EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Implicit Contact Handling for Deformable Objects

Miguel A. Otaduy[1], Rasmus Tamstorf[2], Denis Steinemann[3], and Markus Gross[3]

[1]URJC Madrid, Spain
[2]Walt Disney Animation Studios, USA
[3]ETH Zurich, Switzerland

Abstract

We present an algorithm for robust and efficient contact handling of deformable objects. By being aware of the internal dynamics of the colliding objects, our algorithm provides smooth rolling and sliding, stable stacking, robust impact handling, and seamless coupling of heterogeneous objects, all in a unified manner. We achieve dynamics-awareness through a constrained dynamics formulation with implicit complementarity constraints, and we present two major contributions that enable an efficient solution of the constrained dynamics problem: a time stepping algorithm that robustly ensures non-penetration and progressively refines the formulation of constrained dynamics, and a new solver for large mixed linear complementarity problems, based on iterative constraint anticipation. We show the application of our algorithm in challenging scenarios such as multi-layered cloth moving at high velocities, or colliding deformable solids simulated with large time steps.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Animation—

1. Introduction

Contact is a ubiquitous phenomenon around us, and its importance to computer graphics is testified by a large amount of previous work on the design of efficient algorithms [Bar94, VMT97, BW98, BFA02, GBF03, KEP05, HVTG08]. When handled robustly, contact can lead to vivid effects that dramatically enhance the richness of animations.

We propose a method for contact handling that robustly supports non-penetration, dynamic and static friction, and stacking, even under large time steps. Our method is general and applicable to a large family of objects, ranging from cloth-like thin shells, to rigid bodies or volumetric deformable bodies. We found our method on an implicit collision response formulation based on complementarity constraints and Lagrange multipliers. This implicit formulation, described in Section 3, enjoys great robustness and has been successfully applied to rigid bodies [Bar96, Erl07], but suffers a complexity explosion when extended to deformable objects with many degrees of freedom and many contacts, such as the ones shown in our examples.

We achieve efficient implicit contact handling for deformable objects through two major contributions. In Section 4, we introduce a time-stepping algorithm for geometrically and physically robust handling of non-penetration constraints, based on progressive *constraint manifold refinement* (CMR). And in Section 5, we propose a solver for mixed linear complementarity problems that features a novel concept of *iterative constraint anticipation* (ICA).

In Section 6 we discuss the effect of our contributions on examples such as multi-layered cloth moving at high velocities (see Fig. 1), solid deformable objects simulated with large time steps yet robust contact (see Fig. 7), or the combined handling of diverse deformation models (see Fig. 4). For typical deformation models, our solver has linear cost per iteration and converges in few iterations. It is particularly well suited for large constrained problems, and we demonstrate that it efficiently handles thousands of degrees of freedom and contacts. In our examples, we have combined CMR and ICA, but the two contributions are orthogonal, and they can also be combined with other time-stepping or collision response algorithms. We continue now with a discussion of related work before describing our contributions in detail.

2. Related Work

The simulation of contact among deformable objects covers deformation modeling, collision detection, and contact han-

Figure 1: *A flamenco dancer performs a few steps and a 360 turn (in less than 1sec). Her 6-layered dress (16800 vertices) undergoes challenging contact, but is robustly resolved by our dynamics-aware contact handling.*

dling. We refer readers to recent surveys on collision detection for deformable objects [TKH*05], general deformation modeling [NMK*05], or modeling of cloth [HB00, CK05], for more details on those topics. Our work deals with contact handling or collision response, which has been addressed through many different approaches. They can be classified based on the nature of the response (i.e., forces, impulses, or geometric projection), their local or global effect, or the sequential or simultaneous processing of contacts.

One approach for collision response is the use of elastic repulsive forces (or penalty method) [MW88]. The major problem of the penalty method is that non-penetration robustness and physical robustness require different parameter choices. To robustly avoid non-penetration, repulsive forces must be very stiff or even nonlinear. To robustly avoid non-physical jitter effects, forces must be continuous. However, to the best of our knowledge, there is no method that guarantees continuous repulsive forces in the general case, hence high stiffness can lead to disturbing jitter artifacts. The penalty method is, nevertheless, often the method of choice when very high performance is a major goal [BJ07].

Contact handling is particularly challenging in cloth simulation, as the non-orientability of the surface calls for strict enforcement of non-penetration. Volino and Magnenat-Thalmann have long proposed techniques for efficient simulation of cloth and garments. Their work on collision response includes a geometric projection method for resolving positions, velocities, and/or accelerations [VMT97], which they later refined to deal with competing constraints and with an efficient solver [VMT00]. Baraff and Witkin [BW98] enabled the use of large time steps in cloth simulation through implicit methods, and handled cloth-cloth collisions through repulsive forces. Provot introduced methods for dealing with competing constraints through rigid impact zones [Pro95], or through iterative application of impulses [Pro97]. Bridson et al. [BFA02] introduced a bullet-proof technique for ensuring non-penetration in cloth (or other deformation models) by combining repulsive response, geometric response, and rigid impact zones if needed. Very recently, Harmon et al. [HVTG08] have eliminated excessive sticking effects produced by rigid impact zones. Also very recently, Sifakis et al. [SMT08] have introduced a contact handling technique based on intra-contact volume preservation.

Our work is related to methods that model collision response as a constrained optimization. The approach has gained popularity for rigid body simulation over the last decade [Bar94, RKC02, KEP05, Erl07], as it provides accurate handling of friction and stacking. It has also seen application to deformable objects since the work of Baraff and Witkin [BW92]. The simulation of contact deformations has received large attention in the field of computational mechanics, and we refer to the book of Wriggers [Wri02] for a comprehensive treatment. Similar to many others before, our formulation of contact deformations can be classified in the family of linear complementarity problems (LCP) [CPS92]. The formulation of the LCP using an implicit time-stepping method robustly guarantees the enforcement of the constraints at the end of the time step [ST96].

There are two major differences between iterative constraint handling methods [BFA02, GBF03, MHHR06], and implicit LCP formulations. One difference is the sequential vs. simultaneous formulation of all constraints. A simultaneous formulation guarantees that the solution to one constraint will not violate others. Another difference is the local vs. global effect of the response. For deformable objects, iterative constraint handling methods typically apply a response locally on the colliding nodes [BFA02, MHHR06, SBT07], while implicit methods act globally. Local response works well with small time steps, but, as highlighted in Fig. 2, it may produce spurious energy growth due to excessive local deformation. Bridson et al. [BFA02] alleviated this effect by adding cloth relaxation steps to collision response. Recently, Shinar et al. [SSF08] have combined iterative constaint handling methods for deformable and rigid bodies.

