coneural
Center for Cognitive and Neural Studies

Simple iterative algorithm for contact force computation

Mihai Preda
Procod SRL, Galaţi, Romania
mihai@procod.com

Răzvan V. Florian
Center for Cognitive and Neural Studies (Coneural),
Cluj-Napoca, Romania
florian@coneural.org

Technical Report Coneural-05-01
Version 1.0
August 27, 2005

Abstract

We describe a simple iterative algorithm for contact force computation, similar to the Gauss-Seidel method for solving linear systems. Since this method can be implemented in a few lines of code, it should be recognized as the first option when implementing contact force computation.

1 Introduction

Scenes encountered in computer games and computer graphics commonly contain solid objects. In order to prevent objects penetrating one another, contacts between them must be detected and resolved by applying appropriate contact impulses and forces. Here we focus on the problem of resting contacts, where the relative normal velocity at the contact point is zero, and do not address the problem of collisions, which requires a different, simpler analytical treatment. Thus, we deal with contact forces only, and not with contact impulses. A popular method for contact force computation is an algorithm introduced by Baraff (1994) that introduced important improvements relative to the methods previously available. An alternative method was developed by Faure (1996).

Here we describe a much simpler algorithm, inspired by the Gauss-Seidel iterative method for solving linear systems. Gauss-Seidel-like methods have been previously used for contact force computation in simulators using the time-stepping formulation of dynamics (Jourdan et al., 1998; Liu and Wang, 2005). The paper by Jourdan et al. (1998) was also concerned with simulating frictional contact between a rigid object and an object modeled with finite elements, instead of the more general problem of contacts between rigid bodies. In general, the presentation of Gauss-Seidel methods for contact force computation has been done in the framework of some complex systems, including features not related to contact force computation. Thus, the simplicity and utility of this method have not been made obvious in these previous papers, and the result is that the method does not currently have the popularity that it deserves. Since this method can be implemented in a few lines of code, we believe that it is extremely useful for fast prototyping of games and physics simulation systems, and it should be recognized as the first option when implementing contact force computation.

2 The problem

Our notation follows the one in (Baraff, 1994). We consider a set of rigid objects contacting at n distinct points. The relative normal velocity at each contact point is zero. We also consider here frictionless contact. We note with a_i the relative normal acceleration at the contact point i. When choosing the sign of the normal at the contact point, we adopt the convention that a positive a_i indicates that the objects are breaking contact at point i. Correspondingly, $a_i < 0$ indicates that the objects tend to interpenetrate.

We denote with f_i the magnitude of the normal contact force at contact point i. A positive f_i indicates a repulsive force. Since contact forces are repulsive, we must have $f_i \geq 0$. Another condition on the contact variables is $f_i\, a_i = 0$ at each contact point, since either there is a contact force when $a_i = 0$ and the contact remains, either $f_i = 0$ when $a_i > 0$ and the contact is broken.

We denote by **a** the vector formed by putting the a_i's together, and by **f** the vector formed from the f_i's. The acceleration vector linearly depends on the force vector:

$$\mathbf{a} = \mathbf{A}\,\mathbf{f} + \mathbf{b} \tag{1}$$

A derivation of this relation for a system of rigid bodies that are not subjected to additional constraints other than the resting contacts is presented in the Appendix. The matrix **A** reflects the masses and contact geometries of the bodies, while **b** reflects the external and inertial forces applied to the bodies. At any instant of time, **A** and **b** are known quantities, while **f** is the unknown we are interested in solving for, subject to the following conditions:

$$a_i \geq 0;\ f_i \geq 0;\ f_i\, a_i = 0 \tag{2}$$

at each contact point.

3 The algorithm

The algorithm works by iteratively analyzing each of the n contacts. For each contact, the algorithm modifies the force at the contact to comply with the constraints (Eq. 2), assuming that the other contact points are previously solved. Each iteration brings **f** closer to the solution.

The algorithm is presented in pseudocode below:

Choosing the convergence criterium depends on your application. You may define a constant number of iterations, or monitor the change of **f** over the iterations, or, in an interactive application, adapt the number of iterations to trade off accuracy to speed, as needed.

