EFFICIENT SIMULATION OF CONTACT BETWEEN RIGID AND DEFORMABLE OBJECTS

Eder Miguel* and Miguel A. Otaduy*

*Modeling and Virtual Reality Group
Universidad Rey Juan Carlos (URJC Madrid), Tulipán s/n, E-28933, Móstoles, Spain
e-mails: eder.miguel@gmail.com, miguel.otaduy@urjc.es

Keywords: Contact Simulation, Constraints.

Abstract. Contact coupling between deformable and rigid bodies often induces situations where the surfaces of both types of bodies collide over a large area. This situation introduces serious difficulties in LCP-type contact solvers, because a large amount of contact constraints are inertially coupled through the rigid body. In this paper, we present an algorithm for efficiently simulating contact between rigid and deformable bodies. The solution to the LCP involves two steps: building the system matrix, which tends to be large and dense due to the coupling between the rigid and deformable bodies in contact, and solving the resulting problem. We efficiently handle both steps by reformulating the large LCP and separating constraint sets acting on rigid bodies alone, on deformable bodies alone, and on both rigid and deformable bodies. A modified projected Gauss-Seidel solver handles the partitioned sets of constraints in an efficient manner. We demonstrate our algorithm on several complex and contact-intensive scenarios, such as those involving cloth simulations, or contact between bone and soft-tissue.
1 INTRODUCTION

Many robotics applications involve modeling contact between rigid and deformable objects. Perhaps, the most prominent example of such contact interactions in robotics is grasping, where the soft tissue of the fingers interacts with rigid objects in the environment. Other examples include contact between bones and soft human tissue for biomechanical simulation, rigid and deformable parts in vehicles, or the interaction between the body and rigid parts around us for ergonomics analysis.

The diverse properties of rigid and deformable bodies induce complicated configurations from the point-of-view of contact handling. In particular, in this work, we address problems induced by the difference in numbers of degrees of freedom and by the sampling of the contact interface. This contact interface is typically sampled with a set of contact points (e.g., pairs of face-vertex or edge-edge primitives for triangulated surfaces). When contact takes place between a pair of rigid bodies, the rigidity of the objects itself limits the number of contact points to a few. When contact takes place between a pair of deformable bodies, on the other hand, the compliance of the material allows the surfaces to mold to each other, hence the number of contact points may be high. However, in typical deformable bodies discretized with nodal positions (e.g., in finite element discretizations) each of the contact points affects only a small number of degrees of freedom.

The contact interface between rigid and deformable bodies introduces the following complication. Due to the compliance of the deformable body, the contact interface may present a large surface area, and therefore a large number of contact points. And due to the kinematics of the rigid body, a few degrees of freedom (i.e., the position and orientation of the rigid body) may be affected by a large number of contact points. This is a rare situation between rigid bodies alone or deformable bodies alone, but happens often upon contact between rigid and deformable bodies. Contact solvers based on the formulation of a linear complementarity problem (LCP) \[8\] suffer severe complications when many contact constraints affect a small set of degrees of freedom, as we discuss in Section \[3\].

In this paper, we present an algorithm for efficiently handling contact between rigid and deformable bodies in the context of LCP-type solvers. The solution to the LCP involves two steps: (i) building the system matrix, which upon contact between rigid and deformable bodies tends to be large and dense, and (ii) solving the resulting problem. As described in Section \[5\], our algorithm efficiently handles both steps by reformulating the large LCP and separating constraint sets acting on rigid bodies alone, on deformable bodies alone, and on both rigid and deformable bodies. Contact solvers based on the formulation of a linear complementarity problem (LCP) \[8\] suffer severe complications when many contact constraints affect a small set of degrees of freedom, as we discuss in Section \[3\].

In this paper, we present an algorithm for efficiently handling contact between rigid and deformable bodies in the context of LCP-type solvers. The solution to the LCP involves two steps: (i) building the system matrix, which upon contact between rigid and deformable bodies tends to be large and dense, and (ii) solving the resulting problem. As described in Section \[5\], our algorithm efficiently handles both steps by reformulating the large LCP and separating constraint sets acting on rigid bodies alone, on deformable bodies alone, and on both rigid and deformable bodies. Then, a novel modified projected Gauss-Seidel solver handles the partitioned sets of constraints in an efficient manner. In particular, it exploits low-rank matrix computations in order to maximize matrix sparsity, and therefore reduce the total computational cost. As a result, assuming \(n\) contact constraints, the cost of formulating the LCP matrix and the cost per-iteration of the LCP solver are reduced from \(O(n^2)\) to an expected \(O(n)\).

