming object.

Update rule for rigid objects For rigid objects, the transformation matrix \mathbf{R} is orthogonal. Thus, the eigenvalues of $\mathbf{R}^T \mathbf{R} = \mathbf{R}^{-1} \mathbf{R} = \mathbf{I}$ are one and $\|\mathbf{R}\|_2 = 1$. The sphere centers and radii are updated as in the linear case.

Clustered objects According to the deformation model, all clusters of an object are of the same type, i. e. rigid, linear or quadratic. If an object consists of more than one cluster, the shape matching process computes a transformation matrix for each cluster. In the linear and the quadratic case, the norm of the transformation matrix has to be computed for each cluster.

Collision query

In order to test two objects for collision, the root nodes of the hierarchy are tested for interference. If they interfere, both hierarchies are traversed recursively. Therefore, only children of colliding spheres need to be updated. At leaf nodes, primitives of one object are tested against enclosing spheres of the other hierarchy. Hence, point-based objects can be handled. If connectivity information is available, traditional interference tests among primitives can be performed. For experiments, collision queries for point-based objects and for objects with triangulated surfaces have been implemented.

Performance

In most cases, the cost for updating the hierarchy is linear in C where C is the number of pairs of primitives in close proximity. Assume that C leaf nodes in the hierarchy have to be tested. Due to spatial coherence, they will share about $\frac{C}{2}$ parent nodes which themselves have $\frac{C}{4}$ parent nodes. Thus, in most cases $2C$ nodes have to be updated. However, an exact analysis is difficult since the cost function depends on the distribution of the C leaf nodes.

If the object is clustered, a separate branch in the hierarchy is computed for each cluster. Thus, each bounding sphere belongs to only one cluster and can be updated in constant time. The cost to update the root and those leaves that are within the cluster overlap regions is linear in the number of clusters they belong to. However, neither the number of clusters nor the percentage of cluster overlap depend on the number of primitives. Further, the number of clusters is significantly smaller than the number of primitives.

In the linear and rigid cases, a sphere center is updated with two vector additions and one matrix-vector multiplication. In the quadratic case, two vector additions and three matrix-vector multiplications are required. The radius is updated with one multiplication and one addition in the linear case, with one addition in the rigid case, and with three multiplications and four additions in the quadratic case.

Prior to a hierarchy traversal, the following values have to be computed once per object cluster. In the linear case, the matrix norm of \mathbf{A} and the value d are computed. The norm of \mathbf{A} can be computed in constant time and the computation of d is linear in the number of colliding points C . In the quadratic case, the matrix norms $\mathbf{A}_{\mathcal{L}}$, $\mathbf{A}_{\mathcal{Q}}$ and $\mathbf{A}_{\mathcal{M}}$ can be computed in constant time and d is computed as described for the linear case.

The center of mass of the undeformed model \mathbf{x}_{cm}^0 , the center of mass of the deformed model \mathbf{x}_{cm} , and the deformation matrices \mathbf{A} and $[\mathbf{A}_{\mathcal{L}} \mathbf{A}_{\mathcal{Q}} \mathbf{A}_{\mathcal{M}}]$ are provided by the geometric deformation approach. Further, the radii r^0 and the values q_{max} , m_{max} in the quadratic case can be pre-computed. They only depend on the initial undeformed state.

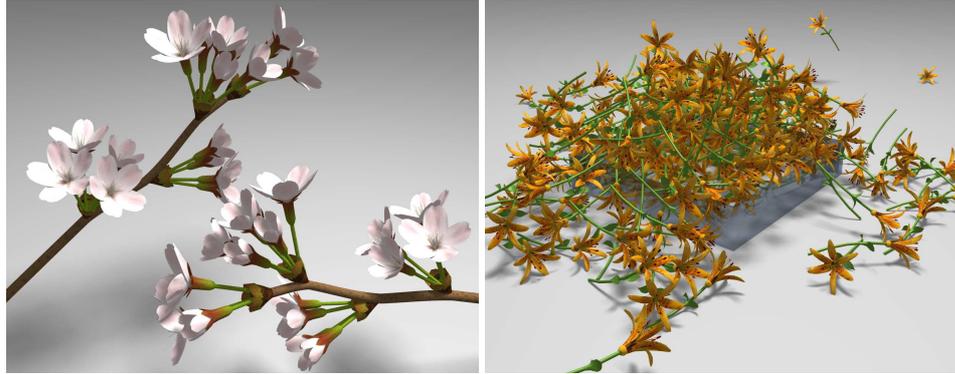


Figure B.5: Left: Simulation of two geometrically complex models. For the performance measurements illustrated in Fig. B.6, the overall number of points in the scene varies from 8k to 160k. The number of sphere updates ranges from 5.2k to 93k for optimally refitted hierarchies, and from 5.4k to 96k for hierarchies that are updated with our approach. Right: Simulation of 300 deformable objects with an overall number of 131k points. Level-of-detail collision handling can be computed at interactive rates as shown in Tab. B.1.

B.4 Results

In this section, three experiments are discussed that illustrate the efficiency of the proposed approach. All measurements were taken on an Intel Xeon, 3.8 GHz CPU.

For the experiments, a simple penalty-based collision response scheme has been implemented. The normalized distance vector between the sphere centers is considered as contact space. If the collision response is performed at the primitive-level, then an edge-based repulsion scheme in the spirit of [BFA02] is applied.

Interference query The approach being presented updates a bounding sphere hierarchy of a deforming object in an output-sensitive manner, i. e. only potentially colliding bounding spheres are updated. This is achieved by considering the deformation matrices obtained from the underlying deformable model. Since (B.15) is a conservative estimation, the updated spheres have larger radii compared to minimally enclosing spheres. As a consequence, the bounding spheres are not as tight-fitting as possible and a collision query generally traverses more nodes as would be required in case of minimally enclosing spheres. However, the following experiment shows that this fact has only a minor influence on the collision query performance. On the other hand, computing the minimum enclosing sphere of a set of points is an expensive operation that is hardly feasible in interactive simulations. For the measurements, two objects have been taken with varying resolutions ranging from 8k to 160k points for both objects. The two objects move towards each other, collide, and deform under the influence of the contact forces. Thus, the sphere hierarchies have to be updated in each time step. Fig. B.5 shows the test

scenario and Fig. B.6 provides the measurements. Since the same dynamic simulation is computed for all measurements, the number of updated spheres is varying according to the resolution of the objects. For optimally refitted hierarchies, the maximum number of sphere updates per simulation step ranges from 5.2k to 93k, depending on the object resolution. For hierarchies that are updated with our approach, the maximum number of sphere updates ranges from 5.4k to 96k.

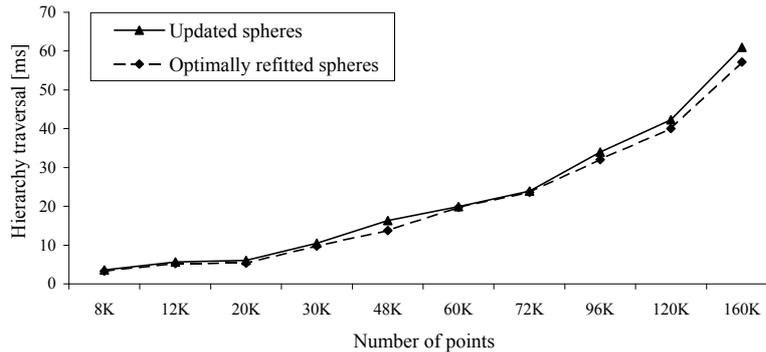


Figure B.6: Comparison of the collision query times of an optimally refitted bounding sphere hierarchy versus the updated sphere hierarchy using the presented approach. The optimally refitted spheres have smaller radii than the conservatively updated spheres. Since in the latter case more spheres have to be visited and updated, the collision query is less efficient. However, the diagram shows that the difference is almost negligible.

Collision detection that is accelerated by bounding volume hierarchies is particularly well-suited for time-critical collision handling. The interference query can be stopped at an early stage, and the collision response can then be performed with the collision information available on that stage. To illustrate this, a simulation of 300 deformable objects with an overall number of 131k points as depicted in Fig. B.5 is performed. Table B.1 shows the performance of the collision handling scheme computed at various levels of the hierarchy ranging from 2 to 6. The number of colliding spheres varies according to the hierarchy level. Further, the computing time for the collision query and the collision response depends on the hierarchy level. If the collision response is computed at a lower level, it is less accurate compared to the response at a higher level. Thus, accuracy and efficiency can be balanced. Table B.1 shows that the presented collision handling scheme can handle 131k points at interactive rates.

