An Energy-Based Approach for Contact Force Computation

François Faure

Francois.Faure@imag.fr, iMAGIS/GRAVIR-IMAG†, BP 53, 38041 Grenoble Cedex 9, France

Abstract

A new approach for computing resting contact forces between solids is presented, which handles both static and sliding friction, while avoiding the problems arising from previous approaches. Each iteration provides values for all unknowns and needs a constant computation time. The first iteration is a global dynamic solution involving inertia and external forces. The subsequent iterations consist of global redistributions of energy through the solids in order to restrict the values within correct bounds, leading to a progressive refinement of the solution. This allows the termination of the computation when a user-defined level of precision is reached. Convergence is proven in the frictionless case.

The method is concise and is easily usable in a motion control system. It was implemented and tested on some examples involving simultaneously static and sliding friction along with motion control.

Keywords: contact forces, friction, dynamics

1. Introduction

The simulation of rigid body motion has been widely addressed in computer graphics research. The dynamics equations are now well-known and used in the computer graphics community. Simulations involving several bodies with various kinds of kinematic joints can be achieved by the solution of a linear equation set. Unfortunately, these sets of equations fully describe the constraints only if the joints provide bilateral constraints. For joints of this kind, relative motion is allowed along some directions, and forbidden along the others. Bilateral constraints do not reflect the constraints between two solids in the general case, since contact forces prevent the solids from interpenetrating but not from going away from each other. Another problem is the computation of friction forces, since tangential forces are bounded in intensity, or constrained in force and direction in some cases. As a consequence, contact accelerations and forces, while satisfying the dynamics equations, must also satisfy some inequalities from which problems arise.

The computation of contact forces between solids is one of the most difficult problems in solid mechanics, since it requires the solution of a system of linear equations while satisfying inequalities in the unknowns. This was proven NP-complete [2] and several iterative methods have been proposed to solve it. It is still a very important issue in computer animation, since none of the previous approaches is completely satisfying, as we shall see in the following.

The first attempts to address this issue formulated the problem as one of minimization [8]. In practice, this requires strong knowledge of numerical optimization and a non-specialist would have difficulty obtaining stable solutions for a wide range of problems. Furthermore, these methods are compute-intensive and do not handle friction.

† iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble, Université Joseph Fourier.

Baraff proposed an adaptation of Dantzig's algorithm [4], which is related to pivoting methods for solving linear and quadratic programming problems. The original method is well-suited for non-interpenetrating constraints, and a modification was included for the computation of tangential forces [3]. This was the first algorithm which simultaneously handled all kinds of friction. It is also rather simple to implement, but some problems remain. In particular, convergence was not proven with friction and we have no guarantee that in this case the algorithm would terminate. The other problem is the control of the computation time, which is an important issue in real-time simulations. The structure of the algorithm prevents us from bounding the number of iterations. The forces and accelerations are successively calculated while modifying the previously computed values. The computation ends when the last value is found. Since for the computation of each value, an unpredictable number of iterations (unless the contacts are frictionless) is necessary to maintain the previously computed values within the correct bounds, it is impossible to get a reliable bound on computation time.

The method we present here avoids the problems arising from previous approaches. Each iteration provides values for all unknowns and needs a constant computation time. The first iteration is a global dynamic solution involving inertia and external forces. The subsequent iterations consist of global redistributions of energy through the solids in order to restrict the values within correct bounds, leading to a progressive refinement of the solution. It is compatible with articulated bodies and, as in some previous simulation algorithms[7], the optimal number of unknowns is considered, since they all correspond to the degrees of freedom of the system. These features make the method potentially usable in a real-time simulation system.

2. Contact model and mechanical equations

Though dynamics equations have been presented in previous papers[7], we consider it necessary to explain how to create our equation system, since it is not a classical formulation of the dynamics equations.

We are considering a set of solids and a set of distinct contact points p_i. The problem of surface contact and degenerate contact points was addressed by Baraff[1]. We do not deal with the problem of collisions. This topic is already treated in the literature[6, 9, 1].

The computation of forces and accelerations requires the tensor expression of three operators related to force, acceleration and velocity. In this section, we first define the tensor expressions of the mechanical operators. We describe our contact model, then considerations concerning variable dependencies which lead to a set of kinematic relations. The last part deals with the expression of energy conservation. More details can be found in any standard text on elementary mechanics [11].

