The Visual Computer manuscript No.
(will be inserted by the editor)

Guillaume Saupin · Christian Duriez · Stephane Cotin · Laurent Grisoni

Efficient Contact Modeling using Compliance Warping

Abstract Contact handling is the key of deformable objects simulation, since without it, objects can not interact with their environment nor with the user. In this paper, we propose a novel and very efficient approach for precise computation of contact response between various types of objects commonly used in computer animation. Being constraint based, this method ensures physical correctness, and respects Singorini's law. It can be used with any deformation model, and is based on the use of the initial compliance matrix and contact warping. Thus, the contact response can be computed efficiently, and the object deformation can still be done in a physically plausible way provided the underlying model is physical.

Keywords Contact Handling · Real-Time Simulation · Physically Based Simulation

1 Introduction

Modeling and simulating the behavior of rigid and deformable objects remains an important research area in computer graphics. Various approaches have been proposed to improve the realism of the simulations ([18] [16] for instance) as well as the computational performance ([9] [8] for instance). This has led to the simulation of more and more complex scenes, with dozens, sometimes hundreds of objects, some of them rigid, deformable or even fluid. However, the majority of the proposed methods do not explicitly address the problem of modeling the contacts that occur when these different objects collide. This is particularly true in the case of deformable objects. While a large effort has been put recently toward collision detection between deformable structures, little has been done regarding the precise modeling of contacts between such objects. Additionally, when interactive simulations are required, the complexity of the deformation model, as well as the large number of degrees of freedom, make it a very challenging problem.

Guillaume Saupin
CEA DRT/DTSI/LIST/LSI
E-mail: guillaume.saupin@cea.fr

Christian Duriez, Stephane Cotin
INRIA Futurs
E-mail: christian.duriez@inria.fr
E-mail: stephane.cotin@inria.fr

In this paper we propose a novel and very efficient approach for precise computation of contact response between various types of objects commonly used in computer animation. Our method offers several advantages over previous work. In particular we propose a formalism where an approximated contact model can be derived from the behavior model of the object while verifying Signorini's law of contact and Coulomb's law of friction. This is illustrated on several examples, including deformable models where an approximated compliance matrix is used to estimate the objects' motion required to solve the contact.

1.1 Previous work

Contact modeling in computer animation is a challenging problem for several reasons. First, the way contacts are handled plays a very important role in the overall behavior of the interacting objects. The choice of the contact model (penalty force, impulse, constraint, ...), how it verifies Signorini's law, and the inclusion or not of friction, highly influence the post-impact motion of the interacting objects. Additionally, when a contact between objects occurs, it induces quick changes in their dynamic behavior. Such changes are difficult to reproduce manually, in particular when deformable objects are involved, and often lead to instabilities or visual inconsistencies when using physics-based animations. Finally, when multiple objects are in contact, the solution space for the new, non-interpenetrating, configuration is reduced. The high computational complexity involved in resolving such contacts can become an important bottleneck of the simulation, sometimes more time consuming than the computation of the dynamics of the object.

Contact modeling has been extensively studied in Mechanics, and research on modeling the non-smooth dynamic behavior of objects in contact remains an active topic [1]. In the field of Computer Graphics, several solutions have been proposed to address this problem. The most popular approach is the penalty method which consists in defining a contact force $F = k\delta$ at each contact point where δ is a measure of the interpenetration between a pair of colliding objects, and k is a stiffness parameter. This stiffness parameter must be large enough to avoid any visible interpenetration, however, its value cannot be determined exactly. Instead, the choice of the value of k depends of the nature of the objects, the type of interactions, and other elements of the simulation, which leads to various heuristics to determine the ideal stiffness parameter. Yet, no matter how k is chosen, interpenetrations between the colliding objects can only be reduced, not "resolved". This a direct consequence of the method itself, which generates forces only when the interpenetration distance δ is negative (assuming δ is chosen to be negative when an interpenetration exists, and positive when the objects are no longer in contact). Signorini's law of contact states that there is a complementarity relation between δ and the contact force f at the point of contact: $0 \leq \delta \perp f \geq 0$. This condition is not met when using penalty forces. In addition, if an explicit time integration scheme is used, and k is large, very small time steps are required to guarantee the stability. As explicit integration schemes are conditionally stable, using a penalty method therefore requires that two criteria are met: k must be large enough to limit interpenetrations, and the time step must be small. Overall this makes this approach, initially simple, rather inefficient for handling contacts.

A possible improvement over the penalty method can be achieved through the use of an implicit integration scheme. Implicit methods have the advantage of providing more stable simulations even with rather large time steps [5]. When combining an implicit integration scheme with a penalty method, it becomes possible to use large stiffness values without compromising the stability of the simulation. Yet, solving the resulting stiff and non-smooth system can be computationally prohibitive when the objective is to reduce as much as possible the interpenetration distance.

Another way to easily handle contact relies on the use of impulse-based methods. Originally employed to handle contact between rigid object [12, 11], these methods have been extended to deformable bodies [6]. Impulse-based relies on velocity correction, and do not involve constraints nor forces. Whenever two objects are colliding, each one is subject to an opposite impulse which avoid the interpenetration. Hence, a body resting on a table is continuously experiencing collisions with the table, and experience associated impulses. Using these methods, each type of contact, i.e. colliding, rolling, sliding, and resting, can be simulated in the same way. However, these methods handles poorly stable and simultaneous contacts, as well as static friction.