Deformation simulations The third experiment depicted in Fig. B.7 shows that the presented approach can be integrated into interactive deformation simulations. In the experiment, three objects with an overall number of 19k points are simulated. The average collision detection time is 4.39 ms and collision response takes 10.66 ms. The overall time of the geometric deformation is 24.33 ms, and the vi-

Level	# Contacts	Det. [ms]	Resp. [ms]	Total [ms]
2	1031	18.41	24.11	42.52
3	1650	22.19	27.01	49.20
4	7511	30.24	32.03	62.27
5	27195	58.61	45.35	103.96
6	42635	93.97	52.88	146.85

Table B.1: The collision response can be computed efficiently at a lower level. At higher levels, the response is more accurate. Fig. B.5 illustrates the test scenario.

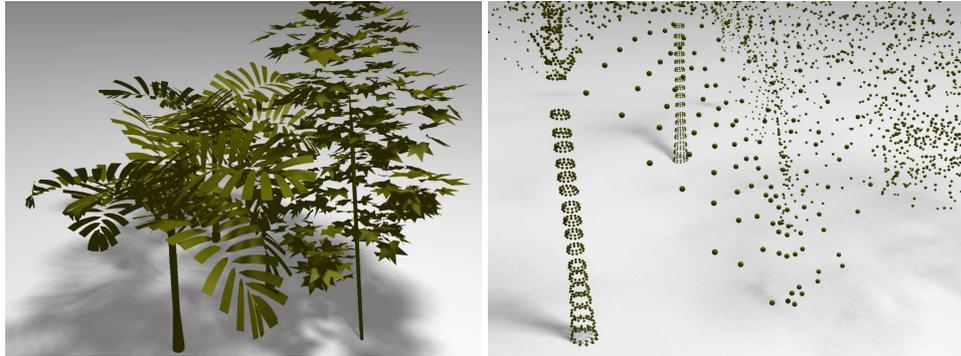


Figure B.7: Interactive animation of three deformable objects. The scene consists of 19k points and can be computed at 21 frames per second (left). The simulated point cloud with 19k mass points (right).

sualization takes 7.8 ms. Thus, the total computing time per frame is 47.19 ms and the animation runs at 21 frames per second. The time for detecting and resolving the collisions is less than 15 ms. It is no longer the bottleneck of the simulation.

The collision query returns a set of pairs of colliding spheres or a set of sphere / point pairs. Since the sphere centers are considered approximations of the enclosed point set, collision response can be consistently computed on pairs of points. A contact normal is computed for each contact pair, where the plausibility of the collision response depends on the accuracy of the contact normal. If topology information is available, e. g. for triangulated surfaces, a more accurate scheme in the spirit of [BFA02] is applied. Thus, even stacking problems can be handled as illustrated in Fig. B.8.

B.5 Discussion

Model partition schemes provide an efficient way to detect collisions. However, for deforming objects, the refit of the hierarchy has been the bottleneck, since it has been necessary to consider each deformed primitive. Thereby, an early cull of non-colliding parts has been impossible.



Figure B.8: By employing the response scheme presented in [BFA02], stacking problems with deformable objects can be reproduced.

The discussed approach addresses this problem by proposing an output-sensitive way to refit the hierarchy. Since the global deformation is described by a transformation matrix, the bounding spheres can be updated by employing this transformation matrix. Consequently, the update of one single sphere can be performed in constant time. Thus, during the interference query, non-colliding parts can be culled and the complexity of the query is $\mathcal{O}(C)$, where C is the number of primitives in close proximity. This contrasts traditional approaches such as [Van97, LAM01, LAM05] that do not lower the theoretical $\mathcal{O}(N)$ complexity although they propose intelligent update strategies.

The approach shares similarities with the BD-tree of James and Pai [JP04]. Still, there are some important differences. The reduced deformation models express the deformation coordinates as a superposition of a set of deformation fields. In contrast, the geometric deformation model of Müller *et al.* [MHTG05] basically simulates an unconnected particle cloud with $3N$ DOFs. The restitutive behavior of the elastic material is achieved by registering the particle cloud with a geometrical transformation. The particles are then moved towards the resulting goal positions. The approach then effectively updates the bounding spheres such that they enclose the goal positions, from which it is known that they coincide in general with the simulated particles. While this holds in general for the non-colliding particles, the colliding particles are moving under the influence of contact forces, and therefore they do not converge towards the computed goal positions. Consequently, the computation of d requires to consider all collided particles.

While this update procedure works well for moderately deformed objects, it will eventually fail for soft objects that suffer from large local deformations. This is because the local deformations influence the shape matching procedure, which in turn results in a deflection between the goal positions and the positions of the simulated particles. These errors are minimal if the global deformation dominates the local deformation, which is the case for all of the simulated objects. Moreover, the approach relies on the temporal coherence, which is usually given in dynamic

simulations.

Future work Increasing the accuracy of the method is a difficult problem, having in mind the limitations noted above. Still, a clever parametrization of the deformation model alleviates the resulting artifacts. A future direction could be dictated by the recent advances in geometric deformation modeling [RJ07], for which the approach has to be adapted accordingly.

Appendix C

Quaternions

Quaternions are an important concept that is widely used in many areas in computer graphics. In general, a unit quaternion represents a rotation. Thus, quaternions can be used to transform a geometric primitive in e. g. a visualization program. However, a unit quaternion can as well be used to represent the orientation of e. g. a rigid object. If torques act on the rigid object, the quaternion must be evolved in time. In mechanics, quaternions with unit length are also denoted as Euler parameters or Euler-Rodriguez parameters. In this appendix, a brief summary of the quaternion algebra and the related quaternion transformations is given.

C.1 Quaternion representation

A quaternion $q \in \mathbb{H}$ is a hyper-complex quantity

$$q = q_1i + q_2j + q_3k + q_4 \quad (\text{C.1})$$

where i , j and k are imaginary numbers. In this thesis, the components q_1 , q_2 , q_3 and q_4 of q are considered to constitute a vector

$$\mathbf{q} = (q_1, q_2, q_3, q_4)^T \in \mathbb{R}^4, \quad (\text{C.2})$$

which allows for a consistent writing in matrix-vector terminology. The *conjugate* of a quaternion \mathbf{q} is

$$\bar{\mathbf{q}} = (-q_1, -q_2, -q_3, q_4)^T, \quad (\text{C.3})$$

the *norm* of a quaternion \mathbf{q} is

$$\|\mathbf{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (\text{C.4})$$

and the *inverse* of a quaternion \mathbf{q} is

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{\|\mathbf{q}\|^2} \quad (\text{C.5})$$

C.2 Quaternion algebra

While the addition of two quaternions \mathbf{p} and \mathbf{q} is commutative, the multiplication is *not*. Instead, the multiplication is defined as

$$\mathbf{q} \circ \mathbf{p} = \begin{pmatrix} q_4 \mathbf{p}_{1:3} + p_4 \mathbf{q}_{1:3} + \mathbf{q}_{1:3} \times \mathbf{p}_{1:3} \\ q_4 p_4 - \mathbf{q}_{1:3}^T \mathbf{p}_{1:3} \end{pmatrix} \quad (\text{C.6})$$

where $\mathbf{q}_{1:3}$ forms a vector $(q_1, q_2, q_3)^T$ from the first three components of \mathbf{q} . Often, it is more convenient to write the quaternion multiplication in matrix-vector notation,

$$\mathbf{q} \circ \mathbf{p} = \mathbf{Q}(\mathbf{q})\mathbf{p} \quad (\text{C.7})$$

where $\mathbf{Q}(\mathbf{q})$ is the *quaternion matrix* associated with \mathbf{q} ,

$$\mathbf{Q}(\mathbf{q}) = \begin{pmatrix} q_4 \mathbf{I} + \tilde{\mathbf{q}}_{1:3} & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{pmatrix} = \begin{pmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix} \quad (\text{C.8})$$

Here, $\tilde{\mathbf{q}}_{1:3}$ is the *skew matrix* of $\mathbf{q}_{1:3}$, i. e.

$$\tilde{\mathbf{q}}_{1:3} = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix} \quad (\text{C.9})$$

The inverse multiplication $\mathbf{p} \circ \mathbf{q}$ can be consistently computed in terms of the *conjugate quaternion matrix* $\overline{\mathbf{Q}}$,

$$\mathbf{p} \circ \mathbf{q} = \mathbf{Q}(\mathbf{p})\mathbf{q} = \overline{\mathbf{Q}}(\mathbf{q})\mathbf{p} \quad (\text{C.10})$$

where $\overline{\mathbf{Q}}(\mathbf{q})$ is the conjugate quaternion matrix associated with \mathbf{q} ,

$$\overline{\mathbf{Q}}(\mathbf{q}) = \begin{pmatrix} q_4 \mathbf{I} - \tilde{\mathbf{q}}_{1:3} & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{pmatrix} = \begin{pmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix} \quad (\text{C.11})$$

C.3 Quaternions and rotations

A *unit quaternion* is a quaternion with length 1, i. e.

$$\|\mathbf{q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \quad (\text{C.12})$$

Unit quaternions constitute the special unitary group $SU(2)$. In turn, there is a surjective homomorphism between $SU(2)$ and $SO(3)$, which is the group of rotations in the three-dimensional space. The mapping is not injective since both the unit quaternion \mathbf{q} and its conjugate $\overline{\mathbf{q}}$ represent the same rotation. This two-to-one ambiguity has to be considered in the treatment of quaternions.