2.1. Mechanical operators

The mechanical operators relate velocity, dynamical operator and contact forces to the degrees of freedom of the set of bodies. For a given solid s, it is convenient to express all these operators at the same point, and we choose to do it at the center of mass. They all have six components, and we use the brackets notation for clarity. The corresponding matrices have six rows and their number of columns is the total number of the degrees of freedom.

2.1.1. Force

The sum $\{F_s\}$ of the joint forces applied on a solid can be expressed as the product of a matrix $\boldsymbol{F}_s$ (related to geometry) with the vector $\boldsymbol{f}$ composed of the forces and torques in the different degrees of freedom. The first three rows correspond to total forces in three independent directions, while the last three rows correspond to the torques at the center of mass.

2.1.2. Velocity

The velocity field $\{V_{s/r}\}$ of a solid s with respect to the coordinate system r can be expressed as the product of a matrix $\boldsymbol{V}_{s/r}$ related to geometry with a vector $\dot{\boldsymbol{q}}$ composed of the velocities of the degrees of freedom. The first three rows correspond to linear velocities at the center of mass, and the last three rows correspond to angular velocities.

2.1.3. Dynamic operator

The dynamic operator $\{D_{s/r}\}$ of a solid s with respect to a coordinate system r is related to the inertia and acceleration of the solid. We can write it as $\boldsymbol{D}_{s/r} \cdot \ddot{\boldsymbol{q}} + \boldsymbol{d}$, where $\boldsymbol{D}$ is related to inertia, $\ddot{\boldsymbol{q}}$ is the acceleration vector, and $\boldsymbol{d}$ is the inertia related to the velocities. The first three rows correspond to linear inertia, while the last three rows correspond to angular inertia at the center of mass.

2.2. Contact model

We use a model [5] in which a contact between two solids s_1 and s_2 will be represented as a contact point p_{12} and three orthonormal vectors $\boldsymbol{i}_{12}$, $\boldsymbol{j}_{12}$ and $\boldsymbol{k}_{12}$, $\boldsymbol{i}_{12}$ being collinear with the normal at the surfaces, as shown on figure 1.

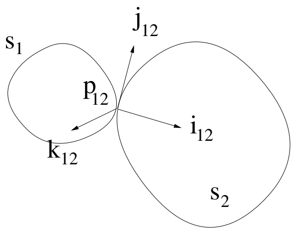

Figure 1: *Two bodies s_1 and s_2 in contact*

There may be several contact points between two solids. Linear and planar contacts require respectively two and three of them. We can represent

$$\{V_{2/1}\} = \{\dot{x}_{21}\boldsymbol{i}_{12} + \dot{y}_{21}\boldsymbol{j}_{12} + \dot{z}_{21}\boldsymbol{k}_{12},\ \ \dot{\alpha}_{21}\boldsymbol{i}_{12} + \dot{\beta}_{21}\boldsymbol{j}_{12} + \dot{\gamma}_{21}\boldsymbol{k}_{12}\}$$

where the upper dot denotes time derivation, $\dot{x}_{21}, \dot{y}_{21}, \dot{z}_{21}$ are the components of linear velocity at point p_{12} and $\dot{\alpha}_{21}, \dot{\beta}_{21}, \dot{\gamma}_{21}$ are the components of angular velocity. Similarly, we can represent the corresponding force operator as:

$$\{F_{s_1 \to s_2}\} = \{f_{x12}\boldsymbol{i}_{12} + f_{y12}\boldsymbol{j}_{12} + f_{z12}\boldsymbol{k}_{12},\ \ t_{x12}\boldsymbol{i}_{12} + t_{y12}\boldsymbol{j}_{12} + t_{z12}\boldsymbol{k}_{12}\}$$

where $f_{x12}, f_{y12}, f_{z12}$ are the components of the force and $t_{x12}, t_{y12}, t_{z12}$ are the components of the torque applied to s_2 by s_1 in the joint. All classical joints can also be modeled in this way. Their contact surfaces are designed to allow only some degrees of freedom. For example, a perfect universal joint [7] will be modeled as:

$$\begin{aligned}\{V_{2/1}\} &= \{\mathbf{0},\ \ \dot{\alpha}_{21}\boldsymbol{i}_{12} + \dot{\beta}_{21}\boldsymbol{j}_{12} + \dot{\gamma}_{21}\boldsymbol{k}_{12}\} \\ \{F_{s_1 \to s_2}\} &= \{f_{x12}\boldsymbol{i}_{12} + f_{y12}\boldsymbol{j}_{12} + f_{z12}\boldsymbol{k}_{12},\ \ \mathbf{0}\}\end{aligned}$$