Overall, the methods described above share an important limitation when dealing with multiple contacts: they consider each contact independently while in reality they are coupled. This limitation can be solved using constraint-based techniques, which can solve "exactly" the contact problem (i.e. no interpenetration at the end of the time step). Such approaches often rely on the use of Lagrange multipliers, which are appropriate for handling bilateral constraints [8]. However, contacts between objects intrinsically define unilateral constraints, which means that Signorini's law of contact is not verified when using techniques based on Lagrange multipliers. As a consequence, when deformable objects collide, they will appear stuck at the end of the time step. Improvements over Lagrange multipliers techniques are possible by using a Linear Complementary Problem (LCP) formulation deriving from Signorini's law. The solution of the LCP gives an accurate description of the contact forces needed to zero out the interpenetration, and prevents objects to stick together. Pauly *et al.* [17] for instance have proposed such an approach to solve contacts between quasi-rigid objects. They use a Lemke solver to compute a contact-free configuration from the LCP formulation. By expanding the LCP, or by using a non-linear solver, the formulation can be extended to model both static and dynamic friction. For rigid objects, see for instance, [3] or [2] and for deformable objects, see [17] or [7]. Computationally efficient methods for solving linear complementary problems have been proposed [15], thus making such approaches appealing even for interactive simulations. Yet, when dealing with deformable models, real-time computation of the solution is almost impossible since the LCP algorithm relies on the computation of the inverse of the stiffness matrix for each object in contact. While this inverse can be pre-computed in the case of linear elastic models [7], this is not possible for non-linear deformable models.

1.2 Contributions

In this paper we propose a method for precisely solving multiple coupled contacts between various types of objects. Friction between colliding objects is taken into account, and the method guarantees that all interpenetration are solved at the end of each time step. Section 2 describes a general technique for handling both rigid and deformable dynamic objects in contact: first, objects are integrated separately by considering their internal behavior and the known external forces, then, after a computation of contact forces, a corrective motion is applied. The proposed approach for processing multiple contacts is compatible with all types of dynamic objects and only requires an estimation of their mechanical compliance. We introduce in Section 3 the use of an approximation of the object's compliance during the corrective motion and show that it improves the computation time significantly with minimal impact on the accuracy. Finally Section 4 summarizes the results obtained on various simulations and Section 5 addresses possible directions for future work.

2 Multicontact on multimodels

The present work is motivated by providing an efficient way of solving contact and friction laws onto fixed time step simulations. We aim at providing a method that is compatible with a large number of object types, including rigid and deformable models. In this section we introduce a common (time-stepping) scheme which allows interaction between deformable and rigid objects. We also present a contact processing approach where the motion of each object is decomposed in a free motion and a corrective motion. The corrective motion is computed using an LCP type algorithm (as in [2,17] for instance). However, the novelty of our approach consist in using an approximation of the behavior model during the corrective motion instead of the exact model itself.

For the contact, we use the physics-based model of Signorini's law. It states that there is a complementarity relation, at each contact point, between the interpenetration gap $\boldsymbol{\delta}$ and the contact force $\mathbf{f}$ along the normal direction, that is:

$$0 \leq \boldsymbol{\delta} \perp \mathbf{f} \geq 0 \tag{1}$$

$\boldsymbol{\delta} \geq 0$ ensures the non-interpenetration, while $\mathbf{f} \geq 0$ guaranties that there is no stick forces, and $\boldsymbol{\delta} \perp \mathbf{f}$ states that there is a contact force $\mathbf{f} > 0$ if and only if $\boldsymbol{\delta} = 0$. When dealing with friction, two tangential directions $\mathbf{t}$ and $\mathbf{s}$ are associated. It creates a frame $\mathfrak{F}_\alpha = [\mathbf{n}_\alpha, \mathbf{t}_\alpha, \mathbf{s}_\alpha]$ for each contact α. With the Coulomb's friction law, the contact force lies within a spacial conical region whose height and direction is given by the normal force, giving two complementarity conditions for stick and slip motions.

$$\begin{array}{ll} [{}^t\delta \quad {}^s\delta] = \mathbf{0} \Rightarrow \|[{}^t\mathfrak{f} \quad {}^s\mathfrak{f}]\| < \mu\ \|{}^n\mathfrak{f}\| & \text{(stick condition)} \\ [{}^t\delta \quad {}^s\delta] \neq \mathbf{0} \Rightarrow\ [{}^t\mathfrak{f} \quad {}^s\mathfrak{f}] = -\mu\ \|{}^n\mathfrak{f}\| \frac{[{}^t\delta \quad {}^s\delta]}{\|\ [{}^t\delta \quad {}^s\delta]\|} & \text{(slip condition)} \end{array} \tag{2}$$

where μ is the friction coefficient.