Angle-axis representation

A unit quaternion can be interpreted as a rotation by an angle ϕ around an arbitrary unit-length axis $\mathbf{n} = (n_x, n_y, n_z)^T$. Thus the quaternion \mathbf{q} is understood as

$$\mathbf{q} = \begin{pmatrix} \sin(\phi/2)\mathbf{n} \\ \cos(\phi/2) \end{pmatrix} \quad (\text{C.13})$$

The vector \mathbf{n} is related to the vector $\mathbf{q}_{1:3}$ by $\mathbf{q}_{1:3} = \sin(\phi/2)\mathbf{n}$.

Rotations and the quaternion product

If unit quaternions are considered, then the quaternion multiplication can be interpreted geometrically as a composition of rotations. If \mathbf{p} and \mathbf{q} are unit quaternions, then the composite rotation \mathbf{r} , first about \mathbf{p} , then about \mathbf{q} , conforms to multiplying \mathbf{p} with \mathbf{q} , i. e. $\mathbf{r} = \mathbf{q} \circ \mathbf{p}$.

Since quaternions represent rotations, they can be employed to rotate vectors or points about arbitrary axes. This is accomplished by first transforming the vector $\mathbf{x} = (x, y, z)^T$ into a quaternion, and then multiplying this quaternion with the rotation quaternions. The transformation of a vector \mathbf{x} into a quaternion \mathbf{q}_x is

$$\mathbf{q}_x(\mathbf{x}) = (\mathbf{x}^T, 0)^T \quad (\text{C.14})$$

The quaternion \mathbf{q}_x is then said to be *dual* to the point \mathbf{x} .

The rotation of the point \mathbf{x} around an axis \mathbf{n} by an angle ϕ is accomplished by multiplying \mathbf{q}_x and the associated rotation quaternion $\mathbf{q} = (\sin \frac{\phi}{2}\mathbf{n}, \cos \frac{\phi}{2})$ according to the rule [Ebe04]

$$\text{Rot}_q(\mathbf{x}) = (\mathbf{q} \circ \mathbf{q}_x \circ \bar{\mathbf{q}})_{1:3} \quad (\text{C.15})$$

The composite rotation of \mathbf{x} first by \mathbf{p} followed by \mathbf{q} is thus

$$\begin{aligned} \text{Rot}_{qp}(\mathbf{x}) &= (\mathbf{q}(\mathbf{p} \circ \mathbf{q}_x \circ \bar{\mathbf{p}})\bar{\mathbf{q}})_{1:3} \\ &= ((\mathbf{q} \circ \mathbf{p}) \circ \mathbf{q}_x \circ \overline{(\mathbf{q} \circ \mathbf{p})})_{1:3} \end{aligned} \quad (\text{C.16})$$

C.4 Quaternion conversions

This section describes how to convert quaternions to rotation matrices and vice versa.

From unit quaternions to rotation matrices

Since unit quaternions represent rotations, a rotation matrix can be parameterized with unit quaternions. The orthonormal rotation matrix \mathbf{R} that corresponds to the quaternion \mathbf{q} is obtained as

$$\mathbf{R} = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix} \quad (\text{C.17})$$

The columns of \mathbf{R} can be interpreted as the axes spanning an orthonormal basis. Therefore, in the context of elastic rods (see Chapter 3), the above identity allows to compute the directors, i. e. the material frame, from the quaternion \mathbf{q} . The above identity holds only if \mathbf{q} is a unit quaternion.

From rotation matrices to unit quaternions

The construction of a quaternion \mathbf{q} from a rotation matrix \mathbf{R} is not as straightforward as the opposite way. Further, a quaternion is a two-to-one representation of a rotation. That means, there are always two quaternions that represent exactly the same rotation. This results in a sign ambiguity in the construction of the quaternion. A robust construction is found in [Sho85] and lists as follows:

ConstructQuaternion(\mathbf{R})

$$q_4^2 = 1/4(1 + R_{11} + R_{22} + R_{33})$$

if $q_4^2 > \varepsilon$ **then**

$$q_4 = \sqrt{q_4^2} \text{ or } q_4 = -\sqrt{q_4^2}$$

$$q_1 = (R_{32} - R_{23})/4q_4$$

$$q_2 = (R_{13} - R_{31})/4q_4$$

$$q_3 = (R_{21} - R_{12})/4q_4$$

else

$$q_4 = 0$$

$$q_1^2 = -1/2(R_{22} + R_{33})$$

if $q_1^2 > \varepsilon$ **then**

$$q_1 = \sqrt{q_1^2}$$

$$q_2 = R_{12}/2q_1$$

$$q_3 = R_{13}/2q_1$$

else

$$q_1 = 0$$

$$q_2^2 = 1/2(1 - R_{33})$$

if $q_2^2 > \varepsilon$ **then**

$$q_2 = \sqrt{q_2^2}$$

$$q_3 = R_{23}/2q_2$$

else

$$q_2 = 0$$

$$q_3 = 1$$

end if

end if

end if

The drawback of the quaternion construction is that the parameters of the quaternion \mathbf{q} cannot be represented consistently as mathematical functions in the entries of the rotation matrix \mathbf{R} .

In the context of CORDE, this function is executed in the initialization of the rods, where the quaternions are computed from the centerline under the assumption that the material torsion is 0. This conforms to the notion of *parallel transport* in [BWR*08].

C.5 Derivation of strain rates in terms of quaternions

In this section, the expression (3.12) that relates the strain rates u_k to the quaternions \mathbf{q} are derived. The derivation of the relation between the angular velocity ω and the quaternions \mathbf{q} is similar.

The Darboux vector \mathbf{u} can be interpreted as the area that is swept by the directors if the directors are spatially evolved. It is related to the directors \mathbf{d}_k by

$$\mathbf{d}'_k = \mathbf{u} \times \mathbf{d}_k, \quad k = 1, 2, 3 \quad (\text{C.18})$$

where the components u_k of \mathbf{u} are directly proportional to the strain rates. Since the \mathbf{d}_k define an orthonormal basis, $\mathbf{u} = u_1 \mathbf{d}_1 + u_2 \mathbf{d}_2 + u_3 \mathbf{d}_3$. For \mathbf{d}'_3 , one then obtains for example

$$\begin{aligned} \mathbf{d}'_3 &= (u_1 \mathbf{d}_1 + u_2 \mathbf{d}_2 + u_3 \mathbf{d}_3) \times \mathbf{d}_3 \\ &= u_1 \mathbf{d}_1 \times \mathbf{d}_3 + u_2 \mathbf{d}_2 \times \mathbf{d}_3 + u_3 \mathbf{d}_3 \times \mathbf{d}_3 \\ &= -u_1 \mathbf{d}_2 + u_2 \mathbf{d}_1 \end{aligned} \quad (\text{C.19})$$

where the identities $\mathbf{d}_3 = \mathbf{d}_1 \times \mathbf{d}_2$ and $\mathbf{d}_k \times \mathbf{d}_k = 0, k = 1, 2, 3$ have been used. Multiplying (C.19) by $-\mathbf{d}_2$ yields

$$-\mathbf{d}_2 \cdot \mathbf{d}'_3 = u_1 \mathbf{d}_2 \cdot \mathbf{d}_2 - u_2 \mathbf{d}_1 \cdot \mathbf{d}_2 = u_1 \quad (\text{C.20})$$

For u_2 and u_3 , one similarly obtains $u_2 = -\mathbf{d}_3 \cdot \mathbf{d}'_1$ and $u_3 = -\mathbf{d}_1 \cdot \mathbf{d}'_2$.