The specific problem of external contacts is that they have to satisfy certain inequalities. To prevent interpenetration and attraction, we must satisfy the relations of frictionless contact:

$$\dot{x}_{21} \geq 0, \quad \ddot{x}_{21} \geq 0, \quad f_{x12} \geq 0 \tag{1}$$

Since we do not consider the problem of collision impact, we assume that the first inequality is satisfied. If we assume that the contact is not frictionless, we must add some additional relations:

- if the relative tangential velocity of the surfaces is null, the ratio between tangential and normal force must be less than or equal to the *friction (Coulomb) coefficient* μ_s. This yields:

$$f_{y12}^2 + f_{z12}^2 \ \leq\ \mu_s^2 . f_{x12}^2 \tag{2}$$

This case is usually called *static friction*.

- in the other case, the bodies are sliding on each other with a relative velocity $V = \sqrt{\dot{y_{21}}^2 + \dot{z_{21}}^2}$. The tangential force is collinear with the tangential velocity with opposite direction, and its strength is proportional to the normal force. The relations are:

$$f_{y12} = -\mu_d.f_{x12}.\dot{y}_{21}/V, \quad f_{z12} = -\mu_d.f_{x12}.\dot{z}_{21}/V \tag{3}$$

This case is usually called *sliding* or *dynamic friction*. The coefficient μ_d is always smaller than μ_s.

2.3. Variable dependencies

Since a velocity operator completely describes the relative velocity between two solids, all its different possible expressions have to be equivalent. Such different expressions occur in so-called *closed kinematic loops*. Figure 2 shows an example of such a case: the velocity operator $\{V_{3/1}\}$ can either be expressed through contact at p_{13} or, by using the transitivity of the velocity operator, through contacts at p_{12} and p_{23}.

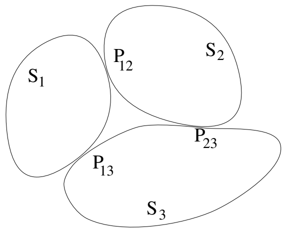

Figure 2: *A closed kinematic loop*

This implies that some relations between the variables must be satisfied. These relations can be expressed as:

$$\boldsymbol{K} \cdot \dot{\boldsymbol{q}} = \boldsymbol{0}$$

where $\dot{\boldsymbol{q}}$ is the velocity vector of all the degrees of freedom, and $\boldsymbol{K}$ a matrix representing the equivalence between the different expressions of the velocity. The relation is the same for accelerations. The solutions of this system can be expressed as:

$$\dot{\boldsymbol{q}} = \boldsymbol{L} \cdot \dot{\boldsymbol{l}}$$

where the components of $\dot{\boldsymbol{l}}$ are the independent kinematic variables of the system (also called *Lagrangian variables*), and $\boldsymbol{L}$ the transpose of an orthogonal projection matrix. If the joints are all external contacts, the number of independent variables is six times the number of solids, since all displacements are *a priori* possible, corresponding to six degrees of freedom per solid with respect to the absolute coordinate system.

2.4. Theorem of Energy

In this section, we explain how we create our equation system using the previously defined operators. This equation system corresponds to energy conservation. Moreover, we add some relations which are specific to sliding friction.

2.4.1. General expression

The Law of Conservation of Energy expresses the energy balance of a body, or of a system of bodies. For a mechanical system, its general expression is:

$$P_f = P_a$$

where P_f is the power given by the forces, and P_a the power of the quantities of acceleration. For a single body, its expression is:

$$\{V_{s/r}\}\{F_s\} = \{V_{s/r}\}\{D_{s/r_g}\}$$

where r is any coordinate system, but r_g is *imperatively* a Galilean coordinate system, similarly to the absolute coordinate system of our scene. We must first emphasize the fact that the equations associated with each solid are not independent, since $\{F_{s_1 \to s_2}\} = -\{F_{s_2 \to s_1}\}$. Thus,

$$\{V_{s_1/r}\}\{F_{s_2 \to s_1}\} + \{V_{s_2/r}\}\{F_{s_1 \to s_2}\} = \{V_{s_2/s_1}\}\{F_{s_1 \to s_2}\}$$

This means that in joints, the forces that are not related to a degree of freedom do not appear in the system of equations corresponding to the sum of the dependent systems of equations. Thus, the equation system corresponding to the global energy is a square system and allows us to find the motion unknowns globally, with the optimal number of unknowns. As far as external contacts are concerned, this approach does not lead to any reduction of the number of unknowns, since each contact provides six degrees of freedom with their associated forces and accelerations. Nonetheless, it leads to consequent reduction of computation time when articulated bodies (with less than six degrees of freedom at each joint) are also involved in the scene.