These complementarity relations could create *singular* events when it changes from one state to an other: the acceleration could be undefined. For instance, when objects collide at instant $t^\star$, their relative velocities $\mathbf{v}(t^\star)$ is discontinuous. To handle this problem, two solutions are proposed: *Event driven* schemes stop the time integration at each new non-smooth event [4]. Then, it solves the contact and friction laws and restarts the time integration with new initial conditions. It provides precise results but it is usable only for a very small number of instantaneous contacts. On the contrary, *time stepping* scheme is based on a mathematical formulation which includes all contact appearing during a fixed time step. Anitescu *and al.* [2] have shown that it can be written as an LCP for rigid objects. Our objective is to extend *time stepping* scheme to deformable objects while keeping it compatible with rigid models.

2.1 Contact integration

First, we rewrite a time-stepping scheme for a generic dynamic model. Equations used to model the dynamic behavior of bodies have led to a synthetic formulation:

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\ (\mathbf{q}, \mathbf{v}, t) + \mathbf{r} \tag{3}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the vector of generalized degrees of freedom (for instance, displacement of a mesh or displacement and rotation of a rigid body), $\mathbb{M}(\mathbf{q}) : \mathbb{R}^n \mapsto \mathcal{M}^{n\times n}$ is the inertia matrix, $\mathbf{v} \in \mathbb{R}^n$ is the vector of velocity. $\mathbb{F}$ represents internal forces and $\mathbb{P}$ gathers external forces. $\mathbf{r} \in \mathbb{R}^n$ is the vector of contact forces contribution.

For a rigid object, equation (3) would be written:

$$\begin{bmatrix} m\ 0 \\ 0\ \ \mathbf{I}(\boldsymbol{\Omega}) \end{bmatrix} \begin{pmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = \begin{pmatrix} \mathbf{p} + \mathbf{r} \\ \mathbf{m_p} - \boldsymbol{\omega} \wedge (\mathbf{I}\boldsymbol{\omega}) + \mathbf{m_r} \end{pmatrix} \tag{4}$$

where $\mathbf{I}$ is the inertia, $\boldsymbol{\omega}$ and $\boldsymbol{\Omega}$ are the angular velocity and position of the rigid virtual object. $\mathbf{m_p}$ and $\mathbf{m_r}$ are the external and contact torques. For a deformable model, $\mathbb{M}(\mathbf{q})$ and $\mathbb{F}\ (\mathbf{q}, \mathbf{v}, t)$ are based on the constitutive law of the deformable material. We will describe in more details their value in section 2.4.

Using *time-stepping* method, the time step is fixed and there is no limitation on the number of discontinuity that could happen during a time step ([2]). In this case, integrator's order is low such as 1 or 2. This could lead to excessive dissipation if the time step are too large. However it provides stable simulations.

Let's consider the time interval $[t_i, t_f]$ which length is $h = t_f - t_i$. We have:

$$\mathbf{M}(\mathbf{v}_f - \mathbf{v}_i) = \int_{t_i}^{t_f} (\mathbb{P}(t) - \mathbb{F}(\mathbf{q}, \mathbf{v}, t))\, dt\ +\ h\mathbf{r}_f \tag{5}$$

$$\mathbf{q}_f = \mathbf{q}_i + \int_{t_i}^{t_f} \mathbf{v} dt \tag{6}$$

To evaluate integrals $\int_{t_i}^{t_f} (\mathbb{P}(t) - \mathbb{F}(\mathbf{q}, \mathbf{v}, t))\, dt$ and $\int_{t_i}^{t_f} \mathbf{v} dt$ we chose an implicit Euler integration scheme:

$$\mathbf{M}(\mathbf{v}_f - \mathbf{v}_i) = h\, (\mathbb{P}(t_f) - \mathbb{F}(\mathbf{q}_f, \mathbf{v}_f, t_f)) +\ h\mathbf{r}_f \tag{7}$$

$$\mathbf{q}_f = \mathbf{q}_i + h\mathbf{v}_f \tag{8}$$

$\mathbb{F}$ is a non-linear function, we apply a Taylor series expansion to $\mathbb{F}$ and make the first order approximation:

$$\mathbb{F}\ (\mathbf{q}_i + d\mathbf{q}, \mathbf{v}_i + d\mathbf{v}) = \mathbf{f}_i + \frac{\delta\mathbb{F}}{\delta\mathbf{q}} d\mathbf{q} + \frac{\delta\mathbb{F}}{\delta\mathbf{v}} d\mathbf{v} \tag{9}$$

Using $d\mathbf{q} = \mathbf{q}_f - \mathbf{q}_i = h\mathbf{v}_f$ and $d\mathbf{v} = \mathbf{v}_f - \mathbf{v}_i$, we obtain:

$$\left(\mathbf{M} + h\frac{\delta\mathbb{F}}{\delta\mathbf{v}} + h^2\frac{\delta\mathbb{F}}{\delta\mathbf{q}}\right) d\mathbf{v} = -h^2\frac{\delta\mathbb{F}}{\delta\mathbf{q}}\mathbf{v}_i - h\left(\mathbf{f}_i + \mathbf{p}_f\right) + h\mathbf{r} \tag{10}$$

where $\mathbf{p}_f$ is the value of function $\mathbb{P}$ at time t_f.

The only forces that are unknown are the one related to the contact $\mathbf{r}$. It will lead to a computation in two steps: the first one with the known value of the forces, and the second one for solving the contact forces and applying the corresponding motion to the generic dynamic model.