We demonstrate our algorithm on several complex and contact-intensive scenarios, such as those involving cloth simulations, or contact between bone and soft-tissue. Thanks to the improved asymptotic complexity of the algorithm, timings for LCP construction and solution are improved by a factor of up to 14x for the examples we show. Even though we limit the demonstrations to contact between rigid and deformable bodies, our algorithm is also applicable in other situations where a large number of contact constraints affects a low number of degrees of freedom, such as contact between deformable bodies with model reduction \[13\] or embedded in low-resolution discretizations \[17\].
2 RELATED WORK

Dynamic simulation problems with contact constraints are often formulated as complementarity problems \cite{23}. We will study approaches that linearize the dynamics and constraint equations in every time step, which leads to a linear complementarity problem (LCP) \cite{8}.

LCPs for contact handling include several variations in aspects such as the formulation of the constraints, the type of dynamic objects that are solved, or the resolution algorithm. LCP-type approaches have been applied mostly to rigid body contact (See \cite{2, 3, 4, 20, 11} for some examples), but also to deformable body contact \cite{6, 10, 19, 13, 18}. They can also be classified based on the nature of the constraints, such as acceleration level \cite{14, 3, 4}, or velocity-level constraints \cite{22, 1}. It is also possible to formulate position-level constraints based on Signorini’s condition \cite{10, 18}, but this approach is non-linear, and it requires time-discretization and linearization in order to finally express a velocity-level LCP.

In terms of the solution method, Lemke’s pivoting method \cite{9} has been the most popular one for a long time, and its convergence has recently been improved for articulated bodies \cite{24}. Other approaches include relaxation solvers such as projected Gauss-Seidel (PGS) \cite{8}, which is also connected to iterative impulse-based solvers \cite{12}.

The constrained dynamics problem constitutes a mixed LCP (MLCP) \cite{8}, which is typically converted first into an LCP and then solved with one of the methods mentioned above. Converting the MLCP into an LCP may be a computationally expensive operation, requiring the solution to multiple instances of forward dynamics for the objects involved in contact \cite{5}. One way to avoid some of this cost is to use a nested relaxation solver that produces sparse LCPs \cite{18}. Another way is to introduce some compliance in the contact constraints, which allows the application of one unique relaxation solver to the complete system. This approach has recently been optimized in order to include a step of subspace minimization \cite{15}.

As discussed in the introduction, our work addresses the high computational cost induced by contact configurations involving rigid and deformable bodies together. But our approach can directly be extended to handle other deformable bodies where few degrees of freedom are affected by many contact constraints, which could be the case for reduced deformation models \cite{13} or for deformable bodies embedded in low-resolution discretizations \cite{17}. Others have addressed other problems appearing in the contact coupling between rigid and deformable bodies, such as the combination of solvers that are best suited for each of the object types separately \cite{21}.

3 PROBLEM STATEMENT

The discretized constrained dynamics of a system combining rigid and deformable bodies can be written as the following MLCP:

\[
A v = J^T \lambda + b,
\]

\[
0 \leq \lambda \perp Jv \geq c.
\]

Here, \(A\) represents the system’s dynamics’ matrix, including mass, damping, and stiffness terms, depending on the numerical integrator of choice. In our examples, we have used linear co-rotational finite element models \cite{16} and mass-spring cloth \cite{7} for the simulation of deformable bodies. All the examples were discretized in time using backward Euler integration with a linearization of forces (i.e., equivalent to doing one step of a Newton solver for the resulting non-linear system). We omit the description of friction here for readability, but our algorithm is applicable directly to an LCP with friction.
The MLCP can be converted into an LCP through Schur complement computation of the matrix $A$, also named constraint anticipation [5]. We follow an iterative constraint anticipation approach [18], which solves the MLCP using two nested relaxation solvers. Each iteration of the outer relaxation solver formulates an LCP as follows:

$$0 \leq \lambda(i) \perp B\lambda(i) \geq d(i),$$

with $B = JD_A^{-1}J^T$, and $d(i) = c - JD_A^{-1}((L_A + U_A)v(i - 1) + b)$.