2.4.2. Tensor expression and additional constraints

We want to create the system of equations corresponding to the global conservation of energy. This corresponds to the equation:

$$\begin{gathered}\sum_i \{V(s_i/s_j)\}\{F_{s_j \to s_i}\} + \sum_i \{V(s_i/r_g)\}\{\tilde{F}_{r_g \to s_i}\} \\ = \\ \sum_i \{V(s_i/r_g)\}\{D(s_i/r_g)\} + \sum_i \{V(s_i/r_g)\}\{\tilde{D}(s_i/r_g)\}\end{gathered}$$

where the terms correspond, from left to right and top to bottom, to the powers of contact forces (unknown) , exterior forces (known), time-dependent acceleration (unknown), velocity-dependent acceleration (known). For each solid s, we write the tensor expression of this relation as:

$$\dot{\boldsymbol{q}}^t \cdot [\boldsymbol{V}_s^t \cdot \boldsymbol{F}_s] \cdot \boldsymbol{f} + \dot{\boldsymbol{q}}^t \cdot \boldsymbol{V}_s^t \cdot \tilde{\boldsymbol{f}} = \dot{\boldsymbol{q}}^t \cdot [\boldsymbol{V}_s^t \cdot \boldsymbol{D}_{s/r_g}] \cdot \ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^t \cdot \boldsymbol{V}_s^t \cdot \boldsymbol{d} \tag{4}$$

where $\boldsymbol{f}$ are the forces corresponding to the degrees of freedom, $\tilde{\boldsymbol{f}}$ the influence of external forces such as weight and air friction, $\boldsymbol{d}$ the velocity-dependent part of the dynamic operator, and $\dot{\boldsymbol{q}} = \boldsymbol{L} \cdot \dot{\boldsymbol{l}}$. This is a scalar expression. The d'Alembert principle (also called virtual work or virtual power principle) states that this expression is true for *any* velocity vector $\dot{\boldsymbol{l}}$. This leads to a set of linear equations:

$$[\boldsymbol{L}^t \cdot \boldsymbol{V}_s^t \cdot \boldsymbol{F}_s] \cdot \boldsymbol{f} + [\boldsymbol{L}^t \cdot \boldsymbol{V}_s^t] \cdot \tilde{\boldsymbol{f}} = [\boldsymbol{L}^t \cdot \boldsymbol{V}_s^t \cdot \boldsymbol{D}_{s/r_g}] \cdot \ddot{\boldsymbol{q}} + [\boldsymbol{L}^t \cdot \boldsymbol{V}_s^t] \cdot \boldsymbol{d}$$

In the following, we will call *virtual works* the vectors involved in this equation. In its usual formulation, the relation $\ddot{\boldsymbol{q}} = \boldsymbol{L} \cdot \ddot{\boldsymbol{l}}$ is also used. We will not use this formulation, because we also have inequalities to satisfy, and thus we need to address each unknown individually. Nevertheless, this relation must be satisfied, and so we join the kinematic matrix $\boldsymbol{K}$ set in section 2.3 to the matrix corresponding to the accelerations. At each sliding contact, we have also two relations corresponding to constraints (3) about direction and intensity of the tangential force. This yields two equations for each sliding contact c, which we can gather as:

$$\boldsymbol{S}_c \cdot \boldsymbol{f} = \boldsymbol{0} \tag{5}$$

We can gather all these relations in the global equation system:

$$\begin{pmatrix} \boldsymbol{P}_{vf} \\ \boldsymbol{0} \\ \boldsymbol{S} \end{pmatrix} \cdot \boldsymbol{f} + \begin{pmatrix} \boldsymbol{p}_{vf} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_{vd} \\ \boldsymbol{K} \\ \boldsymbol{0} \end{pmatrix} \cdot \ddot{\boldsymbol{q}} + \begin{pmatrix} \boldsymbol{p}_{vd} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \tag{6}$$

where

$$\boldsymbol{P}_{vf} = \boldsymbol{L}^T \cdot \sum_i [\boldsymbol{V}_{S_i}^T \cdot \boldsymbol{F}_{S_i}]$$

is the tensor operator of the virtual work of forces,

$$\boldsymbol{P}_{vd} = \boldsymbol{L}^T \cdot \sum_i [\boldsymbol{V}_{S_i}^T \cdot \boldsymbol{D}_{S_i/R_g}]$$

is the tensor operator of the virtual work of accelerations, $\boldsymbol{p}_{vf}$ the virtual work of external forces, $\boldsymbol{p}_{vd}$ the virtual work of the velocity-dependent (thus known) acceleration, and the $\mathbf{0}$'s are appropriate-size zero vectors or matrices. This system of equations (6) gathers all the knowledge we have about the problem before finding a solution.