In the algorithm, q represents the value of a_i in the case that $f_i = 0$. If $q \geq 0$, then it is not necessary to have a contact force and we set $f_i = 0$. If $q < 0$, we need to have a positive contribution $A_{ii}\, f_i$ to $a_i = q + A_{ii}\, f_i$, and we compute f_i such that $a_i = 0$. We note that A_{ii} is always positive because it reflects the positive masses of the bodies.

Algorithm 1 Computing contact forces

```
Require: $n, A_{ij}, b_i (i = 1 \ldots n, j = 1 \ldots n)$
Ensure: $f_i$ according to the conditions in Eq. 2
  for $i = 1$ to $n$ do
    $f_i = 0$
  end for
  repeat
    for $i = 1$ to $n$ do
      $q = b_i$
      for $j = 1$ to $n$ do
        if $j \neq i$ then
          $q = q + A_{ij} f_j$
        end if
      end for
      if $q \geq 0$ then
        $f_i = 0$
      else
        $f_i = f_i - q/A_{ii}$
      end if
    end for
  until convergence
```

4 Discussion

The convergence and the uniqueness of the Gauss-Seidel method is not yet demonstrated. The conditions for convergence of a different Gauss-Seidel method in 2D are discussed in (Jourdan et al., 1998). It is also known that the convergence performance of a Gauss-Seidel method depends on the initial guess (Liu and Wang, 2005); we successfully set up the initial values of the forces to 0, as shown in the algorithm.

However, in our test simulations, in the framework of the Thyrix simulator (Florian, 2003) and for less than 5 simultaneous contacts, the algorithm converged to a solution comparable to the solution of Baraff's algorithm in less than 10 iterations. Moreover, the Gauss-Seidel method does not suffer of problems that Baraff's algorithm sometimes have due to singularities. Thus, we have successfully used the method presented here in conjunction with Baraff's method, as a fail-safe when Baraff's method does not converge.

In conclusion, we have presented an extremely simple algorithm for contact force computation. This algorithm allows rapid prototyping of games and physics simulators.

5 Appendix: Derivation of the contact matrix for a system of independent rigid bodies

We consider a contact i between the rigid objects A and B. We assume that these objects interact only through the resting contact i and eventually other resting contacts, or through forces acting at distance that are fixed during contact calculation, but not through other types of constraints, such as a chain of objects linked through joints. We note with $\mathbf{n}_i$ the normal at the contact point. The contact force acting on A will be $\sigma_{iA} f_i \mathbf{n}_i$ and the force acting on B will be $\sigma_{iB} f_i \mathbf{n}_i$, where the parameters σ represent the direction of the contact force along the normal, $\sigma_{iA,B} = \pm 1$ and $\sigma_{iA} = -\sigma_{iB}$. We also note by C_A the set of the contacts acting on A. The vector $\mathbf{p}_{iA}$ represents the position of contact point i relative to the center of mass of object A.

The forces acting on object A are the external forces summed into $\mathbf{F}_A^E$ and the contact forces $\sum_{j \in C_A} \sigma_{jA} f_j \mathbf{n}_j$. Thus, the acceleration $\mathbf{a}_A$ of object A is

$$\mathbf{a}_A = \frac{1}{m_A} \left(\mathbf{F}_A^E + \sum_{j \in C_A} \sigma_{jA} f_j \mathbf{n}_j \right), \tag{3}$$

where m_A is the mass of A.

The contact forces also generate a torque $\mathbf{p}_{jA} \times \sum_{j \in C_A} \sigma_{jA} f_j \mathbf{n}_j$ on A that adds to the external torques $\mathbf{M}_A^E$ acting on the object. Thus, the angular acceleration of the object is

$$\varepsilon_A = \mathbf{I}_A^{-1} \left(\mathbf{M}_A^E + \sum_{j \in C_A} \sigma_{jA} f_j \, \mathbf{p}_{jA} \times \mathbf{n}_j \right), \tag{4}$$

The acceleration of the contact point i that is connected to A is

$$\mathbf{a}_{iA} = \mathbf{a}_A + \varepsilon_A \times \mathbf{p}_{iA} + \omega_A \times (\omega_A \times \mathbf{p}_{iA}), \tag{5}$$

where ω_A is the angular velocity of A.