$A = D_A - L_A - U_A$ splits matrix $A$ into its (block-)diagonal and the lower and upper triangular matrices. In the rest of the text, we will drop the subindex $i$, and we will always refer to one outer-loop iteration of iterative constraint anticipation.

Fig. 1 shows three different contact situations with rigid and deformable bodies, as well as the sparsity patterns of matrix $B$ for all three situations. As discussed in the introduction, mixing rigid and deformable bodies produces an LCP that is both large and dense. LCPs with the same sparsity pattern also appear when the contact forces over the deformable bodies are integrated explicitly.

Large and dense LCPs imply two negative consequences in terms of computational cost. First, the construction of matrix $B$ has a cost $O(n^2)$, with $n$ the number of contact constraints. Since each contact on a rigid body is inertially coupled to all other contacts on the same body, the formulation of the matrix requires the anticipation of the effect of all pair-wise constraints [5]. Second, the solution to the LCP using relaxation algorithms requires visiting, for each row of the matrix, all columns. Then, again, the cost for iterating over the complete matrix becomes $O(n^2)$.

For the sake of completeness, the right-hand-side in Eq. (3) can be formulated with a cost
Eder Miguel and Miguel A. Otaduy

Figure 2: Schematic decomposition of the bodies and constraints in a contact scenario. Deformable bodies (1) and rigid bodies (2) share a set of constraints (c), while other constraints (a and b) act only on one type of object.

$O(n)$, since it implies sparse matrix - vector multiplications.

4 CONSTRAINT SETS

Even though the LCP matrix $B$ turns out dense and large in situations with contact between rigid and deformable bodies, we will show now that it can be split into the sum of a sparse matrix and a dense but low-rank matrix. In the next section, we will take advantage of this separation of the dense low-rank part in order to present an algorithm to formulate and solve the LCP that is asymptotically (and practically) far more efficient than the straightforward approach.

In order to split the matrix $B$, we separate the bodies in the scene into two sets:

1. Deformable bodies.
2. Rigid bodies.

Similarly, we separate the contact constraints into three sets:

a. Constraints acting only on deformable bodies.

b. Constraints acting only on rigid bodies.

c. Constraints acting on both deformable and rigid bodies.

Fig. 2 shows the connections between body sets and constraint sets. In a multibody setting, this set decomposition should be carried out for each island of objects.

The previous constraint and object separation allows us to express the matrices and vectors of the MLCP in Eq. (1) as:

$$
v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},
$$

$$
\lambda = \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix}, \quad J = \begin{bmatrix} J_{a1} & 0 \\ 0 & J_{b2} \\ J_{c1} & J_{c2} \end{bmatrix}.
$$

Given these definitions, the LCP matrix can be expressed as $B = B_s + B_d$, i.e., the sum of a sparse matrix $B_s$ and a dense matrix $B_d$. These matrices are defined as:

$$
B_s = \begin{bmatrix} J_{a1}D_{A_{11}}^{-1}J_{a1}^T & 0 & J_{a1}D_{A_{11}}^{-1}J_{c1}^T \\ 0 & 0 & 0 \\ J_{c1}D_{A_{c1}}^{-1}J_{a1}^T & 0 & J_{c1}D_{A_{c1}}^{-1}J_{c1}^T \end{bmatrix}, \quad \text{and}
$$
The blocks in $B_d$ are built out of the large but (block-)diagonal matrix $D_{A_11}$, and sparse constraint Jacobians $J_{a1}$ and $J_{c1}$. In $B_d$, on the other hand, the block $J_{c2}D_{A_{22}}^{-1}J_{c2}^T$ is large and dense, because the constraint Jacobian $J_{c2}$ stores values for many contacts acting on just a few degrees of freedom (the degrees of freedom of the rigid bodies). However, this same matrix block $J_{c2}D_{A_{22}}^{-1}J_{c2}^T$ has a low rank, determined by the number of degrees of freedom of the rigid bodies, independently of the large number of contacts.