2.5. More about solution

If all the forces are given, we can compute all the accelerations: by computing the inverse acceleration virtual work matrix, we can obtain the classical formulation

$$\ddot{\boldsymbol{q}} = \boldsymbol{A} \cdot \boldsymbol{f} + \boldsymbol{b} \quad with \quad \boldsymbol{A} = (\sum_i [\boldsymbol{V}_i^t \cdot \boldsymbol{D}_{S_i/R_g}])^{-1} \cdot (\sum_i [V_{S_i}^T \cdot \boldsymbol{F}_{S_i}]) \tag{7}$$

Given every acceleration, there may be no solution since we have more independent equations than unknowns. Since this problem arises from sliding contacts, we proceed in this way: for each sliding contact, we either impose all its forces (satisfying rel.(5)), or we impose only acceleration in the normal direction. This method provides a solvable equation system if, for the other degrees of freedom, we impose either force or acceleration.

The system may have a set of solutions instead of a unique one. Nevertheless, the independent variables of our solution set concern only forces (e.g. in the case of three coplanar forces between two solids). From a mathematical point of view, all these solutions are equivalent. From a physical point of view, the solid body is just a useful mathematical model, and we know that in reality small deformations occur, especially around the contact points. The corresponding deformation energy involved is always the minimum among all possible solutions. For a linear elastic deformation model, this energy is proportional to $\boldsymbol{f^2}$. The consequence is that even when dealing with pure theoretical solid bodies, we can consider that choosing the smallest forces is a good choice. It is also a very convenient strategy, since many of the pseudo-inverse packages provide it explicitly [10].

3. Force and acceleration computation

We have an equation system (6) which allows us to find a solution if we can provide a sufficient set of known values. The problem is that for most degrees of freedom, especially those involved in contacts, neither force nor acceleration is known. That is why we have to use an iterative approach starting from a set of assumptions.

3.1. Formulation of the problem

The set of assumptions must at least satisfy the contact constraints. We call a set of forces and accelerations *kinematically consistent* if it satisfies the acceleration constraints, and *statically consistent* if it satisfies the force constraints (equations 1,2,3). If we find a result that is both kinematically and statically consistent, and satisfies the energy equations, then we have the solution we are looking for. From this point of view, the problem becomes: *find a kinematically consistent set of accelerations and a statically consistent set of forces that satisfy the global equation system (6).*

3.2. Iterative solution

We start with a set $\boldsymbol{f}$ of forces and a set $\boldsymbol{a}$ of accelerations all set to zero. At each iteration, we define the set H of unknowns about which we make assumptions. The appropriate-sized vector $\boldsymbol{v}$ contains the corresponding force and acceleration values. The complement $\bar{H}$ is the set of unknowns which values we calculate. These computed values are gathered in the vector $\bar{\boldsymbol{v}}$. We then build and solve an equation system $\boldsymbol{M} \cdot \bar{\boldsymbol{v}} + \boldsymbol{b} = \mathbf{0}$. All the columns of equation system (6) corresponding to unknown values are written in matrix $\boldsymbol{M}$, while all columns corresponding to known values are added to vector $\boldsymbol{b}$. We then add $\boldsymbol{v}$ and $\bar{\boldsymbol{v}}$ to the current solution.

In practice, at the first iteration, H is chosen such as to impose null tangential and normal acceleration at static contact points, and null normal acceleration at sliding contact points. The vector $\boldsymbol{v}$ is thus null and the vector $\boldsymbol{b}$ corresponds to the rotation inertia and external forces: $\boldsymbol{b} = \boldsymbol{p}_{vf} - \boldsymbol{p}_{vd}$. External influences are now taken into account and during the following iterations $\boldsymbol{b}$ will only correspond to the compensation of inconsistent internal forces.

At each new iteration, the vector $\boldsymbol{v}$ is used to compensate the inconsistent values of the current solution ($\boldsymbol{f}, \boldsymbol{a}$), while $\bar{\boldsymbol{v}}$ is used to recover energy balance after compensation, following the pseudocode given in figure 3. The hat is used to denote the appropriate component of vector $\boldsymbol{v}$.