The formulation of a contact LCP requires the computation of all pair-wise contact force effects, a process re-

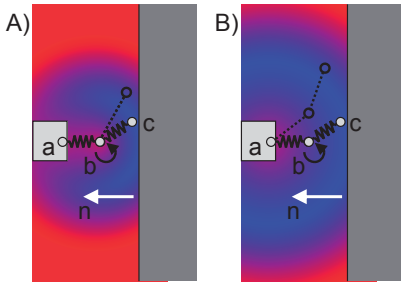

Figure 2: ***Local vs. Global Response.*** *A flexible structure with linear springs* $(\mathbf{a},\mathbf{b})$ *and* $(\mathbf{b},\mathbf{c})$*, and a torsional spring at* $\mathbf{b}$*, collides against a wall. Elastic energy varies with the position of* $\mathbf{c}$ *('blue' is low energy, and 'red' is high). Left (A): local geometric projection moves* $\mathbf{c}$ *in the direction of* $\mathbf{n}$ *keeping* $\mathbf{b}$ *and* $\mathbf{a}$ *fixed, and clearly adds excessive energy. Right (B): a global response method also moves* $\mathbf{b}$ *to an optimal location, producing a remarkably lower elastic energy.*

ferred to as constraint anticipation [Bar96] or computation of the Delassus operator [DDKA06]. With rigid bodies, constraint anticipation is a popular approach because it requires the solution of small linear systems. But with deformable bodies with many degrees of freedom and implicit constraints it becomes excessively expensive. Some have used linear [PPG04] or global co-rotational deformation models [DDKA06] to leverage precomputations. Saupin et al. [SDCG08] approximate and accelerate the formulation of the Delassus operator using compliance warping. Raghupathi and Faure [RF06] maintain an active constraint set and exploit linear system solvers, but this approach may lead to excessive sticking, in particular with large time steps. Very recently, Kaufman et al. [KSJP08] have introduced an approach for rigid bodies and reduced deformation models that formulates and solves non-penetration and frictional LCPs in a staggered manner. In Section 5 we propose a novel efficient solver for large constrained deformation problems.

3. Constrained Dynamics Formulation

We now describe the frictional constrained dynamics formulation on which our contact handling algorithm operates.

3.1. Dynamics and Numerical Integration

For the presentation of our algorithm, we comprise all degrees of freedom of the simulation in one state vector $\mathbf{q}$. We also comprise all velocities in one vector $\mathbf{v}$. In this paper we do not propose any new method for modeling deformations, but we assume that dynamic deformations can be discretized with ODEs of the form:

$$\begin{aligned}\mathbf{M}\dot{\mathbf{v}} &= \mathbf{F},\\ \dot{\mathbf{q}} &= \mathbf{G}\mathbf{v},\end{aligned} \tag{1}$$

where $\mathbf{M}$ denotes the mass matrix, $\mathbf{F}$ is the (generally nonlinear) force vector, and $\mathbf{G}$ relates the velocity vector to the derivative of the state vector. Eq. (1) is very general, and it applies to, e.g., rigid bodies, mass-spring models, discrete shells, or FE elastic models. In our examples, we have used a mass-spring model similar to the one used by Provot [Pro95] or Bridson et al. [BFA02] for cloth, and the linear co-rotational FE model of Müller and Gross [MG04] for volumetric deformable objects.

We consider time discretization methods that, given a state $(\mathbf{q}_0,\mathbf{v}_0)$ at the beginning of a time step, yield a linear velocity update rule:

$$\mathbf{A}\mathbf{v} = \mathbf{b}. \tag{2}$$

This assumption is applicable to each Newton iteration of implicit solvers such as backward Euler or Newmark, and to each step of Runge-Kutta solvers. For the examples in this paper, and as done by [BW98], we have used a backward Euler scheme with linear approximation of forces (equivalent to doing one Newton iteration of the nonlinear solve), and assuming a constant mass matrix per time step. Then, the linear velocity update is defined by:

$$\begin{aligned}\mathbf{A} &= \mathbf{M} - \Delta t\frac{\partial \mathbf{F}}{\partial \mathbf{v}} - \Delta t^2\mathbf{G}\frac{\partial \mathbf{F}}{\partial \mathbf{q}},\\ \mathbf{b} &= \Delta t\mathbf{F}(\mathbf{q}_0,\mathbf{v}_0) + \left(\mathbf{M} - \Delta t\frac{\partial \mathbf{F}}{\partial \mathbf{v}}\right)\mathbf{v}_0.\end{aligned} \tag{3}$$

3.2. Non-Penetration Constraints

The set of object configurations $\mathbf{q}$ free of contact can be limited by a constraint manifold Γ in the high-dimensional configuration space. Collision detection locally samples this constraint manifold. More specifically, grouping all contact points in one vector $\mathbf{p}$, the free space defined by the constraint manifold Γ can be approximated by a set of algebraic inequalities $\mathbf{g}(\mathbf{p}) \geq \mathbf{0}$. Given a pair of colliding points $\mathbf{p}_a$ and $\mathbf{p}_b$, and contact normal $\mathbf{n}$, an individual non-penetration constraint can be written as $\mathbf{n}^T(\mathbf{p}_a - \mathbf{p}_b) \geq 0$. We identify vertex-face and edge-edge contact pairs in two ways: (i) those pairs closer than a distance tolerance δ at the beginning of a time step, and (ii) those that cross each other between two possible simulation states. We have implemented the continuous collision detection test of Provot [Pro97], after pruning distant pairs using axis-aligned bounding boxes and spatial hashing [THM*03]. In practice, we enforce contacts to be separated by a safety distance δ.

In order to enforce non-penetration at the end of the time step, we formulate the constraints implicitly. Specifically, we propose a semi-implicit formulation of contact constraints linearized as:

$$\mathbf{g}(\mathbf{p}) = \mathbf{g}_0 + \frac{\partial \mathbf{g}}{\partial \mathbf{p}}(\mathbf{p} - \mathbf{p}_0) \geq \mathbf{0}, \tag{4}$$

with the rows of the Jacobian $\frac{\partial \mathbf{g}}{\partial \mathbf{p}}$ formed by the contact normals $\mathbf{n}$ at the time of impact, and $\mathbf{g}_0 = \mathbf{g}(\mathbf{p}_0)$.