The relation between the directors \mathbf{d}_k and the quaternion \mathbf{q} is given in (C.17). To compute \mathbf{d}'_k , it has to be considered that \mathbf{d}'_k is a function of \mathbf{q} and \mathbf{q} is a function of the curve parameter s :

$$\mathbf{d}'_k = \mathbf{d}'_k(\mathbf{q}(s)) = \frac{\partial \mathbf{d}_k(\mathbf{q}(s))}{\partial s} = \frac{\partial \mathbf{d}_k}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial s} = \mathbf{J}_k \mathbf{q}' \quad (\text{C.21})$$

where the chain rule of partial differentiation has been employed. \mathbf{J}_k is the Jacobi matrix $\mathbf{J}_k = \frac{\partial \mathbf{d}_k}{\partial \mathbf{q}}$ that is obtained by symbolic differentiation. For u_1 ,

$$u_1 = -\mathbf{d}_2 \cdot \mathbf{J}_3 \mathbf{q}' = -\mathbf{J}_3^T \mathbf{d}_2 \cdot \mathbf{q}' \quad (\text{C.22})$$

which results from applying basic linear algebra identities. Similar expressions are obtained for u_2 and u_3 . To bring (C.22) to the desired form (3.12), the product $-\mathbf{J}_3^T \mathbf{d}_2$ is symbolically evaluated to obtain

$$-\mathbf{J}_3^T \mathbf{d}_2 = 2 \begin{pmatrix} q_4 & q_3 & -q_2 & -q_1 \end{pmatrix}^T = 2\mathbf{B}_1 \mathbf{q} \quad (\text{C.23})$$

with the skew-symmetric matrix \mathbf{B}_1

$$\mathbf{B}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}$$

The strain rate u_1 can then be obtained by combining (C.22) and (C.23). The matrices \mathbf{B}_2 and \mathbf{B}_3 are obtained in a similar manner as

$$\mathbf{B}_2 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

The skew-symmetric matrices \mathbf{B}_k and the resulting vectors $\mathbf{B}_k \mathbf{q}$ have several important properties that are discussed in [Dic94]. A similar analysis [Dic94] can be done to obtain the angular velocities ω_k^0 with respect to the reference frame, resulting in matrices \mathbf{B}_k^0 :

$$\mathbf{B}_1^0 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}_2^0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

and

$$\mathbf{B}_3^0 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Appendix D

The non-linear conjugate gradient method

In this section, the non-linear conjugate gradient (CG) method that is employed in the variational subdivision method in Sec. 4.1.3 is briefly explained. The method is exemplified by employing the CORDE deformation model as presented in Chapter 3.

D.1 Principles

The non-linear CG method is used to optimize a non-linear function $f(\mathbf{g})$, where \mathbf{g} is the vector containing the N variables. In contrast to the linear CG method, the non-linear CG method does not solve a linear system but employs the gradient $\nabla_{\mathbf{g}}f(\mathbf{g})$ alone to compute the search direction.

In the context of the variational subdivision, the non-linear function conforms to the elastic energy of the subdivision region (4.3). For CORDE, the integral in (4.3) can be evaluated symbolically, thus (4.3) is effectively a sum of non-linear functions. To compute the gradient $\nabla_{\mathbf{g}}f(\mathbf{g})$, one has to consider that the energy, differentiated with respect to the coordinates \mathbf{g} , results in the forces. Consequently, the gradient of $f(\mathbf{g})$ corresponds to the elastic forces of the deformation model, that have already been employed to compute the deformation in Chapter 3. That is what makes the implementation of the non-linear CG method quite straight-forward.

In the following discussion, \mathbf{g}_n conforms to the coordinates of \mathbf{g} in the n -th iteration of the CG method. Thus, let \mathbf{g}_0 be the starting configuration. In the implementation, this conforms to the configuration as depicted in Fig. 4.2 middle. Then the non-linear CG method (see Algorithm 3) iteratively optimizes the coordinates \mathbf{g}_n with respect to the function $f(\mathbf{g})$. To accomplish this, it iteratively updates the conjugate direction $\Lambda\mathbf{g}_n$, which conforms to the search direction, and which is in contrast to a standard steepest descent method.

Experiments indicate that 5-10 iterations provide sufficient accuracy. Since the energy density W has been constrained to be a convex function, the CG method

```

NonLinearCG( $f, \mathbf{g}_0$ )
 $\Delta \mathbf{g}_0 \leftarrow -\nabla_{\mathbf{g}} f(\mathbf{g}_0)$ 
 $\alpha_0 \leftarrow \text{LineSearch}(f, \Delta \mathbf{g}_0)$ 
 $\mathbf{g}_1 \leftarrow \mathbf{g}_0 + \alpha_0 \Delta \mathbf{g}_0$ 
// Initialize the conjugate direction
 $\Lambda \mathbf{g}_0 \leftarrow \Delta \mathbf{g}_0$ 
 $n \leftarrow 1$ 
repeat
   $\Delta \mathbf{g}_n \leftarrow -\nabla_{\mathbf{g}} f(\mathbf{g}_n)$ 
  // Compute  $\beta_n$  by employing the Fletcher-Reeves formula
   $\beta_n \leftarrow \frac{\Delta \mathbf{g}_n^T \Delta \mathbf{g}_n}{\Delta \mathbf{g}_{n-1}^T \Delta \mathbf{g}_{n-1}}$ 
  // Update the conjugate direction
   $\Lambda \mathbf{g}_n \leftarrow \Delta \mathbf{g}_n + \beta_n \Lambda \mathbf{g}_{n-1}$ 
   $\alpha_n \leftarrow \text{LineSearch}(f, \Lambda \mathbf{g}_n)$ 
   $\mathbf{g}_{n+1} \leftarrow \mathbf{g}_n + \alpha_n \Lambda \mathbf{g}_n$ 
   $n \leftarrow n + 1$ 
until sufficient accuracy;

```

Algorithm 3: The non-linear conjugate gradient algorithm.

will always end up in a global minimum. By only updating the coordinates \mathbf{g}_i^+ and \mathbf{g}_{i+1}^+ , it is accounted for the boundary conditions.

D.2 Line search

In line 2 and 13 of the nonlinear CG algorithm, the step length parameter α_n has to be computed that minimizes $f(\mathbf{g})$ along a given search direction $\Delta \mathbf{g}$. To accomplish this, a line search strategy is employed. There exist several variants of this technique in the literature [NW99]. Here, a combination of the backtracking line search strategy by employing the Wolfe conditions is employed.

The algorithm (see Algorithm 4) consists of two steps: First, it searches the step length α_k such that α_k decreases sufficiently (according to some heuristic criterion), and such that the slope of the function $f(\mathbf{g} + \alpha_k \nabla_{\mathbf{g}} f(\mathbf{g}))$ is larger than the slope of $f(\mathbf{g})$. The first condition is the *Armijo-condition*, the second condition is the *curvature condition*, together, they constitute the *Wolfe conditions*. The line search algorithm starts with choosing an initial step length α_0 . Then, α_k is computed iteratively, where k is the iteration number.

Having found a good step length α_k , it performs a line search along the given search direction, starting with the step length α_k , and employing a binary search strategy, i. e. halving the step length if $f(\mathbf{g})$ becomes larger, and doubling the step length if $f(\mathbf{g})$ becomes smaller. Fig. D.1 illustrates the line search.