$\boldsymbol{Compute\_forces\_and\_accelerations}$ **:**

$\boldsymbol{f} \leftarrow \boldsymbol{0}, \quad \boldsymbol{a} \leftarrow \boldsymbol{0}$
solve $\boldsymbol{M} \cdot \bar{\boldsymbol{v}} + \boldsymbol{b} = \boldsymbol{0}$ *// initial system*
while $\|\bar{\boldsymbol{v}}\| > \epsilon$
    add $\bar{\boldsymbol{v}}$ to $\boldsymbol{f}$ and $\boldsymbol{a}$
    $\boldsymbol{v} \leftarrow \boldsymbol{0}$
    for each contact i:
        if $a_{xi} \in \bar{H}$ *// normal acceleration computed*
            if $a_{xi} < 0$
                $\boldsymbol{v}[\hat{a_{xi}}] \leftarrow -a_{xi}$
                if i is a sliding contact
                    put a_{xi} in H
                    put $f_{xi}, f_{yi}, f_{zi}, a_{yi}, a_{zi}$ in $\bar{H}$
                else
                    put a_{xi} in H
                    put f_{xi} in $\bar{H}$

        else *// normal force computed*
            if $f_{xi} < 0$
                put f_{xi}, f_{yi}, f_{zi} in H
                put a_{xi}, a_{yi}, a_{zi} in $\bar{H}$
                $\boldsymbol{v}[\hat{f_{xi}}] \leftarrow -f_{xi}$
                $\boldsymbol{v}[\hat{f_{yi}}] \leftarrow -f_{yi}$
                $\boldsymbol{v}[\hat{f_{zi}}] \leftarrow -f_{zi}$
            else
                if $f_{yi}^2 + f_{zi}^2 > \mu_{si}^2 . f_{xi}^2$
                    put f_{yi}, f_{zi} in H
                    put a_{yi}, a_{zi} in $\bar{H}$
                    $f_t \leftarrow \sqrt{f_{yi}^2 + f_{zi}^2}$ *// tangential force*
                    $f_{tm} \leftarrow \mu_{si} . f_{xi}$ *// maximum tangential force*
                    $\boldsymbol{v}[\hat{f_{yi}}] \leftarrow (f_{tm}/f_t - 1) . f_{yi}$
                    $\boldsymbol{v}[\hat{f_{zi}}] \leftarrow (f_{tm}/f_t - 1) . f_{zi}$

    add $\boldsymbol{v}$ to $\boldsymbol{f}$ and $\boldsymbol{a}$ *// accelerations and forces are now consistent*
    create new matrix $\boldsymbol{M}$ and vector $\boldsymbol{b}$
    solve $\boldsymbol{M} \cdot \bar{\boldsymbol{v}} + \boldsymbol{b} = \boldsymbol{0}$ *// we should add $\bar{\boldsymbol{v}}$ to recover energy balance*

Figure 3: *The iterative solution algorithm*

Because of equations (5), the only possible inconsistencies at sliding contact points are attractive forces.

Since each correction leads to a change in assumptions, the matrix $\boldsymbol{M}$ and the vector $\boldsymbol{b}$ are different at each iteration. Since H and $\bar{H}$ are complement sets, putting a value in one set also means removing it from the other.

A simple example of an iterative solution is shown in figure 4. A force is applied on a bar and we want to compute the contact forces and accelerations. The elements of $\boldsymbol{v}$ are written above the bar while the elements of

$\bar{\boldsymbol{v}}$ are written under it. Dashed lines lines denote forces while solid lines denote accelerations. The first solution (1) provides two negative forces. The correction (2) is designed to compensate them, while the assumptions on other contacts are not changed, since they led to consistent values. A negative force appears at another contact in (3). A new correction leads to the correct solution. The bar pivots around the righthand contact. We can now compute the new velocities and positions, and consider the next time step.

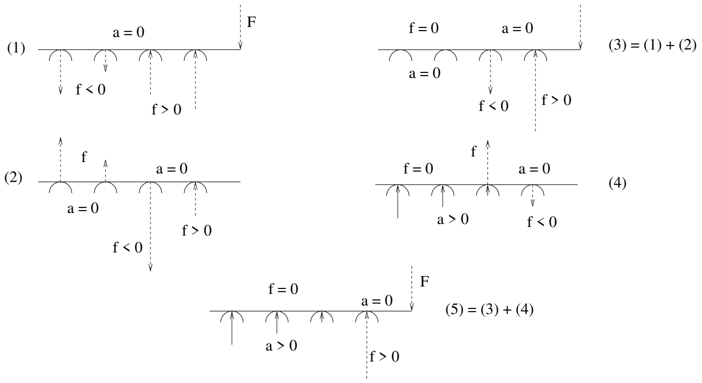

Figure 4: *Example of a force F applied on a bar with four contacts*

The original feature of this algorithm is to create the solution as a sum of individual solutions. By adding the successive solutions, each step is the computation of values that will compensate, by summation, the previous inconsistent values. The global energy equation system remains satisfied, since the constant vectors of the equations system (6) were set to zero after the first solution.