Accounting for the numerical integration method (implicit Euler in our case), we can transform the implicit position constraints into velocity constraints. To do this, we first express the position update of contact points in terms of their velocities, $\mathbf{p} = \mathbf{p}_0 + \Delta t \dot{\mathbf{p}}$. The velocities of contact points can in turn be expressed in terms of the state velocity vector, as $\dot{\mathbf{p}} = \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}} \mathbf{v}$. In our examples, the rows of the Jacobian $\frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}}$ constitute triangle barycentric weights (for cloth) or tetrahedron barycentric weights (for volumetric FEM). To summarize, the semi-implicit position constraints from Eq. (4) are transformed into velocity constraints of the form:

$$\mathbf{J}\mathbf{v} \geq -\frac{1}{\Delta t}\mathbf{g}_0, \tag{5}$$

with $\mathbf{J} = \frac{\partial \mathbf{g}}{\partial \mathbf{p}} \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}}$ the generalized constraint normal.

3.3. Mixed Linear Complementarity Problem

We model collision response with the method of Lagrange multipliers, and express contact forces as $\mathbf{J}^T\lambda$. Additionally, we employ Signorini's contact model [DDKA06], which imposes non-sticking constraints $\lambda \geq \mathbf{0}$ on the contact forces, and the complementarity condition $\mathbf{0} \leq \lambda \perp \mathbf{g}(\mathbf{p}) \geq 0$. Complementarity here means that contact points cannot push ($\lambda > \mathbf{0}$) and be distant ($\mathbf{g}(\mathbf{p}) > \mathbf{0}$) at the same time.

The resulting system of equations is a mixed linear complementarity problem (MLCP, called 'mixed' as it combines linear equalities and inequalities). Given unconstrained velocities $\mathbf{v}^*$, the MLCP that defines the constrained velocities $\mathbf{v}$ can be expressed as:

$$\begin{aligned} &\mathbf{A}\Delta\mathbf{v} = \mathbf{J}^T\lambda, \quad \Delta\mathbf{v} = \mathbf{v} - \mathbf{v}^*, \\ &\mathbf{0} \leq \lambda \;\perp\; \mathbf{J}\Delta\mathbf{v} \geq -\frac{1}{\Delta t}\mathbf{g}_0 - \mathbf{J}\mathbf{v}^*. \end{aligned} \tag{6}$$

The MLCP is equivalent to a quadratic program [CPS92] where the objective function to be minimized is $\frac{1}{2}\Delta\mathbf{v}^T\mathbf{A}\Delta\mathbf{v}$, subject to the contact constraints from Eq. (5). The inclusion of $\mathbf{A}$ in the distance metric of the objective function ensures that our algorithm is aware of the discretization method and the internal dynamics of the deforming objects.

3.4. Friction

We model friction with a pyramid approximation of Coulomb's friction cone, which allows us to naturally incorporate it into the constrained optimization formulation described above. We have opted for a 4-sided pyramid, but more complex approximations are possible if higher accuracy is needed. Given a contact with constraint $\mathbf{n}^T(\mathbf{p}_a - \mathbf{p}_b) \geq 0$, we first compute the tangent unconstrained relative velocity $\mathbf{v}_\mathbf{t}^* = (\mathbf{I} - \mathbf{n}\mathbf{n}^T)(\mathbf{v}_a^* - \mathbf{v}_b^*)$. We then align the 4-sided friction pyramid with this velocity, which amounts to computing unit tangent vectors $\mathbf{t}_1 = \mathbf{v}_\mathbf{t}^*/\|\mathbf{v}_\mathbf{t}^*\|$ and $\mathbf{t}_2 = \mathbf{n} \times \mathbf{t}_1$. And we add friction forces to the dynamics equation in Eq. (6) using Lagrange multipliers, which yields forces $\gamma_1\mathbf{t}_1$ and $\gamma_2\mathbf{t}_2$ for each contact. In vector form, these friction forces can be expressed as $\mathbf{H}^T\gamma$. The 4-sided pyramid aligned with the unconstrained velocity has served as a good approximation in our examples.

We handle friction in the constrained dynamics problem by augmenting the MLCP from Eq. (6) in the following way. Given a friction coefficient μ, we express the following complementary constraints for each contact: a linearized Coulomb cone constraint $\|\gamma_1\| \leq \mu\lambda$, and a velocity constraint $v_{\mathbf{t}_1} \geq 0$ that prevents friction from reversing motion (and similarly for γ_2). Altogether, the problem from Eq. (6) turns into the following nonlinear complementarity problem:

$$\begin{aligned} &\mathbf{A}\Delta\mathbf{v} = \mathbf{J}^T\lambda + \mathbf{H}^T\gamma, \\ &\mathbf{0} \leq \lambda \;\perp\; \mathbf{J}\Delta\mathbf{v} \geq -\frac{1}{\Delta t}\mathbf{g}_0 - \mathbf{J}\mathbf{v}^*, \\ &\|\gamma\| \leq \mu\lambda \;\perp\; \mathbf{H}\Delta\mathbf{v} \geq -\mathbf{H}\mathbf{v}^*. \end{aligned} \tag{7}$$

In Section 5.4 we will describe our solution to frictional constrained dynamics, which is based on a decoupling of the non-penetration and frictional problems.

4. Constraint Manifold Refinement

The solution to our constrained dynamics problem alone does not guarantee a penetration-free state at the end of a time step. There are two possible reasons: the linearization of the contact constraints, and the fact that the collision response induced by some constraints $\mathbf{g}$ may in turn violate other constraints that were not accounted for. In this section, we describe the first of our major technical contributions, a time-stepping algorithm based on constraint manifold refinement that will guarantee a collision-free state. We also give an overview of how other different novel aspects of our work fit into the overall algorithm.

We assume that a time step starts with a collision-free state, $(\mathbf{q}_0, \mathbf{v}_0)$. Then, the dynamics described by the linearized discrete system $\mathbf{A}$ from Eq. (2) would yield an unconstrained state $(\mathbf{q}^*, \mathbf{v}^*)$. However, we aim for a state $(\mathbf{q}, \mathbf{v})$ that satifies the constraint manifold Γ, and minimizes the 'distance' to the unconstrained state subject to the metric $\mathbf{A}$.

The constraint manifold Γ that determines valid dynamics solutions is nonlinear and highly dimensional. As an alternative, we propose the creation of a local conservative bound of Γ, through a process of progressive constraint manifold refinement (CMR). CMR intertwines steps of collision response computation and continuous collision detection for reaching a collision-free state at the end of the time step. An example application of the CMR algorithm is schematically depicted in Fig. 3. Along with the initial and unconstrained states, CMR maintains a set of linear constraints $\mathbf{g}$ (described in Section 3.2) that locally approximate Γ.