2.2 Free motion and corrective motion

The computation of each time step begins with the *free motion*: position and velocity of the object are computed according to the physical laws describing its dynamics, without taking in account the collisions with other objects. Using dynamic equations, the solution $\mathbf{v}_{free}$ is found by solving the non-linear equations:

$$\mathbf{M}(\mathbf{v}_{free} - \mathbf{v}_i) = h\left(\mathbb{P} - \mathbb{F}(\mathbf{q}_{free}, \mathbf{v}_{free})\right) \tag{11}$$

$$\mathbf{q}_{free} = \mathbf{q}_i + h\mathbf{v}_{free} \tag{12}$$

Once the free position is known, all the objects in the simulation are tested for collision. This results in a list of contact points associated with contact normals. It allows to define a contact space in which contact response will be solved by a LCP. This part is described in more detail in Subsection 2.3.

Then, knowing the contact forces $\mathbf{r}$ for the set of detected contacts, contact reactions will be integrated in a corrective motion $d\mathbf{v}_c$ using:

$$\left(\mathbf{M} + h\frac{\delta\mathbb{F}}{\delta\mathbf{v}} + h^2\frac{\delta\mathbb{F}}{\delta\mathbf{q}}\right) d\mathbf{v}_c = h\mathbf{r} \tag{13}$$

This correction is added to the free motion to obtain the final motion: $\mathbf{v}_f = \mathbf{v}_{free} + d\mathbf{v}_c$ and $\mathbf{q}_f = \mathbf{q}_i + h\mathbf{v}_f$

In order to be compatible with the Signorini's law, which is based on contact distances and forces, equation (13) can be rewritten to link the contact force $\mathbf{r}$ to the displacement $\Delta\mathbf{q}$ from free position and the final position:

$$\Delta\mathbf{q} = \mathbf{q}_f - \mathbf{q}_{free} = \underbrace{\left(\frac{1}{h^2}\mathbf{M} + \frac{1}{h}\frac{\delta\mathbb{F}}{\delta\mathbf{v}} + \frac{\delta\mathbb{F}}{\delta\mathbf{q}}\right)^{-1}}_{\mathbf{C}}\mathbf{r} \tag{14}$$

with $d\mathbf{v}_c = \Delta\mathbf{q}/h$. The matrix $\mathbf{C}$ provides the displacements due to contact forces, and we call it the compliance thereafter. This compliance will be useful to build a LCP from the Signorini's law.

2.3 LCP formulation

From contact detection, we have a set of potential contact spots $\alpha = 1...n_c$ with their associate direction $\mathbf{n}_\alpha$. We do not know yet if they will be active ($\mathbf{f}_\alpha > 0$) or not ($\mathbf{f}_\alpha = 0$). We have also an interpenetration distance measure $\boldsymbol{\delta}_\alpha$ on the same direction, that must respect the Signorini's law (*i.e.* it must be positive or null)

For every contact and every object, we can build the mapping function $\mathbb{A}$ that links this interpenetration to the final positions $\mathbf{q}_f$ of each dynamic colliding object. Let's consider the contact α, between object 1 and 2 at point P, we have:

$$\boldsymbol{\delta}_\alpha = \mathbb{A}_\alpha(\mathbf{q_1}, t_f) - \mathbb{A}_\alpha(\mathbf{q_2}, t_f) \tag{15}$$

with $\mathbb{A}_\alpha(\mathbf{q}, t)$ a mapping function which depends on the contact α and the positions $\mathbf{q_1}$ and $\mathbf{q_2}$ of the colliding objects. To obtain a kinematic relation between the two spaces (contact and motion), we use a linearization of equation (15). Let's consider $\mathbb{H}_\alpha(\mathbf{q}) = \frac{\partial\mathbb{A}}{\partial\mathbf{q}}$, we obtain at time t for each contact:

$$\Delta\boldsymbol{\delta}_\alpha = \mathbb{H}_\alpha(\mathbf{q_1})\Delta\mathbf{q_1} - \mathbb{H}_\alpha(\mathbf{q_2})\Delta\mathbf{q_2} \tag{16}$$

When the contact α is solved, the dual relation can be applied for the contact forces $\mathbf{f}_\alpha$:

$$\mathbf{r}_1 = \mathbb{H}_\alpha^T(\mathbf{q_1})\mathbf{f}_\alpha \quad \mathbf{r}_2 = -\mathbb{H}_\alpha^T(\mathbf{q_2})\mathbf{f}_\alpha \tag{17}$$

The transformation matrices for each contact and object can be stacked together to form a matrix $\mathbf{H}$ that describes the relative displacement $\Delta\boldsymbol{\delta}$ and contact forces $\mathbf{f}$ in the contact surface frames between all the contact points in the system.

$$\Delta\boldsymbol{\delta} = \mathbf{H}\Delta\mathbf{q} \quad \mathbf{r} = \mathbf{H}^T\mathbf{f} \tag{18}$$

Once the operator $\mathbf{H}$ is defined, we begin the computation with the description of object's mechanics in contact space:

$$\boldsymbol{\delta} = \underbrace{\mathbf{HCH}^T}_{\mathbf{W}}\mathbf{f} + \boldsymbol{\delta}^{free} \tag{19}$$

Where $\mathbf{W}$ is the Delassus operator [13] which gives the mechanical coupling between contacts, in the contact space. $\boldsymbol{\delta}^{free}$ represents the interpenetration found during the free motion.