We can see this method as the systematic *redistribution* of the total inconsistent energy throughout the rest of the system, leading to a progressive refinement. New errors may have been generated but their amount in terms of virtual work decreases, as shown in the next subsection. Since a small virtual work correspond to a small influence, the computation may be stopped when the error (the virtual work of inconsistent values) is smaller than a user-selected value, just after adding the correction vector $\boldsymbol{v}$. This provides a solution that is both kinematically and statically consistent, with a small error on energy conservation. We can also stop the computation when a maximum number of iterations is reached.

3.3. Convergence

Convergence is achieved if the inconsistent forces and accelerations decrease to zero. At each iteration, the addition of $\boldsymbol{v}$ to the solution compensates all the previous inconsistent values. The new inconsistent values come from the addition of $\bar{\boldsymbol{v}}$. It is trivial to show that the sum of consistent values is consistent, the total inconsistence is thus less than or equal to the inconsistent part of $\bar{\boldsymbol{v}}$.

Since the square matrices $\sum_i \boldsymbol{V}_{S_i}^t \cdot \boldsymbol{F}_{S_i}$ and $\sum_i \boldsymbol{V}_{S_i}^t \cdot \boldsymbol{D}_{S_i/R_g}$ are positive by construction and $\boldsymbol{L}^t$ is an orthogonal projection matrix, any absolute value reduction of a force or acceleration leads to a decrease of the norm of the corresponding virtual work. We also emphasize that after the first iteration, the constant vectors of equations (6) are null so the virtual work of $\bar{\boldsymbol{v}}$ equals the virtual work of $\boldsymbol{v}$. Thus, the virtual work of the new inconsistent values is less than or equal to the virtual work of the previous ones (compensated by $\boldsymbol{v}$). We must now address the question whether it can decrease to a non-null value.

Now, let us write $\boldsymbol{v} = (\boldsymbol{f}, \boldsymbol{a})$ where $\boldsymbol{f}$ is the set of forces and $\boldsymbol{a}$ the set of accelerations in $\boldsymbol{v}$, and $\bar{\boldsymbol{v}} = (\bar{\boldsymbol{a}}, \bar{\boldsymbol{f}})$. Using equation 7 with null constant vectors, we can write:

$$A \cdot \begin{pmatrix} \boldsymbol{f} \\ \bar{\boldsymbol{f}} \end{pmatrix} = \begin{pmatrix} \bar{\boldsymbol{a}} \\ \boldsymbol{a} \end{pmatrix}$$

with all consistent values in $\boldsymbol{f}$ and $\boldsymbol{a}$ and, since $\boldsymbol{A}$ is positive semidefinite: $\boldsymbol{f} \cdot \bar{\boldsymbol{a}} + \bar{\boldsymbol{f}} \cdot \boldsymbol{a} \geq 0$. This scalar expression has to be strictly positive when $\boldsymbol{A}$ is positive definite. Semidefiniteness occurs when several combinations of forces can lead to the same result, as with three coplanar forces between the same solids. Since, in this case, our strategy is to choose always the minimum forces, positiveness is ensured.

Thus, some values of $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{a}}$ have to be in the same direction as their corresponding consistent accelerations or forces. In the frictionless case, this implies that if $\boldsymbol{v}$ is consistent, $\bar{\boldsymbol{v}}$ can not be totally inconsistent. Thus, the virtual work corresponding to the new inconsistent values has to be smaller than the previous one (corresponding to $\boldsymbol{v}$). The matrix $\boldsymbol{A}$ depends only on the solids and their positions. There is thus a minimum decrease rate ϵ_A of the inconsistent virtual work norm among all the possible sets of assumptions H, so the norm of the inconsistent virtual works along the iterations is bounded above by a geometric series in $1 - \epsilon_A$ which converges to zero.

We have now shown that in the frictionless case, the inconsistent virtual works decrease to zero. Since the matrix $\boldsymbol{P}_{vd}$ of equation (6) is always invertible (case of forward dynamics), the accelerations have to decrease to null values. The forces may not, but the nullity of their virtual work would mean that they have no influence. This would simply be a case where they compensate each other with a null sum. Furthermore, this situation should not occur if we apply the minimum-force-strategy presented in section 2.5.