5 ALGORITHM

We present now our algorithm for resolving the LCP of contact scenes combining rigid and deformable objects. This algorithm takes advantage of the low-rank nature of the part of the LCP matrix due to the rigid bodies, $B_d$. In the context of a projected-Gauss-Seidel (PGS) solver, we iteratively refine the collision response on the rigid bodies, and thus avoid traversing a dense matrix in each PGS iteration.

5.1 Rigid Response Refinement

During the resolution of the LCP, we maintain at all times the response velocity of the rigid bodies produced by the current values of the Lagrange multipliers $\lambda$. During the $j^{th}$ iteration of the PGS solver, after traversing the $k^{th}$ row, let us define as $\lambda(j, k)$ the temporary vector of Lagrange multipliers that contains values from the previous and the current iterations:

$$\lambda(j, k) = [\lambda_1(j), \ldots, \lambda_k(j), \lambda_{k+1}(j-1), \ldots, \lambda_n(j-1)].$$

Then, we define the temporary response of the rigid bodies as $v_2(j, k) = D_{A_{22}}^{-1}J_{c2}^T\lambda(j, k)$. We can precompute $P = D_{A_{22}}^{-1}J_{c2}^T$, and then $v_2(j, k) = P\lambda(j, k)$.

When a new row is processed by PGS, the rigid body response can simply be updated as $v_2(j, k) = v_2(j, k-1) + P_k(\lambda_k(j) - \lambda_k(j-1))$, where $P_k$ is the $k^{th}$ column of $P$. If the scene contains many rigid bodies, $P_k$ is sparse and has non-zero terms only for the rigid bodies affected by the $k^{th}$ constraint.

Rigid response refinement allows for inexpensive low-rank updates in the context of the PGS iterations, as we will show next.

5.2 Modified PGS Solver

Given the decomposition of $B_s$ and $B_d$ into diagonal and lower and upper triangular matrices, i.e., $B_s = D_s - L_s - U_s$ and $B_d = D_d - L_d - U_d$, an iteration of PGS can be expressed as:

$$(D_s - L_s + D_d - L_d) \lambda(j) \geq d + (U_s + U_d) \lambda(j-1).$$

(5)

For the $k^{th}$ row, this can be written as:

$$(D_{s,k} - L_{s,k} + D_{d,k} - L_{d,k}) \lambda(j) \geq d_k + (U_{s,k} + U_{d,k}) \lambda(j-1).$$

(6)

Based on the definition of the temporary vector of Lagrange multipliers, $\lambda$, we can rewrite this expression:

$$d_k + (D_{d,k} + U_{s,k}) \lambda(j-1) - B_{d,k} \lambda(j, k-1).$$

(7)
In this expression, the matrix on the left-hand side is sparse, while the dense matrix $B_d$ on the right-hand side is multiplying a vector of known Lagrange multipliers. Hence, we can exploit the low-rank definition of $B_d$ in order to compute the right-hand side of the inequality. In particular, we apply the concept of rigid response refinement described above:

$$J_{2,k}v_2(j, k - 1) = B_{d,k} \overline{\lambda}(j, k - 1)$$

so that then:

$$(D_{s,k} + D_{d,k}) \lambda(j) \geq d_k + L_{s,k} \lambda(j) + (D_{d,k} + U_{s,k}) \lambda(j - 1) - J_{2,k}v_2(j, k - 1).$$

This final expression lends itself to an efficient implementation. The modified PGS solver requires $O(1)$ computational cost per row, hence a total $O(n)$ per iteration. Moreover, the matrix $B_d$ no longer needs to be computed, hence the cost for building the LCP matrix is also reduced to $O(n)$. This asymptotic cost analysis is verified with the data from our experiments.

The algorithm for each iteration $j$ of the LCP solver can be summarized as follows:

For each constraint $k$:
- Compute the right-hand side $r_k$ in Eq. (9).
- Compute $\lambda_k^* = r_k / (D_{s,kk} + D_{d,kk})$.
- Project $\lambda_k(j) = \max(\lambda_k^*, 0)$.
- Update $v_2(j, k) = v_2(j, k - 1) + P_k(\lambda_k(j) - \lambda_k(j - 1))$.