The initial guesses for the step sizes for the first loop are $\alpha_0^{(r)} = ch^2 m_i^{-1}$ for the

```

LineSearch( $f, \mathbf{g}_n, \Delta \mathbf{g}_n$ )
// First: compute the step length  $\alpha_0$ 
// Choose starting  $\alpha_0$  (see text)
repeat
  cond1  $\leftarrow f(\mathbf{g}_n + \alpha_0 \Delta \mathbf{g}_n) \leq f(\mathbf{g}_n) + c_1 \alpha_0 \mathbf{g}_n^T \nabla_{\mathbf{g}} f(\mathbf{g}_n)$  // Armijo-condition
  cond2  $\leftarrow \mathbf{g}_n^T \nabla_{\mathbf{g}} f(\mathbf{g}_n + \alpha_0 \mathbf{g}_n) \geq c_2 \mathbf{g}_n^T \nabla_{\mathbf{g}} f(\mathbf{g}_n)$  // curvature condition
  if cond1 and cond2 then break
   $\alpha_0 \leftarrow \frac{1}{2} \alpha_0$ 
until break
// Second: minimize  $f(\mathbf{g}_n)$  along the search direction  $\Delta \mathbf{g}_n$ .
 $s_0 \leftarrow 0, \Delta s_0 \leftarrow 0$ 
 $s_1 \leftarrow 0, \Delta s_1 \leftarrow 1$ 
repeat
  if  $f(\mathbf{g}_n + \alpha_0(s_k + \Delta s_k) \Delta \mathbf{g}_n) < f(\mathbf{g}_n + \alpha_0(s_{k-1} + \Delta s_{k-1}) \Delta \mathbf{g}_n)$  then
     $s_{k+1} \leftarrow s_k + \Delta s_k$ 
     $\Delta s_{k+1} \leftarrow \Delta s_k$ 
  else
     $s_{k+1} \leftarrow s_k$ 
     $\Delta s_{k+1} \leftarrow \frac{1}{2} \Delta s_k$ 
  end if
   $k \leftarrow k + 1$ 
until sufficient accuracy;
return  $\alpha_0(s_k + \Delta s_k)$ 

```

Algorithm 4: The line search algorithm.

mass-points \mathbf{r}_i and $\alpha_0^{(q)} = ch^2 \text{trace}(\mathbf{J})^{-1}$ for the quaternions \mathbf{q}_i , with h the simulation time step, and \mathbf{J} the inertia tensor. Intuitively, these guesses guarantee convergence for the applied time step. These initial guesses are multiplied with a constant c , here $c = 16$. The first loop then iteratively makes the guesses α_0 smaller until the Wolfe conditions hold. If no α_0 exists such that the Wolfe conditions hold, the loop is stopped after a few iterations. c_1 and c_2 are two constants with $0 < c_1 < c_2 < 1$. In the experiments, $c_1 = 0.3$ and $c_2 = 0.99$.

The second loop basically performs a binary search for the minimum $f(\mathbf{g}_n)$ along the search direction $\Delta \mathbf{g}_n$. The value $\alpha_0(s_k + \Delta s_k)$ that is returned corresponds to the optimum step length that minimizes f along the given search direction.

Bibliography

- [ACYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2005), 617–625.
- [Ant95] ANTMAN S. S.: *Nonlinear Problems of Elasticity*. Springer Verlag, 1995.
- [AUK92] ANJYO K., USAMI Y., KURIHARA T.: A simple method for extracting the natural beauty of hair. *SIGGRAPH Computer Graphics* 26, 2 (1992), 111–120.
- [BAC*06] BERTAILS F., AUDOLY B., CANI M.-P., QUERLEUX B., LEROY F., LÉVÊQUE J.-L.: Super-helices for predicting the dynamics of natural hair. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2006), pp. 1180–1187.
- [Bar89] BARAFF D.: Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics (Proc. Siggraph)* (1989), vol. 23, pp. 223–232.
- [Bar91] BARAFF D.: Coping with friction for non-penetrating rigid body simulation. In *Proc. Computer Graphics and Interactive Techniques* (1991), pp. 31–41.
- [Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. In *Proc. Computer Graphics and Interactive Techniques* (1994), pp. 23–34.
- [Bau72] BAUMGARTE J.: Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 1 (1972), 1–16.
- [BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics* (2002), 594–603.
- [BH00] BRIGGS W., HENSON V.: *A Multigrid Tutorial*. SIAM, 2000.

- [BJ05] BARBIC J., JAMES D. L.: Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Transaction on Graphics (Proc. SIGGRAPH)* (2005), 982–990.
- [BKCNO3] BERTAILS F., KIM T.-Y., CANI M.-P., NEUMANN U.: Adaptive Wisp Tree: a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 207–213.
- [BLM04] BROWN J., LATOMBE J.-C., MONTGOMERY K.: Real-time knot tying simulation. *The Visual Computer* 20, 2–3 (2004), 165–179.
- [Blo94] BLOOMENTHAL J.: An implicit surface polygonizer. In *Graphics Gems IV*. 1994, pp. 324–349.
- [BMWG07] BERGOU M., MATHUR S., WARDETZKY M., GRINSPUN E.: Tracks: toward directable thin shells. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 50.
- [BNC96] BRO-NIELSEN M., COTIN S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum* 15, 3 (1996), 57–66.
- [BP04] BOYER F., PRIMAULT D.: Finite element of slender beams in finite transformations: a geometrically exact approach. *International Journal of Numerical Methods in Engineering*, 59 (2004), 669–702.
- [BPK05] BISCHOFF S., PAVIC D., KOBBELT L.: Automatic restoration of polygon models. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 24, 4 (2005), 1332–1352.
- [BPWG07] BOTSCH M., PAULY M., WICKE M., GROSS M.: Adaptive space deformations based on rigid cells. *Computer Graphics Forum* 26, 3 (2007), 339–347.
- [BS06] BENDER J., SCHMITT A.: Constraint-based collision and contact handling using impulses. In *Proc. Computer Animation and Social Agents* (2006), pp. 3–11.
- [BW92] BARAFF D., WITKIN A.: Dynamic simulation of non-penetrating flexible bodies. *Computer Graphics (Proc. Siggraph)* 26, 2 (1992), 303–308.
- [BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proc. SIGGRAPH* (1998), pp. 43–54.

- [BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 16.
- [BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete Elastic Rods. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2008). to appear.
- [CCK05] CHOE B., CHOI M. G., KO H.-S.: Simulating complex hair with robust collision handling. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), pp. 153–160.
- [CDM*02] CUTLER B., DORSEY J., MCMILLAN L., MÜLLER M., JAGNOW R.: A procedural approach to authoring solid models. In *ACM Transaction on Graphics (Proc. SIGGRAPH)* (2002), pp. 302–311.
- [CDM04] CUTLER B., DORSEY J., MCMILLAN L.: Simplification and improvement of tetrahedral models for simulation. In *Proc. Eurographics Symposium on Geometry Processing* (2004), pp. 95–104.
- [CDRR04] CHENG S.-W., DEY T. K., RAMOS E. A., RAY T.: Quality meshing for polyhedra with small angles. In *Proc. Symposium on Computational Geometry* (2004), pp. 290–299.
- [CGC*02a] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: A multiresolution framework for dynamic deformations. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 41–47.
- [CGC*02b] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: Interactive skeleton-driven dynamic deformations. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 21, 3 (2002), 586–593.
- [CK05a] CHOI K., KO H.: Research problems in clothing simulation. *Computer-Aided Design* 37, 6 (2005), 585–592.
- [CK05b] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Transaction on Graphics* 21, 3 (2005), 604–611.
- [CLW06] CAO D. Q., LIU D., WANG C. H.-T.: Three dimensional non-linear dynamics of slender structures: Cosserat rod element approach. *International Journal of Solids and Structures* 43, 3–4 (2006), 760–783.
- [CSZ07] CHANG J., SHEPHERD D. X., ZHANG J. J.: Cosserat-beam-based dynamic response modelling. *Computer Animation and Virtual Worlds* 18, 4-5 (2007), 429–436.

- [CW05] CIRAK F., WEST M.: Decomposition contact response (DCR) for explicit finite element dynamics. *International Journal for Numerical Methods in Engineering* 64, 8 (2005), 1078–1110.
- [CX04] CHEN L., XU J.: Optimal delaunay triangulations. *J. Computational Mathematics* 22, 2 (2004), 299–308.
- [DAK04] DURIEZ C., ANDRIOT C., KHEDDAR A.: Signorini's contact model for deformable objects in haptic simulations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004), pp. 3232–3237.
- [DBS91] DEY T. K., BAJAJ C. L., SUGIHARA K.: On good triangulations in three dimensions. In *Proc. ACM symposium on Solid modeling foundations and CAD/CAM applications* (1991), pp. 431–441.
- [DDBC99] DEBUNNE G., DESBRUN M., BARR A., CANI M.-P.: Interactive multiresolution animation of deformable models. In *Computer Animation and Simulation* (1999), pp. 133–144.
- [DDCB00] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Adaptive simulation of soft bodies in real-time. In *Computer Animation* (2000), pp. 15–20.
- [DDCB01] DEBUNNE G., DESBRUN M., CANI M.-P., BARR A. H.: Dynamic real-time deformations using space and time adaptive sampling. In *Proc. SIGGRAPH* (2001), pp. 31–36.
- [DDKA06] DURIEZ C., DUBOIS F., KHEDDAR A., ANDRIOT C.: Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 12, 1 (2006), 36–47.
- [DG95] DESBRUN M., GASCUEL M.-P.: Animating soft substances with implicit surfaces. In *SIGGRAPH'95: Proc. 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 287–290.
- [Dic94] DICHMANN D. J.: *Hamiltonian Dynamics of a Spatial Elastica and the Stability of Solitary Waves*. PhD thesis, 1994. University of Maryland.
- [DMG05] DEQUIDT J., MARCHAL D., GRISONI L.: Time-critical animation of deformable solids: Collision detection and deformable objects. *Computer Animation and Virtual Worlds* 16, 3-4 (2005), 177–187.