At each iteration of CMR, we perform a frictional constrained dynamics update as described in Section 3.4. This results in a tentative simulation state $(\mathbf{q}, \mathbf{v})$ that satisfies the

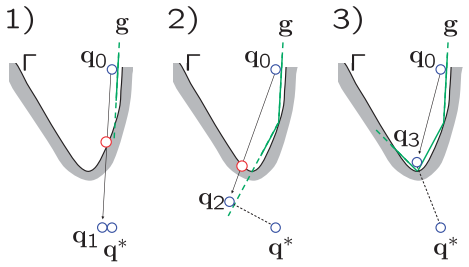

Figure 3: ***CMR Time Stepping.*** *A point at* $\mathbf{q}_0$ *would move in an unconstrained setting to* $\mathbf{q}^*$*. CMR finds a location* $\mathbf{q}_3$*, as close as possible to* $\mathbf{q}^*$*, subject to constraints* $\mathbf{g}$*, which form a local piecewise linear approximation of the constraint manifold* Γ*. In each iteration of CMR, the point moves to a tentative goal position using the current approximation* $\mathbf{g}$*, and collision detection is executed for refining* $\mathbf{g}$ *if necessary.*

constraints $\mathbf{g}$. Unfortunately, this state is not guaranteed to be collision-free, due to the local approximation of Γ. The second step in CMR is to perform continuous collision detection that checks for contacts in the linear motion between $\mathbf{q}_0$ and $\mathbf{q}$. If no contacts appear, the tentative state is valid and we can start a new time step. If contacts appear, we refine the approximation of the constraint manifold Γ by augmenting the set of constraints $\mathbf{g}$, and perform a new iteration of CMR. For each new contact, we add an implicit linearized constraint, whose formulation is detailed in Section 3.2. Our constrained dynamics formulation naturally handles multiple constraints per degree of freedom of the system.

Given initial state $(\mathbf{q}_0, \mathbf{v}_0)$, and using one Newton iteration of implicit Euler for the numerical integration, the full algorithm per time step proceeds as follows:

1. Linearize and discretize the dynamics, i.e., compute $(\mathbf{A}, \mathbf{G}, \mathbf{b})$, at $(\mathbf{q}_0, \mathbf{v}_0)$, as described in Section 3.1.
2. Solve for unconstrained velocities in $\mathbf{A}\mathbf{v}^* = \mathbf{b}$.
3. Execute tolerance-based collision detection at $\mathbf{q}_0$ as described in Section 3.2, and initialize the constraint set $\mathbf{g}$ with newly-found constraints $\mathbf{g}^*$.
4. Loop CMR while $\mathbf{g}^* \neq \emptyset$, at most 5 times:
   - 4.1. Compute collision response $\Delta\mathbf{v}$ based on $(\mathbf{q}_0, \mathbf{v}^*, \mathbf{g})$, as described in Section 5.
   - 4.2. Compute tentative $\mathbf{v} = \mathbf{v}^* + \Delta\mathbf{v}$ and $\mathbf{q} = \mathbf{q}_0 + \Delta t\mathbf{G}\mathbf{v}$.
   - 4.3. Execute continuous collision detection between $\mathbf{q}_0$ and $\mathbf{q}$ as described in Section 3.2.
   - 4.4. Augment the constraint set $\mathbf{g}$ with newly found constraints $\mathbf{g}^*$.

The convergence of CMR depends on the quality of the linear approximation of the constraint manifold Γ, which depends in turn on the velocity of the motion, the curvature of Γ, and the size of the time step. However, we found that, for reasonable time steps, CMR finds a safe constrained state after 2 or 3 iterations in 90% of the cases (as reported in Section 6). We also found that, after 5 iterations, halving the time step and making a new attempt with CMR turns out to be more efficient. Hence, in our examples, we allowed the time step to be halved at most twice, and all collisions were resolved. If necessary, one could add a bullet-proof step as rigid impact zones [Pro95, BFA02] or the improved method of Harmon et al. [HVTG08].

For better performance, we initialize the constraint set $\mathbf{g}$ by doing a collision detection test that returns all primitive pairs closer than the tolerance. This is particularly useful when the scene contains stacked objects, such as the multi-layered dress or the stacked letters from Fig. 7, as it dramatically reduces the number of contacts returned otherwise by the first continuous collision detection step.

Intertwining steps of collision detection and response is a common strategy in contact handling [BFA02, HVTG08], but CMR exhibits notable differences that provide higher robustness at large time steps. Recall from Section 3.2 that our derivation starts with a position-level constrained formulation, and we turn this nonlinear constrained problem into a velocity-level LCP formulation by incorporating the time integration method. This is different from other velocity-level LCP formulations that directly formulate non-approaching velocity constraints. With our formulation, the state at the end of each iteration of CMR satisfies the local linear approximation of the constraint manifold Γ. With other velocity-level formulations, one still needs to evolve positions, which, under large time steps, may lead to artifacts such as visible gaps between objects. In practice, we have also experienced a decrease in the number of collision detection iterations when using position constraints.

5. Nested Relaxation Solver for MLCPs

We present in this section our solver for constrained dynamics problems. It is founded on a novel technique of iterative constraint anticipation (ICA), and employs two nested relaxation loops. First, we discuss how MLCPs are typically solved with the technique of constraint anticipation. We then describe the concept of ICA and formulate the two relaxation loops for the frictionless case. Next, we include the friction solve, which is based on decoupling the formulation of normal and tangential response. We conclude with an outline of the complete algorithm and its discussion.

5.1. Constraint Anticipation

The MLCP from Eq. (6) can be converted into a regular LCP by Schur-complement computation:

$$\mathbf{0} \leq \lambda \quad \perp \quad \mathbf{B}\lambda \geq \mathbf{c}, \tag{8}$$

$$\text{with } \mathbf{B} = \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T \text{ the Schur complement of } \mathbf{A},$$

$$\text{and } \mathbf{c} = -\frac{1}{\Delta t}\mathbf{g}_0 - \mathbf{J}\mathbf{v}^*.$$

Figure 4: ***Unified Contact Solver.*** *Our constrained dynamics approach naturally supports diverse deformation models in a unified solver. In the figure, rigid bodies are yellow, linear co-rotational FEM models are orange, and mass-spring cloth is red. The left and middle images show the robust handling of cloth compressed between rigid and deformable solids.*

Baraff [Bar96] named this conversion to an LCP *constraint anticipation*, since the computation of $\mathbf{A}^{-1}\mathbf{J}$ is equivalent to solving for the independent collision response produced by a unit force applied on each constraint. Constraint anticipation is efficient for rigid bodies [Bar96, Erl07], as $\mathbf{A}$ is block-diagonal and can be easily inverted. However, it is very inefficient for general deformable bodies with implicit constraints, as it requires the solution of one large linear system per constraint. Moreover, with rigid bodies $\mathbf{B}$ is sparse, while with deformable bodies and implicit constraints it is dense and large, thereby making the solution of the LCP very expensive as well.