This equation and the Signorini's law (1) create an LCP that can be solved by several algorithms. An excellent overview of LCP and corresponding solvers by *Murty* can be found in [15]. The solution of this LCP provides a physically based collision response. We remark that Matrix $\mathbf{H}$ in equation (19) is parse and can benefit form a sparse format storage to compute $\mathbf{W}$ from $\mathbf{C}$ quickly.

The LCP formulation can be extended to solve the Coulomb's law by using k-sided pyramidal friction cone [2]. However, it leads to large LCPs that need a long time to be solved by a direct solver. However, some iterative solution are proposed, we choose the Gauss-Seidel iterative solver described in [7].

2.4 Contact involving deformable models

The way we wrote the time-stepping scheme is compatible with all dynamic deformable models, as long as the motion of the colliding points can be mapped to the deformable model motion (*i.e.* an equation (15) can be found for the deformable object). Using equation (3) we can introduce the deformation parameters: the mass function $\mathbb{M}(\mathbf{q})$ corresponds to the amount of mass distributed on the nodes which samples the deformation. It is evaluated in a matrix $\mathbf{M}$, which is very often constant and diagonal (mass-lumping method). $\mathbb{F}$ represents internal forces from constitutive laws that gather elasticity, viscosity, incompressibility, and all other deformation parameters from the material. In most models, it is a non-linear function on which we can apply the Taylor series expansion, as in equation (9):

$$\mathbb{F}\,(\mathbf{q}+\partial\mathbf{q}, \mathbf{v}+\partial\mathbf{v}) \approx \mathbb{F}\,(\mathbf{q}, \mathbf{v}) + \mathbb{K}(\mathbf{q})\partial\mathbf{q} + \mathbb{B}(\mathbf{q}, \mathbf{v})\partial\mathbf{v} \quad (20)$$

Matrix function $\mathbb{B}$ represents the damping and matrix function $\mathbb{K}$ the stiffness between sampling nodes. If $\mathbb{F}$ is a non-linear equation, these matrices are not constant.

Let's take $\mathbf{B} = \mathbb{B}(\mathbf{q}_i, \mathbf{v}_i)$ and $\mathbf{K} = \mathbb{K}(\mathbf{q}_i)$ their value at the beginning of the time interval. The compliance matrix, which is necessary for the computation of the LCP is then obtained by computing:

$$\mathbf{C} = \left(\frac{1}{h^2}\mathbf{M} + \frac{1}{h}\mathbf{B} + \mathbf{K}\right)^{-1} \quad (21)$$

Consequently, $\mathbf{C}$ must be recomputed at each time step. This is very time-consuming and can quickly become a bottleneck for the simulation.

However, in the special case of small displacement, function $\mathbb{F}$ is a constant linear application:

$$\mathbb{F}\,(\mathbf{q}, \mathbf{v}, t) = \mathbf{K}\mathbf{q} + \mathbf{B}\mathbf{v} \quad (22)$$

In that particular case, the matrix $\mathbf{C}$ is constant, and can be precomputed. It leads to fast computation of $\mathbf{W}$ and of the corrective motion [7]. However considering deformations with only small displacement is a strong limitation.

2.5 Approximation

The time-stepping presented above extends the approach, known for rigid bodies, to general dynamic system and, in particular, to deformable objects. This type of formulation is particularly suited to provide stable results even with highly-constrained models (with contact constraints in opposite directions) and with non-smooth dynamic events. We note an advantage of the method: the non-linear free motion can be computed separately with an adapted solver for each model. This part, which can be costly for some deformable objects, can then be easily parallelized or ported onto GPU

But the formulation has one drawback. The computation of the LCP matrix $\mathbf{W}$ necessitates the compliance value of the colliding objects, or at least, the part of the compliance matrix which concerns colliding nodes. With large deformation model, this computation could be very slow. This difficulty could prevent from extending in an efficient way, the time-stepping scheme to all deformable objects.

However, we introduce the idea of an approximation to speed-up the computations. Based on our scheme, the decomposition onto two successive motions allows the use of two different behavior models: We can use an approximate value of the compliance $\widetilde{\mathbf{C}}$ during the corrective motion while maintaining the accurate dynamic model during the free motion. Then, if $\widetilde{\mathbf{C}}$ is sufficiently close to the right value of the compliance, we can use it to compute the matrix of the LCP ($\mathbf{W}$) and the corrective motion ($\Delta\mathbf{q}$, $d\mathbf{v}_c$). This way, even if the motion will be slightly altered, we can still ensure that the corrective motion follows Signorini and Coulomb laws.

In the following section, we provide some details on the use of an approximate compliance value during the corrective motion. We also present an efficient approximation for non-linear elastic and viscoelastic models undergoing large displacements.

3 Contact compliance warping

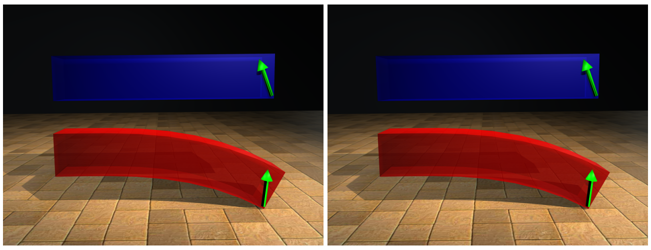
In this section, we present an efficient solution to introduce and solve elastic models in the time-stepping scheme presented above. We consider the case of deformations with large rotations. The solution is based on an approximation of the mechanical compliance that is used to compute the contact forces and subsequent correction motion. **Hence, the cost of the compliance building is independent of the deformable model complexity**. It only depends on the number of contacts points involved.