- [DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH* (1999), pp. 317–324.
- [DMTKT93] DALDEGAN A., MAGNENAT-THALMANN N., KURIHARA T., THALMANN D.: An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum* 12, 3 (1993), 211–221.
- [Ebe04] EBERLY D.: *Game Physics*. Morgan Kaufmann Publishers, 2004.
- [EEHS00] ETZMUSS O., EBERHARDT B., HAUTH M., STRASSER W.: Collision adaptive particle systems. In *Proc. Pacific Graphics* (2000), pp. 338–347.
- [FG03] FISCHER K., GARTNER B.: The Smallest Enclosing Ball of Balls: Combinatorial Structure and Algorithms. In *Symposium on Computational Geometry* (2003), pp. 292–301.
- [FH04] FELZENSZWALB P., HUTTENLOCHER D.: Distance transforms of sampled functions. In *Cornell Computing and Information Science TR2004-1963* (2004).
- [FL01] FISHER S., LIN M. C.: Deformed distance fields for simulation of non-penetrating flexible bodies. In *Proc. Eurographic workshop on Computer animation and simulation* (2001), pp. 99–111.
- [GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Transaction on Graphics* 22, 3 (2003), 871–878.
- [GBT06] GISSLER M., BECKER M., TESCHNER M.: Local constraint methods for deformable objects. In *Third Workshop in Virtual Reality, Interactions and Physical Simulations* (2006), pp. 25–32.
- [GEW05] GEORGII J., ECHTLER F., WESTERMANN R.: Interactive simulation of deformable bodies on GPUs. In *SimVis* (2005), pp. 247–258.
- [GKS02] GRINSPUN E., KRYSL P., SCHRÖDER P.: CHARMS: a simple framework for adaptive simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 21, 3 (2002), 281–290.
- [GLM96] GOTTSCHALK S., LIN M., MANOCHA D.: OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *Proc. Siggraph* (1996), pp. 171–180.

- [GLM06] GAYLE R., LIN M., MANOCHA D.: Adaptive dynamics with efficient contact handling for articulated robots. In *Proc. of Robotics: Science and Systems* (2006).
- [GNRZ02] GUIBAS L., NGUYEN A., RUSSEL D., ZHANG L.: Collision Detection for Deforming Necklaces. In *Symposium on Computational geometry* (2002), pp. 33–42.
- [Gol81] GOLDSTEIN H.: *Classical Mechanics*, 2 ed. Addison-Wesley, 1981.
- [GOM*06] GALOPPO N., OTADUY M. A., MECKLENBURG P., GROSS M., LIN M. C.: Fast simulation of deformable models in contact using dynamic deformation textures. In *Proc. Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2006), pp. 73–82.
- [GOT*07] GALOPPO N., OTADUY M. A., TEKIN S., GROSS M., LIN M. C.: Soft articulated characters with fast contact handling. *Computer Graphics Forum* 26, 3 (2007), 243–253.
- [GS06] GRÉGOIRE M., SCHÖMER E.: Interactive simulation of one-dimensional flexible parts. In *Proc. ACM Symposium on Solid and physical modeling* (2006), pp. 95–103.
- [GTS02] GREEN S., TURKIYYAH G., STORTI D.: Subdivision-based multilevel methods for large scale engineering simulation of thin shells. In *Proc. ACM Symposium on Solid Modeling and Applications* (2002), pp. 265–272.
- [GTT89] GOURRET J.-P., THALMANN N. M., THALMANN D.: Simulation of object and human skin formations in a grasping task. *SIGGRAPH Computer Graphics* 23, 3 (1989), 21–30.
- [GW05] GEORGII J., WESTERMANN R.: A multigrid framework for real-time simulation of deformable volumes. In *Proc. Workshop On Virtual Reality Interaction and Physical Simulation* (2005), pp. 50–57.
- [Had06] HADAP S.: Oriented strands: dynamics of stiff multi-body system. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 91–100.
- [HFS03] HIROTA G., FISHER S., STATE A.: An improved finite-element contact model for anatomical simulations. In *The Visual Computer*, vol. 19. Springer, 2003, pp. 291–309.
- [HMB01] HUH S., METAXAS D., BADLER N.: Collision resolutions in cloth simulation. In *Proc. Computer Animation* (2001), pp. 122–127.

- [HPH96] HUTCHINSON D., PRESTON M., HEWITT T.: Adaptive refinement for mass/spring simulations. In *Proc. 7th Eurographics Workshop on Animation and Simulation* (1996), pp. 31–45.
- [HSO03] HAUSER K., SHEN C., O'BRIEN J.: Interactive deformation using modal analysis with constraints. In *Proc. Graphics Interface* (2003), pp. 247–255.
- [HTG04] HEIDELBERGER B., TESCHNER M., GROSS M.: Detection of collisions and self-collisions using image-space techniques. In *Proc. of WSCG* (2004), pp. 145–152.
- [HTK*04] HEIDELBERGER B., TESCHNER M., KEISER R., MUELLER M., GROSS M.: Consistent penetration depth estimation for deformable collision response. In *Proc. Vision, Modeling, Visualization* (2004), pp. 339–346.
- [Hub95] HUBBARD P. M.: Collision Detection for Interactive Graphics Applications. *IEEE Transactions on Visualization and Computer Graphics* 1, 3 (1995), 218–230.
- [HVTG08] HARMON D., VOUGA E., TAMSTORF R., GRINSPUN E.: Robust treatment of simultaneous collisions. *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2008). to appear.
- [ISS04] ITO Y., SHIH A. M., SONI B. K.: Reliable isotropic tetrahedral mesh generation based on an advancing front method. In *Proc. 13th International Meshing Roundtable* (2004), pp. 95–106.
- [ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 131–140.
- [JP99] JAMES D. L., PAI D. K.: Artdefo: accurate real time deformable objects. In *SIGGRAPH'99: Proc. of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 65–72.
- [JP02] JAMES D. L., PAI D. K.: DyRT: dynamic response textures for real time deformation simulation with graphics hardware. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 21, 3 (2002), 582–585.
- [JP04] JAMES D. L., PAI D. K.: BD-tree: output-sensitive collision detection for reduced deformable models. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 23, 3 (2004), 393–398.

- [Ju04] JU T.: Robust repair of polygonal models. *ACM Transaction on Graphics (Proc. SIGGRAPH)* (2004), 888–895.
- [Keh97] KEHRBAUM S.: *Hamiltonian Formulations of the Equilibrium Conditions Governing Elastic Rods: Qualitative Analysis and Effective Properties*. PhD thesis, 1997. University of Maryland.
- [KEP05] KAUFMAN D. M., EDMUNDS T., PAI D. K.: Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 24, 3 (2005), 946–956.
- [KFCO06] KLINGNER B. M., FELDMAN B. E., CHENTANEZ N., O'BRIEN J. F.: Fluid animation with dynamic meshes. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 820–825.
- [KHM*96] KLOSOWSKI J., HELD M., MITCHELL J., SOWIZRAL H., ZIKAN K.: Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics* 4, 1 (1996), 21–36.
- [KJM08] KALDOR J., JAMES D. L., MARSCHNER S.: Simulating knitted cloth at the yarn level. In *Proc. SIGGRAPH* (2008). to appear.
- [KMH*04] KEISER R., MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Contact handling for deformable point-based objects. In *Proc. Vision, Modeling, Visualization* (2004), pp. 339–347.
- [KPGF07] KUBIAK B., PIETRONI N., GANOVELLI F., FRATARCANGELI M.: A robust method for real-time thread simulation. In *Proc. ACM Symposium on Virtual reality software and technology* (2007), pp. 85–88.
- [KS98] KOBBELT L., SCHRÖDER P.: A multiresolution framework for variational subdivision. *ACM Transactions on Graphics* 17, 4 (1998), 209–237.
- [KZ03] KLEIN J., ZACHMANN G.: Adb-trees: Controlling the error of time-critical collision detection. In *8th International Fall Workshop Vision, Modeling, and Visualization* (2003), pp. 37–45.
- [KZ05] KAVAN L., ZARA J.: Fast Collision Detection for Skeletally Deformable Models. *Computer Graphics Forum (Proc. Eurographics)* 24, 3 (2005), 363–372.
- [LAM01] LARSSON T., AKENINE-MÖLLER T.: Collision Detection for Continuously Deforming Bodies. In *Proc. Eurographics* (2001), pp. 325–333.