In the presence of friction, the problem is that the compensation of an attractive force also requires the compensation of the associated tangential force. The values of $\bar{\boldsymbol{v}}$ may then have to be totally compensated, leading to a null decrease of the virtual work. This case is encountered in inconsistent configurations[2]. The result is an endless alternation between a penetrating acceleration and an attractive force.

4. Implementation and results

The algorithm is easily implementable in a physically-based articulated solids simulation system. It allows the simultaneous handling of external contacts and ordinary joints which, unlike contacts, need no assumption. An interesting feature is that the algorithm is also well-suited to the handling of motion control, by simply joining the desired acceleration relations to the system (6). The algorithm was tested on several examples involving static and dynamic friction. Fast decrease of error was observed. The four-ball solid showed on figure 5 at the end of this document is an example of static and sliding friction involving four parallel and four coplanar forces. The friction coefficient of the contact between the plane and the darkest ball is set to 0.7, whereas the others are set to 0.2. At each time step, the iterative solution of this over-constrained device requires between one iteration (with four sliding contacts) and five (in the case of one static and three sliding contacts) to converge to a consistent solution. In the case where five iterations were required, it was possible to end the computation at the third iteration without any perceptible difference in the result. An initial rotation velocity of the plane was given, and the torque which is necessary to keep it constant (null acceleration) is also computed at each time step. This example runs in real-time on an SGI Indigo2 workstation.

5. Conclusion

We have presented a new approach to deal with the problem of contact forces. It is based on energy transfer between the bodies in contact. Each step of the algorithm consists of a global solution which satisfies the conservation of energy. The sum of the solutions converges towards a result which satisfies both the conservation of energy and the inequality constraints in the case of frictionless contact. Furthermore, it is naturally compatible with motion control involving any kind of joints.

This new approach combines the advantages of the different previous approaches. As with minimization, we can easily terminate the computation when either desired precision or an allowed computation time is reached. Like Baraff's approach[3], it simultaneously handles all types of friction with easy implementation, and convergence is proven in the frictionless case.

References

1. David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 223–232, July 1989.
2. David Baraff. Coping with friction for non-penetrating rigid body simulation. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 31–40, July 1991.
3. David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 23–34. ACM SIGGRAPH, ACM Press, July 1994.
4. R.W. Cottle and G.B. Dantzig. Complementary pivot theory of mathematical programming. *Linear algebra and its applications*, 1:103–125, 1968.
5. François Faure. Modélisation cinématique du contact pour la dynamique inverse *(in french)*. In *Troisièmes journées de l'Association Française d'Informatique Graphique*, pages 15–22, November 1995.
6. James K. Hahn. Realistic animation of rigid bodies. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 299–308, August 1988.
7. Paul M. Isaacs and Michael F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 215–224, July 1987.
8. P. Lötstedt. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical computing*, 5(2):370–393, 1984.
9. Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 289–298, August 1988.
10. Press, Teukolski, Vetterling, and Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.
11. J. Wittenberg. *Dynamics of Systems of Rigid Bodies.* B.G. Teubner, Stuttgart, Germany, 1977.

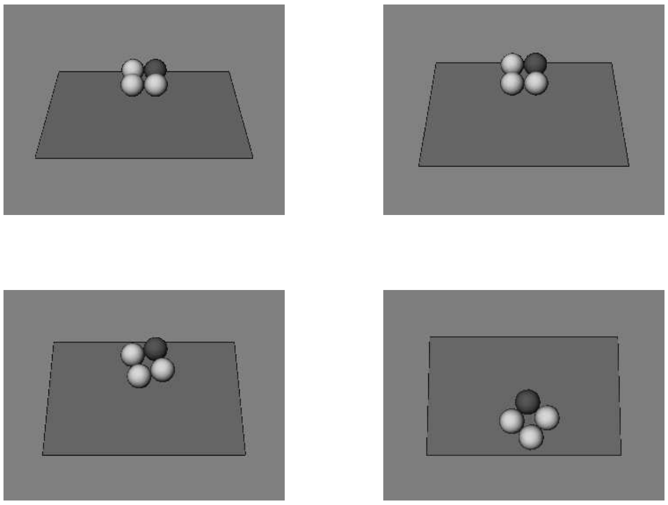

Figure 5: *A four-contact solid sliding on a rotating plane*