3.1 Approximated contact model

The approximation on the *compliance* we introduce is only used in the corrective motion step. Thus, it only impact on a small part of the simulation. Moreover, this approximation is based on the object physical model and is similar to the exact *compliance.*

Moreover, we still use the exact Signorini and Coulomb laws to govern contact correction, and compute free motion using the object constitutive law.

This approach is somehow opposite to the penalty methods, that are also approximate contact models: penalty methods rely on a simplification of the Signorini's law but integrate the penalty contact forces with the whole model. The advantage of our strategy lies in the guaranty that no interpenetration will occur at the end of the time step, whatever the stiffness of the object is.



(a) (b)

Fig. 1 (a): The contact normal is rotated back to the initial configuration ; (b): Estimation of node rotation based on adjacent elements

Some other constraints based method also introduced a simplified model. In [8], Lagrange multipliers are used to compute a collision response and avoid interpenetration. An approximation is provided by considering that all contacts are only coupled by a common rigid core. However, the approximation is only valid for small deformations and is not based on accurate contact and friction laws.

To obtain a good approximation, the compliance approximation must reflect the mechanical coupling between contacts, in the directions of contacts. We propose to use the object *compliance* $\mathbf{C}^0$ in its rest position, in conjunction with an estimation of nodes rotation during the deformation. See Figure 1.

Using the well-know Rayleigh damping, with a mass and b sitffness coefficients, the *compliance* is given by:

$$\mathbf{C}^0 = \left(\frac{\mathbf{M}^0}{h^2}(1+a) + \mathbf{K}^0(1+b)\right)^{-1}$$

The resulting approached compliance $\widetilde{\mathbb{C}}(\mathbf{q})$ is given by:

$$\widetilde{\mathbb{C}}(\mathbf{q}) = \mathbb{R}(\mathbf{q})\mathbf{C}^0\mathbb{R}^T(\mathbf{q}) \tag{23}$$

Where $\mathbb{R}(\mathbf{q})$ is a 3×3 block diagonal matrix that gathers the rotation associated with object nodes. This simplification speeds up the computation of the *compliance* needed in the time stepping scheme, because the matrix $\mathbf{C}^0$ could be precomputed.

Such an approximation shares an analogy with deformable corotational models [14, 10]. Corotational models define a deformation tensor independent from rigid rotations. Corotational methods extract a local rotation for each element, so that, in its local frame, each element deformation remains small. The hypothesis of a linear elastic material can then be applied and the element stiffness matrix can be considered constant. Thus, the stiffness matrix of the element in the global frame, $\mathbf{K}_e$, is given by:

$$\mathbf{K}_e = \mathbf{R}_e\mathbf{K}_e^0\mathbf{R}_e^T \tag{24}$$

where $\mathbf{R}_e$ is the element estimated rotation and $\mathbf{K}_e^0$, a precomputed stiffness matrix based on the element rest shape.

The element stiffness of such a model could be defined as:

$$\mathbf{K}_e \approx \begin{bmatrix} \mathbf{R}_1 & & & \\ & \mathbf{R}_2 & 0 & \\ & 0 & \mathbf{R}_3 & \\ & & & \mathbf{R}_4 \end{bmatrix} \mathbf{K}^0 \begin{bmatrix} \mathbf{R}_1 & & & \\ & \mathbf{R}_2 & & \\ & & \mathbf{R}_3 & \\ & & & \mathbf{R}_4 \end{bmatrix}^T \tag{25}$$

Where, the $i \in [|1,4|]$ refers to tetrahedral element four nodes.

Using this approximation, it is possible to extract a global rotation matrix $\mathbb{R}(\mathbf{q})$ for the global stiffness:

$$\mathbb{K}(\mathbf{q}) \approx \mathbb{R}(\mathbf{q})\mathbf{K}^0\mathbb{R}^T(\mathbf{q}) \tag{26}$$

where $\mathbf{K}^0$ is the constant global stiffness matrix estimated at the rest position, and $\mathbb{R}$ gathers the node rotations.

We use the same kind of approximation, but for the *compliance*.

3.2 Corrective motion

When building the Delassus operator, we use the approximate *compliance*. Therefore, to get a motion coherent with the LCP, we compute the corrective motion using the same *compliance*. The exact steps involved in the corrective motion follows:

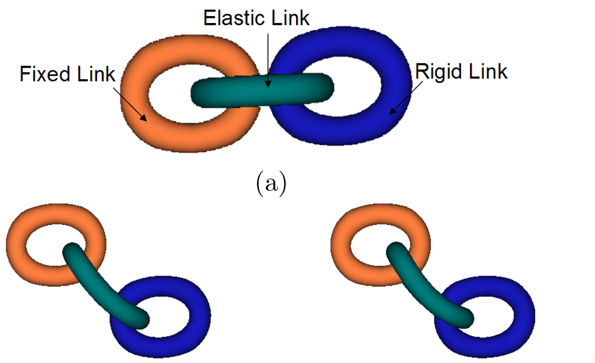
1. We map the contact forces in the original coordinate frame of object k: $\mathbf{r}_k^0 = (\mathbf{H}_k^*)^T\mathbf{f}$
2. We compute the displacement in the original coordinate: $\mathbf{dx}_k^0 = \mathbf{C}_k^0\mathbf{r}_k^0$
3. We rotate back the displacement to the current coordinate frame: $\mathbf{dx}_k = \mathbf{R}_k\mathbf{dx}_k^0$

This method is efficient because a reduced part of the nodes are usually involved in the contact and consequently, $\mathbf{r}_k^0$ is highly sparse. Moreover, the displacement computation provides a perfect correction of the detected interpenetration and follows Signorini and Coulomb laws.

Nevertheless, as a consequence, the corrective motion is not completely based on the object constitutive law. However, small error on deformation are visually less disturbing that error on interpenetration. Moreover, the approximation is partly corrected by the free motion, based on exact constitutive law, of the next time-step.

4 Examples

We have performed all our simulation on an Intel Celeron M 520 at 1.6 GHz with 1Gb of RAM. To give a try to our method, we simulate objects in several scenarios, involving different contact configurations, and measure performances and correctness. The object deformation is based on elasticity theory and uses a corotational model similar to the one presented in [10].



(a)

(b) (c)

Fig. 2 (a): Initial Position. (b) : Deformation using exact *compliance* ; (c) : Deformation using approximated *compliance* ;

	Largest Deformation	Final Shape
3 Links Chain Horizontal	1.5%	1.7%
3 Links Chain Vertical	4.4%	3.1%

Table 1 Relative error on mesh deformation for the largest deformation and for the final shape. The 3 links are initially aligned either in vertical or horizontal position.

	Tetra. Num.	Contact Num.	Exact Compl.	Appro. Compl.
Dino Rain	40K	2000	N/A	3.007s
3 Links Chain	862	35	13794.1ms	8.563ms
3 Links Chain	862	13	6695ms	1.520ms
3 Links Chain	862	36	16908.5ms	9.36ms
3 Links Chain	862	12	8203.6ms	1.366ms
5 Links Chain	2*862	67	11027.8ms	4.72ms

Table 2 Compliance computation at contact points for different models

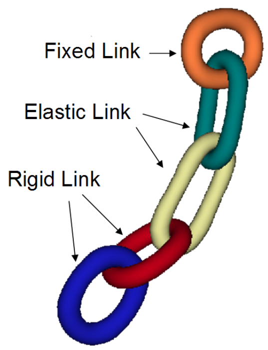


Fig. 3 Different contact configuration are handle in this example : Deformable - Deformable ; Deformable - Rigid ; Rigid - Rigid

4.1 Model validation

We want to assess the quality of our approximation through simulation examples. To do that, we compare the deformation of several models in different situations using our approximation and the exact compliance. However, as the computation of the exact value of the compliance is highly time consuming, we only did the comparison on simple cases.

For instance, we simulate a deformable torus in contact with two rigid rings, one fixed and the other free. See Figure 2-(a). Under the effect of the gravity they fall, impact and oscillate. There is about 35 detected contact points with friction when the 3 links are in contact.

As seen in Figures 2-(b) and 2-(c), there is no visible difference between the two models. The relative, based on the norm of the difference between the object deformation using exact contact correction and our method, is given in Table 1 for two different simulations. For the first one, the chain falls vertically. In the second one, the links are initially positioned horizontally, and the falls create more rotations. For both cases, our method allows high speed-up at the cost of a small error. See Table 2.

4.2 Performances

We have compared the efficiency of our method to contact correction based on the exact *compliance* use, and exact displacement integration. The results are presented in Table 2. Our method allows a huge speed-up depending on the number of contact points.

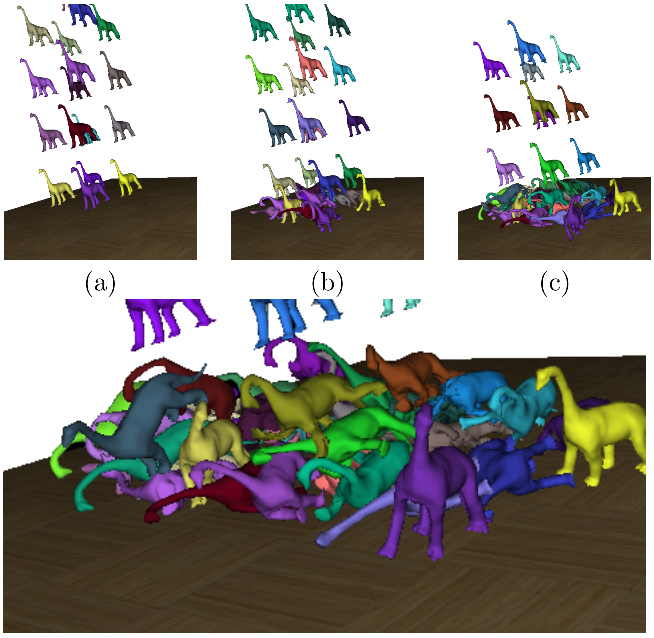
Moreover, our method does not depend on the complexity of the model used to simulate the object deformation. **If the same number of contact points are involved, our method is as fast on a simple model than on a complex one**. On the opposite, using the exact *compliance* would have required to solve a system that depends on the model complexity.