- [LAM03] LARSSON T., AKENINE-MÖLLER T.: Efficient Collision Detection for Models Deformed by Morphing. *The Visual Computer* 19, 2-3 (2003), 164–174.
- [LAM05] LARSSON T., AKENINE-MÖLLER T.: A Dynamic Bounding Volume Hierarchy for Generalized Collision Detection. In *Workshop on Virtual Reality Interactions and Physical Simulations* (2005), pp. 91–100.
- [Lev66] LEVINTHAL C.: Molecular model-building by computer. *Scientific American* 214 (1966), 42–52.
- [LGCM05] LENOIR J., GRISONI L., CHAILLOU C., MESEURE P.: Adaptive resolution of 1D mechanical B-spline. In *Proc. Computer Graphics and Interactive Techniques in Australasia and South East Asia* (2005), pp. 395–403.
- [LLC*07] LU, LIN, CHOI, YI-KING, WANG, WENPING, KIM, MYUNG-SOO: Variational 3d shape segmentation for bounding volume computation. *Computer Graphics Forum* 26, 3 (2007), 329–338.
- [LMGC02] LENOIR J., MESEURE P., GRISONI L., CHAILLOU C.: Surgical thread simulation. *Modelling & Simulation for Computer-aided Medicine and Surgery* (2002), 102–107.
- [LS07] LABELLE F., SHEWCHUK J. R.: Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 57.
- [MBTF03] MOLINO N., BRIDSON R., TERAN J., FEDKIW R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *Proc. 12th International Meshing Roundtable* (2003), pp. 103–114.
- [MC94] MIRTICH B., CANNY J. F.: *Impulse-Based Dynamic Simulation*. Tech. Rep. UCB/CSD-94-815, EECS Department, University of California, 1994.
- [MDM*02] MÜLLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 49–54.
- [MFD99] M. A., F.A. P., D.E. S.: Time-stepping for three-dimensional rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering* 177 (1999), 183–197.

- [MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proc. Graphics Interface* (2004), pp. 239–246.
- [MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.
- [MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 24, 3 (2005), 471–478.
- [MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 141–151.
- [MOK95] MYSZKOWSKI K., OKUNEV O., KUNII T.: Fast collision detection between complex solids using rasterizing graphics hardware. *The Visual Computer* 11, 9 (1995), 497–512.
- [MS01] MILENKOVIC V., SCHMIDL H.: Optimization-based animation. In *Proc. Computer graphics and Interactive Techniques* (2001), pp. 37–46.
- [MT03] MÜLLER M., TESCHNER M.: Volumetric meshes for real-time medical simulations. In *Proc. BVM* (2003), pp. 279–283.
- [MW88] MOORE M., WILHELMS J.: Collision detection and response for computer animation. *Computer Graphics (Proc. Siggraph)* 22, 4 (1988), 289–298.
- [NR04] NADLER B., RUBIN M. B.: Post-buckling behavior of nonlinear elastic beams and three-dimensional frames using the theory of a cosserat point. *Mathematics and Mechanics of Solids*, 9 (2004), 369–398.
- [NT03] NOORUDDIN F. S., TURK G.: Simplification and repair of polygonal models using volumetric techniques. In *IEEE Transaction on Visualization and Computer Graphics* (2003), pp. 191–205.
- [NW99] NOCEDAL J., WRIGHT S.: *Numerical Optimization*. Springer Series in Operation Research, 1999.
- [OBH02] O'BRIEN J. F., BARGTEIL A. W., HODGINS J. K.: Graphical modeling and animation of ductile fracture. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 21, 3 (2002), 291–294.

- [OD99] O’SULLIVAN C., DINGLIANA J.: Real-time Collision Detection and Response Using Sphere-Trees. In *Spring Conference on Computer Graphics* (1999), pp. 83–92.
- [OGRG07] OTADUY M., GERMANN D., REDON S., GROSS M.: Adaptive deformations with fast tight bounds. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 181–190.
- [OH99] O’BRIEN J. F., HODGINS J. K.: Graphical modeling and animation of brittle fracture. In *SIGGRAPH ’99: Proc. of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 137–146.
- [Pai02] PAI D.: Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum (Proc. Eurographics)* 21, 3 (2002), 347–352.
- [PB81] PLATT S. M., BADLER N. I.: Animating facial expressions. *SIGGRAPH Computer Graphics* 15, 3 (1981), 245–252.
- [PCP01] PLANTE E., CANI M.-P., POULIN P.: A layered wisp model for simulating interactions inside long hair. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)* (2001), pp. 139–148.
- [PLK02] PHILLIPS J., LADD A., KAVRAKI L. E.: Simulated knot tying. In *Proc. IEEE International Conference on Robotics & Automation* (2002), pp. 841–846.
- [PPG04] PAULY M., PAI D. K., GUIBAS L. J.: Quasi-rigid objects in contact. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 109–119.
- [Pro97] PROVOT X.: Collision and self-collision handling in cloth model dedicated to design garments. *Graphics Interface 97* (1997), 177–189.
- [PW89] PENTLAND A., WILLIAMS J.: Good vibrations: Modal dynamics for graphics and animation. *SIGGRAPH Computer Graphics* 23, 3 (July 1989), 215–222.
- [QT96] QIN H., TERZOPOULOS D.: D-NURBS: A Physics-Based Framework for Geometric Design. *IEEE Transactions on Visualization and Computer Graphics* 2, 1 (1996), 85–96.
- [Qui94] QUINLAN S.: Efficient Distance Computation between Non-Convex Objects. In *IEEE International Conference on Robotics and Automation* (1994), pp. 3324–3329.

- [RCT91] ROSENBLUM R. E., CARLSON W. E., TRIPP E.: Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation* 2, 4 (1991), 141–148.
- [RGL05] REDON S., GALOPPO N., LIN M. C.: Adaptive dynamics of articulated bodies. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 24, 3 (2005), 936–945.
- [RJ07] RIVERS A. R., JAMES D. L.: FastLSM: fast lattice shape matching for robust real-time deformation. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 26, 3 (2007), 82.
- [RKC02] REDON S., KHEDDAR A., COQUILLART S.: Gauss'least constraints principle and rigid body simulations. *IEEE International Conference on Robotics and Automation 1* (2002), 517–522.
- [RNG00] REMION Y., NOURRIT J.-M., GILLARD D.: A dynamic animation engine for generic spline objects. *Journal of Visualization and Computer Animation* 11, 1 (2000), 17–26.
- [RO00] RADOVITZKY R., ORTIZ M.: Tetrahedral mesh generation based in node insertion in crystal lattice arrangements and advancing-front delaunay triangulation. *Comput. Meth. in Appl. Mech. and Eng.* 187 (2000), 543–569.
- [Rub00] RUBIN M.: *Cosserat Theories: Shells, Rods and Points*. Kluwer Academic Publishers, 2000.
- [SBT07a] SPILLMANN J., BECKER M., TESCHNER M.: Efficient updates of bounding sphere hierarchies for geometrically deformable models. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 101–108.
- [SBT07b] SPILLMANN J., BECKER M., TESCHNER M.: Non-iterative computation of contact forces for deformable objects. *Journal of WSCG* 15, 1-3 (2007), 33–40.
- [Sch97a] SCHOBERL J.: Netgen - an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Visual. Sci.* (1997), 41–52.
- [Sch97b] SCHWARZ H. R.: *Numerische Mathematik*. Teubner, 1997.
- [SFR90] SIMO J. C., FOX D. D., RIFAI M. S.: On a stress resultant geometrically exact shell model. part III: Computational aspects of the nonlinear theory. *Computer Methods in Applied Mechanics and Engineering* 79 (1990), 21–70.