By introducing Eqs. 3 and 4 in 6, we get:

$$\mathbf{a}_{iA} = \frac{\mathbf{F}_A^E}{m_A} + \mathbf{I}_A^{-1} \mathbf{M}_A^E \times \mathbf{p}_{iA} + \omega_A \times (\omega_A \times \mathbf{p}_{iA}) + \sum_{j \in C_A} \sigma_{jA} f_j \left[\frac{\mathbf{n}_j}{m_A} + \mathbf{I}_A^{-1} (\mathbf{p}_{jA} \times \mathbf{n}_j) \times \mathbf{p}_{iA} \right] \tag{6}$$

The relative normal acceleration between the bodies at i is

$$\mathbf{a}_i = (\sigma_{iA} \, \mathbf{a}_{iA} + \sigma_{iB} \, \mathbf{a}_{iB}) \cdot \mathbf{n}_i \tag{7}$$

We see thus that $\mathbf{a}_{iA}$ contributes to the contact matrix $\mathbf{A}$ with a term

$$A_{ijA} = \sigma_{iA} \, \sigma_{jA} \left[\frac{\mathbf{n}_j}{m_A} + \mathbf{I}_A^{-1} (\mathbf{p}_{jA} \times \mathbf{n}_j) \times \mathbf{p}_{iA} \right] \cdot \mathbf{n}_i \tag{8}$$

corresponding to the element A_{ij} and to the contact vector $\mathbf{b}$ with a term

$$b_{iA} = \sigma_{iA} \left[\frac{\mathbf{F}_A^E}{m_A} + \mathbf{I}_A^{-1} \mathbf{M}_A^E \times \mathbf{p}_{iA} + \omega_A \times (\omega_A \times \mathbf{p}_{iA}) \right] \cdot \mathbf{n}_i \tag{9}$$

corresponding to the element b_i.

If contact j is a contact between A and B, $j \in C_A$ and $j \in C_B$ (it may be i or another one), we also have a contribution A_{ijB} to A_{ij} from B, analogous to A_{ijA}, and finally we have $A_{ij} = A_{ijA} + A_{ijB}$. If $j \in C_A$ and $j \notin C_B$, then $A_{ij} = A_{ijA}$. Analogously, if $k \notin C_A$ and $k \in C_B$, then $A_{ik} = A_{ikB}$. In practice, when implementing the computation of A_{ij} we will initially set it to zero and then add the contributions from the different objects in contact. We also have $b_i = b_{iA} + b_{iB}$.

By computing A_{ii} from the equations above, we get $A_{ii} = 1/m_A + 1/m_B$, and thus $A_{ii} > 0$.

References

Baraff, D. (1994), Fast contact force computation for nonpenetrating rigid bodies, *in* A. Glassner, ed., 'Proceedings of SIGGRAPH 1994, Computer Graphics Proceedings, Annual Conference Series', ACM Press, pp. 23–34.
http://www-2.cs.cmu.edu/~baraff/papers/sig94.pdf

Faure, F. (1996), An energy-based method for contact force computation, *in* 'Computer Graphics Forum', Vol. 15, pp. 357–366.
http://www.cg.tuwien.ac.at/~francois/Public/Work/papers/contactForce.ps.gz

Florian, R. V. (2003), Thyrix: A simulator for articulated agents capable of manipulating objects, Technical Report Coneural-03-02, Center for Cognitive and Neural Studies, Cluj, Romania.
http://www.coneural.org/reports/Coneural-03-02.pdf

Jourdan, F., Alart, P. and Jean, M. (1998), 'A Gauss-Seidel like algorithm to solve frictional contact problems', *Computer Methods in Applied Mechanics and Engineering* **155**, 31–47.

Liu, T. and Wang, M. Y. (2005), 'Computation of three-dimensional rigid-body dynamics with multiple unilateral contacts using time-stepping and Gauss-Seidel methods', *IEEE Transactions on Automation Science and Engineering* **2**(1), 19–31.
http://www2.acae.cuhk.edu.hk/~cmdl/publications/preprints/tase-paper.pdf