4.3 Mixed deformable and rigid links

This example illustrates that we can handle various types of bodies. We mixed deformable and rigid rings to obtain the following contact configurations: Deformable - Deformable, Deformable - Rigid, Rigid - Rigid. See Figure 3. All these contact configurations are handled efficiently and in a stable way by our method.

4.4 Dinosaurs waterfall

To evaluate our method stability and robustness when a huge number of contacts are involved, we simulated the fall of 40 dinosaurs ($\sim 1K$ tetrahedrons each). This simulation experiences drastic changes in a huge number of



(a) (b) (c)

(d)

Fig. 4 (a, b, c) : The dinosaurs accelerate during a free fall ; (d) : The dinosaurs collapsing and collision generate many contact and friction points.

contact and friction points ($\sim 2K$ friction contacts). See Figure 4. In this demanding case, the simulation remains stable and its computation is quite fast. See Table 2.

5 Conclusion and future work

This paper introduces several contributions related to contact modeling between various types of objects commonly used in computer graphics. First, we have presented a time-stepping scheme which allows simulation of both deformable and rigid objects. We then proposed a contact processing approach where the motion of each object is decomposed in a free motion and a corrective motion. The corrective motion uses a contact model which follows Signorini and Coulomb laws for contact and friction modeling but uses an approximation of the behavior model of the corresponding object. In our case, contact forces and displacements are computed using an approximate compliance. We show how this approximate compliance can be computed for a geometrically non-linear deformable model. As a result, our approach dramatically speeds-up the contact correction, and allows scenes with many complex interacting objects. Our approach guarantees no interpenetration at the end of the time step, and only introduces a minimal error in the objects motion.

Since the proposed method is independent of the deformation model used, and we are planning to extend it to other deformable models, in particular models used for the simulation of cloth, where many contacts occur. Regarding performances, our method can be easily adapted to SIMD techniques and therefore we plan to port it on the GPU to increase further performances.

References

1. Acary, V., Brogliato, B.: Numerical Methods for Non-smooth Dynamical Systems: Applications in Mechanics and Electronics, *Lecture Notes in Applied and Computational Mechanics*, vol. 35. Springer Verlag (2008)
2. Anitescu, M., Potra, F., Stewart, D.: Time-stepping for three-dimentional rigid body dynamics. Computer Methods in Applied Mechanics and Engineering (177), 183–197 (1999)
3. Baraff, D.: Analytical methods for dynamic simulation of non-penetrating rigid bodies. SIGGRAPH '89 **23**(3), 223–232 (1989)
4. Baraff, D.: Fast contact force computation for nonpenetrating rigid bodies. In: SIGGRAPH' 94, pp. 23–34. Addison Wesley (1994)
5. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: SIGGRAPH '98, pp. 43–54. ACM Press, New York, NY, USA (1998)
6. Bridson, R., Fedkiw, R., Anderson, J.: Robust treatment of collisions, contact and friction for cloth animation. In: SIGGRAPH '05, p. 2. ACM, New York, NY, USA (2005)
7. 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), 36–47 (2006)
8. Galoppo, N., Otaduy, M.A., Mecklenburg, P., Gross, M., Lin, M.C.: Fast simulation of deformable models in contact using dynamic deformation textures. In: SCA '06, pp. 73–82. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006)
9. Georgii, J., Westermann, R.: A multigrid framework for real-time simulation of deformable bodies. Computers & Graphics **30**(3), 408–415 (2006)
10. Hauth, M., Straßer, W.: Corotational simulation of deformable solids. In: WSCG 2004, pp. 137–145 (2004)
11. Mirtich, B., Canny, J.: Impulse-based dynamic simulation. In: WAFR: Proceedings of the workshop on Algorithmic foundations of robotics, pp. 407–418. A. K. Peters, Ltd., Natick, MA, USA (1995)
12. Mirtich, B., Canny, J.: Impulse-based simulation of rigid bodies. In: SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics, pp. 181–ff. ACM, New York, NY, USA (1995)
13. Moreau, J.J., Jean, M.: Numerical treatment of contact and friction: the contact dynamics method. Engineering Systems Design and Analysis **4**, 201–208 (1996)
14. Müller, M., Dorsey, J., McMillan, L., Jagnow, R., Cutler, B.: Stable real-time deformations. In: SCA '02, pp. 49–54. ACM Press, New York, NY, USA (2002)
15. Murty, K.: Linear Complementarity, Linear and Nonlinear Programming. Internet Edition (1997)
16. Nealen, Andrew, Muller, Matthias, Keiser, Richard, Boxerman, Eddy, Carlson, Mark: Physically based deformable models in computer graphics. Computer Graphics Forum **25**(4), 809–836 (2006)
17. Pauly, M., Pai, D.K., Guibas, L.J.: Quasi-rigid objects in contact. In: SCA '04, pp. 109–119. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2004)
18. Terzopoulos, D., Witkin, A.: Physically based models with rigid and deformable components. IEEE Computer Graphics and Applications. **8**(6), 41–51 (1988)