- [She98] SHEWCHUK J. R.: Tetrahedral mesh generation by delaunay refinement. In *Proc. 14th Annual Symposium on Computational Geometry* (1998), pp. 86–95.
- [She02] SHEWCHUK J. R.: What is a good linear element? - interpolation, conditioning, and quality measures. *Proc. 11th International Meshing Roundtable* (2002), 115–126.
- [Sho85] SHOEMAKE K.: Animating rotation with quaternion curves. In *SIGGRAPH '85: Proc. of the 12th annual conference on Computer graphics and interactive techniques* (1985), pp. 245–254.
- [SLF08] SELLE A., LENTINE M. G., FEDKIW R.: A mass spring model for hair simulation. *ACM Transaction on Graphics (Proc. SIGGRAPH)* (2008). to appear.
- [SM06] SCHWAB A. L., MEIJAARD J. P.: How to draw euler angles and utilize euler parameters. In *Proc. IDETC/CIE 2006, ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* (2006).
- [SOG07] STEINEMANN D., OTADUY M. A., GROSS M.: Efficient bounds for point-based animations. In *Proc. IEEE/Eurographics Symposium on Point-Based Graphics* (2007), pp. 57–64.
- [SOG08] STEINEMANN D., OTADUY M., GROSS M.: Fast adaptive shape matching deformations. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008). to appear.
- [SOS05] SHEN C., O'BRIEN J. F., SHEWCHUK J. R.: Interpolating and approximating implicit surfaces from polygon soup. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 23, 3 (2005), 896–904.
- [SSIF07] SIFAKIS E., SHINAR T., IRVING G., FEDKIW R.: Hybrid simulation of deformable solids. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), pp. 81–90.
- [ST05] SPILLMANN J., TESCHNER M.: Contact surface computation for coarsely sampled deformable objects. In *Proc. Vision, Modeling, Visualization* (2005), pp. 289–296.
- [ST07] SPILLMANN J., TESCHNER M.: CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 63–72.

- [ST08] SPILLMANN J., TESCHNER M.: An Adaptive Contact Model for the Robust Simulation of Knots. *Computer Graphics Forum (Proc. Eurographics) 27*, 2 (2008), 497–506.
- [ST09] SPILLMANN J., TESCHNER M.: Cosserat Nets. *IEEE Transactions on Visualization and Computer Graphics* (2009). to appear.
- [Sta97] STAM J.: Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum 16*, 4 (1997), 159–164.
- [Stu64] STUELPNAGEL J.: On the parameterization of the three-dimensional rotation group. *SIAM*, 6 (1964), 422–430.
- [SWT06] SPILLMANN J., WAGNER M., TESCHNER M.: Robust tetrahedral meshing of triangle soups. In *Proc. Vision, Modeling, Visualization VMV'06* (2006), pp. 9–16.
- [TGAB07] THEETTEN A., GRISONI L., ANDRIOT C., BARSKY B.: *Geometrically exact dynamic splines*. Tech. rep., INRIA, 2007.
- [TGDM07] THEETTEN A., GRISONI L., DURIEZ C., MERLHIOT X.: Quasi-dynamic splines. In *Proc. ACM symposium on Solid and physical modeling* (2007), pp. 409–414.
- [THM*03] TESCHNER M., HEIDELBERGER B., MÜLLER M., POMERANTES D., GROSS M. H.: Optimized spatial hashing for collision detection of deformable objects. In *Proc. Vision, Modeling, Visualization* (2003), pp. 47–54.
- [THMG04] TESCHNER M., HEIDELBERGER B., MULLER M., GROSS M.: A versatile and robust model for geometrically complex deformable solids. In *Proc. Computer Graphics International* (2004), pp. 312–319.
- [TKH*05] TESCHNER M., KIMMERLE S., HEIDELBERGER B., ZACHMANN G., RAGHUPATHI L., FUHRMANN A., CANI M., FAURE F., MAGNENAT-THALMANN N., STRASSER W.: Collision Detection for Deformable Objects. *Computer Graphics Forum 24*, 1 (2005), 61–81.
- [TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *Proc. SIGGRAPH* (1994), pp. 311–318.
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH) 21*, 4 (1987), 205–214.

- [TPF89] TERZOPOULOS D., PLATT J., , FLEISCHER K.: Heating and melting deformable models (from goop to glop). In *Proc. Graphics Interface* (1989), pp. 219–226.
- [Tur90] TURK G.: *Interactive collision detection for molecular graphics*. Tech. Rep. TR90-014, University of North Carolina at Chapel Hill, 1990.
- [TWS06] THOMASZEWSKI B., WACKER M., STRASSER W.: A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proc. ACM SIGGRAPH/Eurographics symposium on Computer animation* (2006), pp. 107–116.
- [Van97] VAN DEN BERGEN G.: Efficient Collision Detection of Complex Deformable Models Using AABB Trees. *Journal of Graphics Tools* 2, 4 (1997), 1–13.
- [Van04] VAN DEN BERGEN G.: *Collision detection in interactive 3d environments*. Morgan Kaufmann publishers, 2004.
- [VCT95] VOLINO P., COURCHESNE M., THALMANN N. M.: Versatile and efficient techniques for simulating cloth and other deformable objects. In *SIGGRAPH '95: Proc. of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 137–144.
- [vdHNGT03] VAN DER HEIJDEN G., NEUKIRCH S., GOSS V., THOMPSON J.: Instability and self-contact phenomena in the writhing of clamped rods. *International Journal of Mechanical Sciences* 45, 1 (2003), 161–196.
- [VMT00] VOLINO P., MAGNENAT-THALMANN N.: Accurate collision response on polygonal meshes. In *Proc. Computer Animation* (2000), pp. 154–163.
- [VMT06] VOLINO P., MAGNENAT-THALMANN N.: Resolving surface collisions through intersection contour minimization. *ACM Transaction on Graphics (Proc. SIGGRAPH)* 25, 3 (2006), 1154–1159.
- [VTG97] VELHO L., TERZOPOULOS D., GOMES J.: Implicit manifolds, triangulations and dynamics. *Journal of Neural, Parallel and Scientific Computations. Special Issue in Computer Aided Geometric Design* (1997), 103–120.
- [WBD*05] WANG F., BURDET E., DHANIK A., POSTON T., TEO C.: Dynamic thread for real-time knot-tying. In *Proc. First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (2005), pp. 507–508.

- [WBG07] WICKE M., BOTSCH M., GROSS M.: A finite element method on convex polyhedra. *Computer Graphics Forum (Proc. Eurographics)* 26, 3 (2007), 355–364.
- [WDGT01] WU X., DOWNES M., GOKTEKIN T., TENDICK F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Computer Graphics Forum* 20, 10 (2001), 349–358.
- [WH04] WAKAMATSU H., HIRAI S.: Static modeling of linear object deformation based on differential geometry. *I. J. Robotic Res.* 23, 3 (2004), 293–311.
- [Wit01] WITKIN A.: Physically based modeling: Constrained dynamics, 2001. ACM SIGGRAPH course notes.
- [WT08] WOJTAN C., TURK G.: Fast viscoelastic behavior with thin features. *ACM Transaction on Graphics (Proc. SIGGRAPH)* (2008). to appear.
- [WTF06] WEINSTEIN R., TERAN J., FEDKIW R.: Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics* 12, 3 (2006), 365–374.
- [ZS*99] ZORIN D., SCHRÖDER P., ET AL.: Subdivision for modeling and animation. *SIGGRAPH 99 Course Notes 2* (1999).

Curriculum vitae

Personal data

Name	Jonas Spillmann
Date of birth	June 24th, 1978
Place of birth	Zürich, Switzerland

Education

2005 – 2008	Ph.D. candidate and teaching assistant Computer Graphics, University of Freiburg
2004	Diploma in computer science (Dipl. Inf. Ing. ETH)
1998 – 2004	Studies in computer science Swiss Federal Institute of Technology, Switzerland
1993 – 1998	Mathematical and Scientific High School MNG Limmattal, Urdorf, Switzerland

Publications

- [ST09] SPILLMANN J., TESCHNER M.: Cosserat Nets. *IEEE Transactions on Visualization and Computer Graphics*, to appear.
- [ST08] SPILLMANN J., TESCHNER M.: An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum* 27, 2 (2008), pp. 497–506.
- [ST07] SPILLMANN J., TESCHNER M.: CoRDE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 63–72.
- [SBT07b] SPILLMANN J., BECKER M., TESCHNER M.: Non-iterative computation of contact forces for deformable objects. *Journal of WSCG* 15, 1-3 (2007), 33–40.
- [SBT07a] SPILLMANN J., BECKER M., TESCHNER M.: Efficient updates of bounding sphere hierarchies for geometrically deformable models. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 101–108.
- [SWT06] SPILLMANN J., WAGNER M., TESCHNER M.: Robust tetrahedral meshing for triangle soups. In *Proc. Vision, Modeling, Visualization* (2006), pp. 9–16.
- [ST05] SPILLMANN J., TESCHNER M.: Contact Surface Computation for coarsely sampled deformable objects. In *Proc. Vision, Modeling, Visualization* (2005), pp. 289–296.