The handling of friction using a pyramid approximation of Coulomb’s model can be done similarly in the context of the PGS solver, with a block-matrix implementation. Then, the iteration for each constraint requires the computation of a Lagrange multiplier $\lambda_k$ in the normal direction, and two additional values for the tangential directions. Some examples of such solvers with friction are given in [10, 18]. Another possibility is to use staggered projections for normal and friction response [13].

## 6 RESULTS

We have evaluated the performance of the algorithm in practice on two examples that combine rigid and deformable bodies. One example consists of two rigid vertebrae squeezing a deformable intervertebral disk (Fig. 3). The other example shows a piece of cloth that knocks down a rigid bunny model (Fig. 5). Both examples were tested on an Intel Core 2 Duo 2.3GHz-processor laptop with 3GB of memory.

Fig. 4 compares the timings between our algorithm and the standard approach for building the LCP system and for solving one iteration of the LCP, in the vertebrae example. The timings are plotted against the number of contact constraints in the scenario, which allows us to validate our asymptotic cost analysis. Both the construction and solution costs show a clear linear trend using our algorithm. With the standard algorithm, on the other hand, they show a quadratic trend for almost the first half of the contacts. Then, this trend becomes linear. The reason is that the quadratic trend is only present until the number of contacts reaches the maximum number of contacts suffered by a single rigid body. For higher numbers of contacts, these are split among rigid bodies, hence the cost is dominated by the linear trend. Nevertheless, the factor of this linear trend is higher with the standard algorithm than with ours.
Figure 3: An intervertebral disk being squeezed between two vertebrae. The vertebrae are simulated as rigid bodies, while the disk is simulated using FEM, with a 6021-tetrahedra mesh. The surface meshes contain 5888 triangles for each vertebra, and 8192 for the disk.

Figure 4: Evaluation of the LCP construction and solution cost Vs. the number of contacts in the vertebrae scenario.

Fig. 6 compares as well the construction and solution cost of the LCP, for the bunny example. In this case, the timings are plotted against the simulation steps. Our algorithm outperforms the standard approach for building and solving the LCP by a factor of up to 14x. The main practical consequence of this speed-up is that the examples shown here run at close to interactive rates.

7 DISCUSSION AND FUTURE WORK

The results of our tests demonstrate that our algorithm for simulating contact between rigid and deformable bodies effectively enjoys a linear cost for constructing the LCP system and for solving one iteration of PGS. Compared to the quadratic cost of the standard algorithm, this difference provides important speed-ups that allow for interactive simulations under moderate scene complexity.

The key contribution that leads to the reduced cost is a decomposition of the system matrix into a sparse submatrix and a dense but low-rank submatrix. Taking advantage of the low-rank nature of the second submatrix, we have designed a modified version of the projected- Gauss-Seidel solver that avoids formulating large and dense matrices. As shown in the paper, our algorithm is simple to formulate, and brings little implementation overhead in comparison with standard PGS solvers for LCPs.

Despite the speed-ups provided by our algorithm, there are still some limitations and open issues. As already mentioned, our algorithm relies on a Gauss-Seidel relaxation solver. Such
relaxation solvers may suffer from slow convergence, and we did indeed experience slow convergence in some contact situations when large surfaces collided in violent impacts, or when the deformable bodies suffered a large strain during contact. We plan to evaluate approaches that address the slow convergence of relaxation solvers for LCPs [15], and investigate the integration of our algorithm for efficiently handling contact between rigid and deformable bodies.

Currently, we are applying our approach only to rigid bodies in contact with densely discretized deformable bodies (FE meshes or mass-spring models), but the key concepts presented here can be applied in general to any dynamic model where a large number of contacts affect a small number of degrees of freedom.

Acknowledgments

Eder Miguel was supported by a fellowship from La Caixa. We would also like to thank Javi S. Zurdo for help with demos, Alvaro Pérez for help with renders, and the rest of the GMRV group at URJC Madrid.

REFERENCES