5.2. Iterative Constraint Anticipation

In the constraint anticipation method discussed above, the computation of Lagrange multipliers λ and collision response $\Delta\mathbf{v}$ are decoupled into two separate steps. We propose instead an algorithm where both λ and $\Delta\mathbf{v}$ are solved in tandem. Our algorithm exploits as well the concept of constraint anticipation, but instead of fully anticipating at once the effect of the constraints as in Eq. (8), it progressively anticipates their effect in an iterative manner. In essence, given the collision response $\Delta\mathbf{v}(i-1)$ from a certain iteration, it transforms the MLCP into an LCP through constraint anticipation, computes Lagrange multipliers $\lambda(i)$ for a new iteration, and then refines the collision response $\Delta\mathbf{v}(i)$.

Let us assume that the Lagrange multipliers $\lambda(i)$ are known at the current iteration, and we aim to compute the collision response. One approach for doing this is to solve for $\Delta\mathbf{v}$ in $\mathbf{A}\Delta\mathbf{v} = \mathbf{J}^T\lambda$ through iterative relaxation (in our case, block-Jacobi relaxation):

$$\mathbf{D}_\mathbf{A}\Delta\mathbf{v}(i) = (\mathbf{L}_\mathbf{A} + \mathbf{U}_\mathbf{A})\Delta\mathbf{v}(i-1) + \mathbf{J}^T\lambda(i), \qquad (9)$$

where $\mathbf{D}_\mathbf{A}$ is block diagonal, $\mathbf{L}_\mathbf{A}$ strictly lower triangular, $\mathbf{U}_\mathbf{A}$ strictly upper triangular, and $\mathbf{A} = \mathbf{D}_\mathbf{A} - \mathbf{L}_\mathbf{A} - \mathbf{U}_\mathbf{A}$. In our cloth and volumetric FEM examples we decompose $\mathbf{A}$ into 3×3 blocks.

The formulation above allows us to anticipate the effect of contact constraints in an iterative manner. Specifically, ICA allows us to substitute $\Delta\mathbf{v}(i)$ into the constraint equation in MLCP (6), and obtain the following LCP, which can then be used for computing $\lambda(i)$:

$$\begin{aligned} &\mathbf{0} \le \lambda(i) \quad \perp \quad \mathbf{B}\lambda(i) \ge \mathbf{c}(i), \qquad (10)\\ &\text{with } \mathbf{B} = \mathbf{J}\mathbf{D}_\mathbf{A}^{-1}\mathbf{J}^T,\\ &\text{and } \mathbf{c}(i) = -\frac{1}{\Delta t}\mathbf{g}_0 - \mathbf{J}\mathbf{v}^* - \mathbf{J}\mathbf{D}_\mathbf{A}^{-1}(\mathbf{L}_\mathbf{A} + \mathbf{U}_\mathbf{A})\Delta\mathbf{v}(i-1). \end{aligned}$$

ICA presents two major advantages over the full constraint anticipation discussed in Section 5.1: (i) The matrix of the LCP, $\mathbf{B}$, requires the simple inversion of the block diagonal matrix $\mathbf{D}_\mathbf{A}$, as opposed to solving a large linear system per constraint; and (ii) $\mathbf{B}$ is very sparse, not dense any more, since off-diagonal terms are non-zero only if two constraints affect the same simulation node. Moreover, note that ICA reformulates at each iteration only the right-hand side $\mathbf{c}(i)$, and the matrix $\mathbf{B}$ needs to be computed only once per constrained dynamics update.

ICA yields two nested problems: as outer problem, the iterative relaxation of $\Delta\mathbf{v}$ and, as inner problem, the sparse LCP for computing λ. We next describe the two relaxation solvers used in these nested problems.

5.3. Nested Relaxation Loops

We solve the inner problem in ICA, i.e., the LCP in Eq. (10), using projected Gauss-Seidel (G-S) relaxation [CPS92]. Then, the j^{th} iteration is formulated as:

$$\mathbf{0} \le \lambda(i,j) \perp (\mathbf{D}_\mathbf{B} - \mathbf{L}_\mathbf{B})\lambda(i,j) \ge \mathbf{c}(i) + \mathbf{U}_\mathbf{B}\lambda(i,j-1), \quad (11)$$

where $\mathbf{D}_\mathbf{B}$ is diagonal, $\mathbf{L}_\mathbf{B}$ strictly lower triangular, $\mathbf{U}_\mathbf{B}$ strictly upper triangular, and $\mathbf{B} = \mathbf{D}_\mathbf{B} - \mathbf{L}_\mathbf{B} - \mathbf{U}_\mathbf{B}$. We found that the inner problem typically converges in few iterations (less than 10 in 90% of the cases) with a warm start from the previous iteration of the outer problem.

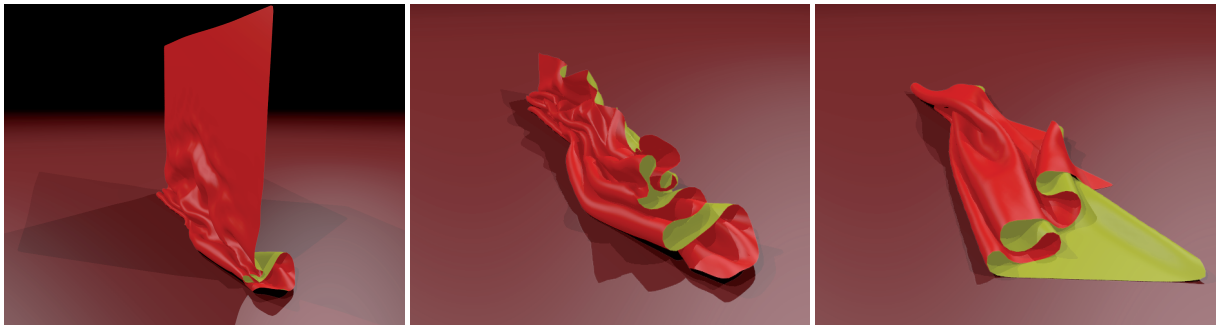

Figure 5: ***Falling Cloth.*** *A piece of cloth is dropped and it produces many stacking folds on the ground. Our contact handling ensures the lack of inter-penetrations and a response that does not add energy to the cloth.*

Once the Lagrange multipliers λ are computed, the outer problem in ICA must be resolved (i.e., the refinement of $\Delta\mathbf{v}$). We obtained good convergence in practice by doing simply one block-G-S iteration as described by:

$$(\mathbf{D_A} - \mathbf{L_A})\Delta\mathbf{v}(i) = \mathbf{U_A}\Delta\mathbf{v}(i-1) + \mathbf{J}^T\lambda(i). \qquad (12)$$

Note that we propose block-G-S for refining collision response due to better convergence, while we propose block-Jacobi in Eq. (9) for iteratively anticipating the constraints. The effect of this mismatch is that the Lagrange multipliers $\lambda(i)$ at the i^{th} iteration do not necessarily guarantee that constraints are resolved for $\Delta\mathbf{v}(i)$. However, this is not a problem, as constraints *are resolved* once the nested relaxation converges. One could think of using block-G-S for iteratively anticipating the constraints, but that would result in a dense matrix $\mathbf{B}$ and worse performance.

5.4. Friction

Our approach for solving the frictional constrained dynamics problem is to interlace the solution of normal and tangential constraints, similar to [DDKA06]. For the rest of this section, we drop the subindex i from λ and γ for better readability. In the inner loop of ICA we first assume the frictional response γ to be known, which results in an LCP that defines the normal response λ:

$$\mathbf{0} \le \lambda \quad \perp \quad \mathbf{B}_\lambda\lambda \ge \mathbf{c}_\lambda(i) - \mathbf{B}_{\lambda\gamma}\gamma, \qquad (13)$$
$$\text{with } \mathbf{B}_\lambda = \mathbf{J}\mathbf{D}_\mathbf{A}^{-1}\mathbf{J}^T, \quad \mathbf{B}_{\lambda\gamma} = \mathbf{J}\mathbf{D}_\mathbf{A}^{-1}\mathbf{H}^T,$$
$$\text{and } \mathbf{c}_\lambda(i) = -\frac{1}{\Delta t}\mathbf{g}_0 - \mathbf{J}\mathbf{v}^* - \mathbf{J}\mathbf{D}_\mathbf{A}^{-1}(\mathbf{L_A} + \mathbf{U_A})\Delta\mathbf{v}(i-1).$$

We then assume the normal response to be known, which, given the linearized Coulomb friction model from Section 3.4, results in an LCP that defines the frictional response:

$$\|\gamma\| \le \mu\lambda \quad \perp \quad \mathbf{B}_\gamma\gamma \ge \mathbf{c}_\gamma(i) - \mathbf{B}_{\lambda\gamma}^T\lambda, \qquad (14)$$
$$\text{with } \mathbf{B}_\gamma = \mathbf{H}\mathbf{D}_\mathbf{A}^{-1}\mathbf{H}^T,$$
$$\text{and } \mathbf{c}_\gamma(i) = -\mathbf{H}\mathbf{v}^* - \mathbf{H}\mathbf{D}_\mathbf{A}^{-1}(\mathbf{L_A} + \mathbf{U_A})\Delta\mathbf{v}(i-1).$$

5.5. Complete ICA Algorithm

As discussed in Section 5.3, we resolve the inner loop in ICA through projected-G-S relaxation. In the frictional case, we apply this relaxation to both normal and friction LCPs formulated above. In our implementation, we interlace normal and friction relaxation at the level of each contact. In other words, we refine the normal response of one contact, then the friction response of the same contact, and we move to the next contact. The complete ICA algorithm can be summarized as follows:

1. Initialize $\Delta\mathbf{v}(0), \lambda(0), \gamma(0)$ with values from the previous constrained dynamics solve. If a constraint was just added, set $\lambda(0) = \gamma(0) = 0$.
2. Formulate the sparse matrices $\mathbf{B}_\lambda, \mathbf{B}_\gamma, \mathbf{B}_{\lambda\gamma}$.
3. (Outer loop) Do ICA until convergence:
   - 3.1. Formulate $\mathbf{c}_\lambda(i), \mathbf{c}_\gamma(i)$.
   - 3.2. (Inner loop) Iterate LCPs with proj. G-S:
     - 3.2.1. For each contact j, iterate 3 times:
       - 3.2.1.1. Compute λ_j in $\mathbf{B}_\lambda\lambda + \mathbf{B}_{\lambda\gamma}\gamma - \mathbf{c}_\lambda(i) = 0$.
       - 3.2.1.2. Project $\lambda_j \ge 0$.
       - 3.2.1.3. Compute $\gamma_{1,j}$ in $\mathbf{B}_{\lambda\gamma}^T\lambda + \mathbf{B}_\gamma\gamma - \mathbf{c}_\gamma(i) = 0$.
       - 3.2.1.4. Project $\|\gamma_{1,j}\| \le \mu\lambda_j$.
       - 3.2.1.5. Compute $\gamma_{2,j}$ in $\mathbf{B}_{\lambda\gamma}^T\lambda + \mathbf{B}_\gamma\gamma - \mathbf{c}_\gamma(i) = 0$.
       - 3.2.1.6. Project $\|\gamma_{2,j}\| \le \mu\lambda_j$.
   - 3.3. Refine $\Delta\mathbf{v}(i)$ with one iteration of block-G-S: $(\mathbf{D_A} - \mathbf{L_A})\Delta\mathbf{v}(i) = \mathbf{U_A}\Delta\mathbf{v}(i-1) + \mathbf{J}^T\lambda(i) + \mathbf{H}^T\gamma(i)$.

In practice, we terminate the inner loop of ICA when the residuals of the complementarity conditions in both non-penetration and frictional LCPs change less than 5% between two iterations, and the distance along the constraint normal is larger than half the collision detection tolerance at all contacts. We terminate the outer loop when the previous conditions hold and, in addition, the residual of collision response $\mathbf{A}\Delta\mathbf{v}(i) - \mathbf{J}^T\lambda(i) - \mathbf{H}^T\gamma(i)$ changes less than 5% between two iterations. Note that the residuals of the LCPs need to be re-evaluated after applying collision response.

5.6. Discussion

Let us assume a deformable object with n degrees of freedom, m contact constraints, and semiregular meshing. The cost of the traditional solver for MLCPs based on constraint anticipation (as discussed in Section 5.1), can be decomposed as follows: the cost of computing the LCP matrix $\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^T$ is at best $O(n \cdot m + m^2)$, while each iteration of the LCP solver takes $O(m^2)$, due to the dense matrix. On the other hand, the cost per iteration of our next relaxation solver in ICA is only $O(n+m)$. As shown by the performance data reported in Section 6, our ICA algorithm clearly outperforms full constraint anticipation in practice, and the performance gain grows with the size of the problem.

The existence of a solution for the frictionless constrained dynamics problem is guaranteed if the time step starts with a collision-free configuration, since a collision response $\Delta\mathbf{v} = -\mathbf{v}^*$ trivially satisfies all constraints. Note however that a scene with scripted moving objects could suffer from non-physical situations like cloth pinched between an arm and the body, but such situations require global untangling approaches as addressed by Baraff et al. [BWK03]. Beyond existence of a solution, we have observed that the convergence of the solver depends on the convergence of the block-G-S relaxation of the dynamics. Convergence is guaranteed if $\mathbf{A}$ is diagonally dominant [GV96], but less restrictive situations work well in practice. In our implementation, we monitor the residual to check whether G-S converges. If it does not, we halve the time step and restart a loop of CMR.

6. Results

All our experiments were executed on a 3.2GHz processor PC with 2.0GB of memory. The dress in the flamenco example from Fig. 1 and the falling cloth from Fig. 5 were simulated as mass-spring systems. The dress contains 6 layers and 16800 vertices, and the falling cloth contains 2500 vertices. The scene in Fig. 7 consists of 164 letters, with 23000 tetrahedra in total, simulated with co-rotational FEM. The scene in Fig. 4 combines cloth pieces, FEM deformable bodies, and rigid bodies.

The falling cloth in Fig. 5 shows the robust geometric handling of non-penetration and stacking folds of cloth. The accompanying video clearly demonstrates the smooth behavior under both small and large friction coeficients. For rendering purposes, we added one step of constrained Loop subdivision as a post-process. The simulation took approximately 8 minutes with a time step of 4 ms for a total of 4 seconds of simulation, with 69% of the time devoted to constrained dynamics solve and 31% to collision detection.

The flamenco dance simulation from Fig. 1 imposes extremely challenging conditions for contact handling, as the overlaid dress pieces are pulled and/or compressed at high speed by the motion of the body. There are some particularly complex situations, e.g., when the dancer kicks the dress, or performs a 360 turn in less than one second. In all these situations, our algorithm produces physically robust collision response, which results in inelastic impacts and smooth sliding. Note that, in this example, the shirt of the dancer was not simulated. It was added as a post-process for cosmetic reasons, and intersections with the rest of the model were not checked. The simulation of the flamenco dance took approximately 15 hours to compute with a time step of 2 ms for a sequence of about 10 seconds. The bottleneck in this example turned out to be continuous collision detection, with an average of 1.5 seconds per query. The computation of one constrained dynamics solve took 1.2 second on average, although at sparse times (about 0.1%) the relaxation solver required up to 10 times longer. The number of contacts grew up to 6000 in this example.

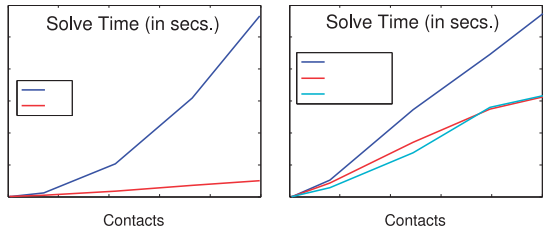

Figure 6: ***Performance Evaluation.*** *Time needed to compute a full time step (30 ms) in the* letters *demo (see Fig. 7), plotted vs. the number of contacts. Timings are computed by averaging results over windows of* 600 *ms in the simulation. Left: comparison between the full constraint anticipation approach (CA) vs. our ICA. Right: comparison for different FEM mesh resolutions (all with ICA).*

The stacking letters in Fig. 7 were modeled as triangle meshes embedded in tetrahedral meshes. The scene starts with layers of letters being dropped on each other, the pile comes to rest, and then the walls push in and compress the pile. The simulation exhibits very vivid motion effects, thanks to the accurate geometric handling of sharp features, and the physically robust handling of friction forces. The complete simulation needed just over 90 minutes for the computation of about 40 seconds of animation, with a time step of 30 ms. Fig. 6 evaluates the performance of our solver as a function of the number of contacts and the resolution of the tetrahedral meshes, as well as in comparison with full constraint anticipation. In the right plot we can see that our ICA algorithm has in practice quasi-linear complexity w.r.t. the number of contacts. Performance is practically independent of mesh resolution when the letters have less than 140 tetrahedra on average, which, as expected, implies that the inner loop of ICA (i.e., the refinement of Lagrange multipliers) dominates the cost at low mesh resolutions. In the left plot we can see, instead, that the full constraint anticipation suffers,

Figure 7: ***Stacking Deformable Letters.*** *This animation shows accurate geometric handling of sharp features, as well as physically robust impacts, sliding, and stick-slip effects, at large time steps (*30 *ms).*

as expected, quadratic complexity (at best). This difference clearly shows the benefits of our contact handling method for large contact problems. Moreover, the data for the left plot was obtained in a frictionless simulation, as the full constraint anticipation method suffered severe convergence problems with many less contacts in the frictional case. We have also evaluated the performance of CMR. The plot to the right shows a histogram of the number of iterations needed by CMR for finding a valid state that satisfies the constraint manifold. CMR found a valid solution in 3 iterations or less in more than 90% of the cases. The statistics were very similar for the rest of the examples.

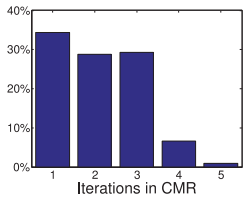

Fig. 4 shows a scene where we have combined various types of objects. Specifically, rigid bodies, FEM corotational deformable bodies, and mass-spring cloth pieces. The objects are dropped on top of each other, and our contact handling algorithm robustly handles the stack in a unified manner. The algorithm is perhaps suboptimal if only rigid bodies are used (in comparison to approaches like [GBF03, KEP05, Erl07]), but its main feature is that no special care is needed for dealing with contact between cloth and solid bodies or FEM deformable models and rigid bodies. The example exhibits complex situations where the cloth pieces are trapped between bodies that are falling, but both non-penetration and friction are dealt with robustly.

7. Summary and Future Work

We have presented an efficient algorithm for geometrically and physically robust contact handling among deformable objects. Our algorithm features dynamics-awareness by formulating an implicit constrained dynamics problem. Two major contributions, a time stepping algorithm based on constraint manifold refinement, and an MLCP solver based on iterative constraint anticipation, allow us to efficiently and robustly solve constrained deformation dynamics.

In most of the cases, our constrained dynamics solver performed almost as fast as the unconstrained solve, and showing close to linear cost. But there are sparse situations when convergence may be slow. Fortunately, there is room for improvement in our solver's relaxation loops, by incorporating multigrid algorithms. However, the extension of multigrid solvers to constrained problems that change dynamically is not trivial. As shown in the examples, our algorithm seamlessly handles the combination of deformable and rigid bodies, but its performance is not optimal in that case, because the LCP matrices partially lose their sparsity. The recent approach of Shinar et al. [SSF08] could perhaps help in this situation. Integration of shock propagation [GBF03, Erl07] would also be interesting for handling more efficiently the stacking of rigid bodies.

Our algorithm does not handle pinched or tangled situations, although one could think of ways to incorporate existing untangling algorithms as part of the time integration. However, our algorithm dealt robustly with large simulation time steps. This is in part thanks to the position-level constraint formulation in the CMR algorithm, and we believe that other collision response approaches could benefit from it. In our examples we always relied on the constrained formulation, but there are situations where repulsion forces may be more efficient for contact pruning, and it would be interesting to test them in combination with our approach. Very recently, Kaufman et al. [KSJP08] have presented a contact handling algorithm that, similar to ours, interlaces the solve of normal and frictional response. However, they execute a full iteration on normal response before moving on to frictional response, which provides them with certain convergence guarantees. We plan to explore the application of such ideas to our ICA algorithm.

Last, we plan to complement our efficient and robust collision response algorithm with more elaborate deformation models, both for cloth and for deformable solids.

Acknowledgements

This project was partially funded by Spanish Science and Innovation Dept. (proj. TIN2007-67188), Comunidad de Madrid (proj. S-0505/DPI/0235; GATARVISA), and the NCCR Co-Me of the Swiss NSF. We would also like to thank the members of the CG lab at ETH and the GMRV group at URJC.

References

[Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. *Proc. of ACM SIGGRAPH* (1994).

[Bar96] BARAFF D.: Linear-time dynamics using Lagrange multipliers. *Proc. of ACM SIGGRAPH* (1996).

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *Proc. of ACM SIGGRAPH* (2002).

[BJ07] BARBIČ J., JAMES D. L.: Time-critical distributed contact for 6-dof haptic rendering of adaptively sampled reduced deformable models. *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2007).

[BW92] BARAFF D., WITKIN A. P.: Dynamic simulation of nonpenetrating flexible bodies. *Proc. of ACM SIGGRAPH* (1992).

[BW98] BARAFF D., WITKIN A. P.: Large steps in cloth simulation. *Proc. of ACM SIGGRAPH* (1998).

[BWK03] BARAFF D., WITKIN A. P., KASS M.: Untangling cloth. *Proc. of ACM SIGGRAPH* (2003).

[CK05] CHOI K.-J., KO H.-S.: Advanced topics on clothing simulation and animation, 2005. ACM SIGGRAPH Conference Course Notes.

[CPS92] COTTLE R., PANG J., STONE R.: *The Linear Complementarity Problem*. Academic Press, 1992.

[DDKA06] DURIEZ C., DUBOIS F., KHEDDAR A., ANDRIOT C.: Realistic haptic rendering of interacting deformable objects in virtual environments. *Proc. of IEEE TVCG 12*, 1 (2006).

[Erl07] ERLEBEN K.: Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. on Graphics 26*, 2 (2007).

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *Proc. of ACM SIGGRAPH* (2003).

[GV96] GOLUB G. H., VAN LOAN C. F.: *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996.

[HB00] HOUSE D. H., BREEN D. E.: *Cloth Modeling and Animation*. AK Peters, 2000.

[HVTG08] HARMON D., VOUGA E., TAMSTORF R., GRINSPUN E.: Robust treatment of simultaneous collisions. *Proc. of ACM SIGGRAPH* (2008).

[KEP05] KAUFMAN D. M., EDMUNDS T., PAI D. K.: Fast frictional dynamics for rigid bodies. *Proc. of ACM SIGGRAPH* (2005).

[KSJP08] KAUFMAN D. M., SUEDA S., JAMES D. L., PAI D. K.: Staggered projections for frictional contact in multibody systems. *Proc. of ACM SIGGRAPH Asia* (2008).

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. *Proc. of Graphics Interface* (2004).

[MHHR06] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Proc. of VRIPHYS* (2006).

[MW88] MOORE M., WILHELMS J.: Collision detection and response for computer animation. *Proc. of ACM SIGGRAPH* (1988).

[NMK*05] NEALEN A., MÜLLER M., KEISER R., BOXERMANN E., CARLSON M.: Physically based deformable models in computer graphics. *Eurographics STAR* (2005).

[PPG04] PAULY M., PAI D. K., GUIBAS L. J.: Quasi-rigid objects in contact. *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004).

[Pro95] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Proc. of Graphics Interface* (1995).

[Pro97] PROVOT X.: Collision and self-collision handling in cloth model dedicated to design garment. *Proc. of 8th Eurographics Workshop on Computer Animation and Simulation* (1997), 177–189.

[RF06] RAGHUPATHI L., FAURE F.: QP-Collide: A new approach to collision treatment. *French Working Group on Animation and Simulation (GTAS)* (2006).

[RKC02] REDON S., KHEDDAR A., COQUILLART S.: Gauss' least constraints principle and rigid body simulations. *Proc. of ICRA* (2002).

[SBT07] SPILLMANN J., BECKER M., TESCHNER M.: Noniterative computation of contact forces for deformable objects. *Journal of WSCG* (2007).

[SDCG08] SAUPIN G., DURIEZ C., COTIN S., GRISONI L.: Efficient contact modeling using compliance warping. *Proc. of Computer Graphics International* (2008).

[SMT08] SIFAKIS E., MARINO S., TERAN J.: Globally coupled impulse-based collision handling for cloth simulation. *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008).

[SSF08] SHINAR T., SCHROEDER C., FEDKIW R.: Two-way coupling of rigid and deformable bodies. *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008).

[ST96] STEWART D. E., TRINKLE J. C.: An implicit time-stepping scheme for rigid body dynamics with inelastic collsions and Coulomb friction. *International Journal of Numerical Methods in Engineering 39* (1996).

[THM*03] TESCHNER M., HEIDELBERGER B., MÜELLER M., POMERANETS D., GROSS M.: Optimized spatial hashing for collision detection of deformable objects. *Proc. of Vision, Modeling and Visualization* (2003).

[TKH*05] TESCHNER M., KIMMERLE S., HEIDELBERGER B., ZACHMANN G., RAGHUPATHI L., FURHMANN A., CANI M.-P., FAURE F., MAGNENAT-THALMANN N., STRASSER W., VOLINO P.: Collision detection for deformable objects. *Computer Graphics Forum 24*, 1 (2005).

[VMT97] VOLINO P., MAGNENAT-THALMANN N.: Developing simulation techniques for an interactive clothing system. *Proc. of Virtual Systems and Multimedia* (1997).

[VMT00] VOLINO P., MAGNENAT-THALMANN N.: Accurate collision response on polygonal meshes. *Proc. of Computer Animation* (2000).

[Wri02] WRIGGERS P.: *Computational Contact Mechanics*. Wiley, 2002.