Analysis and Evaluation for Constraint Enforcement System*

Min Hong¹ · Doo-Soon Park¹ · Yoo-Joo Choi²†

제한 시스템의 분석 및 평가
홍 민 · 박두순 · 최유주

ABSTRACT

Stable and effective constraint enforcement system is one of the crucial components for physically-based dynamic simulations. This paper presents analysis and evaluation for traditional constraint enforcement systems (Lagrange Multiplier method, Baumgarte stabilization method, Post-stabilization method, Implicit constraint enforcement method, Fast projection method) to provide a guideline to users who need to integrate a suitable constraint enforcement system into their dynamic simulations. The mathematical formulations for traditional constraint enforcement systems are presented in this paper. This paper describes a summary of evaluation which consists of constraint error comparison, computational cost, and dynamic behavior analysis to verify the efficiency of each traditional constraint enforcement system.

Key words: Physically-based simulation, Constraint enforcement, Dynamic simulation, Constraint error

요 약

물리적 기반의 다이내믹 시뮬레이션에 있어서 안정적이고 효율적인 제한 시스템은 매우 중요한 요소 중 하나이다. 본 논문은 기존에 널리 사용되고 있는 제한 시스템들 (Lagrange Multiplier method, Baumgarte stabilization method, Post-stabilization Method, Implicit constraint enforcement method, Fast projection method)에 대한 분석과 평가를 통해 제한 시스템을 사용하고자 하는 사용자들에게 적합한 선택을 할 수 있는 지침을 제공하고자 한다. 본 논문은 기존의 제한 방법들을 위한 수학적 수식들이 설명되어 있고, 제한 오차 비교, 계산 비용, 동적 움직임 분석 등을 통해 기존 제한 시스템들 각각에 대한 평가를 제공한다.

주요어: 물리적 기반 시뮬레이션, 제한 적용, 다이내믹 시뮬레이션, 제한 오차

1. Introduction

Many dynamic simulation systems (Witkin and Baraff, 2003; Goldenthal et al., 2007; Barzel and Barr, 1988; Platt and Barr, 1988) have been developed to estimate the results of moving objects using powerful computers. Due to the increasing demands for visual realism and needs for physically accurate motion in computer animation and simulation, physically-based dynamic simulations (Choi and Ko, 2002; Hong and Kim 2005, Song et al., 2005) have been widely applied to represent the realistic dynamic motion of rigid or deformable objects. The dynamic simulation of unconstrained deformable body made big strides over the past decade, so it is becoming very hard to distinguish computer simulated objects with real objects in TV commercials or movies. However, to achieve the physically realistic behavior of objects, the robust and fast control of dynamic simulation becomes an even more critical issue. Geometric constraint enforcement is an imperative technique to control the
behavior of objects in wide range of applications from a simple rigid link models to the complicated motion of manifold objects.

Maintaining hard constraints under a tight error bound is an important issue, since a small but accumulating constraint drift could result in the ruin of dynamic simulations. In stead of constraint enforcement, very stiff springs or penalty forces(Witkin et al., 1987; Terzopoulos et al., 1987; Hirota et al., 2001) are applied to maintain given constraints. However, these cause the stiff differential equations that may occur the unstability of simulations. Therefore, special integration methods or reduced integration time step size is required for practical simulations(Hauth et al., 2003). Constraint-based control using Lagrange multipliers is well studied and broadly used for rigid body dynamic systems(Baraff, 1996). The Lagrange multiplier method, associated with ordinary differential equations, has a fundamental limitation of an accumulated constraint error due to the numerical discretization. Therefore, the accumulated constraint error becomes apparent and it causes severe instability problems in constrained dynamic simulations. Several breakthroughs(Baraff, 1996; Baumgarte, 1972; Cline and Pai, 2003) have been proposed in stabilizing the constraint or minimizing error. However, these methods require some additional processes such as parameter tweaking(Baumgarte, 1972) or extra computational burdens(Cline and Pai, 2003) to maintain the stability of simulations and to reduce the constraint errors.

Hong et al.(2005) presented implicit constraint enforcement method to control the rigid links and the excessive elongation of cloth patch. Unlike simple distance reduction approach(Provot, 1995) which applied dynamic inverse constraints, the implicit constraint enforcement technique provides relatively good stability over large time steps and preserves the natural dynamic behavior of simulations without any ad-hoc stabilization terms. Recently fast projection method(Goldenthal et al., 2007) was introduced to efficiently simulate the inextensible cloth. This method acts as a velocity filter and successfully provides inextenibility of cloth with good convergence behavior. This paper minutely describes possible traditional methods which enforce constraints in dynamic simulations and provides the comparison from the point of view of constraint error, of computational cost, and of dynamic behavior.

2. Traditional Constraint Enforcement Systems

In this section, the formulation and distinctive features of traditional methods will be briefly explained. Once the geometric constraints are defined with a set of equations, the appropriate constraint forces should be computed to keep up given constraints. One of the most popular methods for integrating geometric constraints into a dynamic system is using Lagrange multipliers and constraint forces. The constraint-based formulation using Lagrange multipliers results in a mixed system of ordinary differential equations and algebraic expressions.

For example, suppose that the simulation requires the distance restriction between two nodes. When \(x_0, y_0, z_0\) is position of one node and \(x_1, y_1, z_1\) is position of the other node, the distance \(r\) between two nodes should be continuously maintained during the whole simulation. This constraint can be defined by following Eq. (1).

\[
\Phi(q,t) = (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2 - r^2 = 0
\]  

(1)

We can write this constraint equations using \(3n\) of generalized coordinates, \(q\), where \(n\) is the number of discrete masses and the generalized coordinates are simply the Cartesian coordinates of the discrete masses.

\[
q = [x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_{n-1}, y_{n-1}, z_{n-1}]^T
\]  

(2)

2.1 Lagrange Multiplier Method

Let \(\Phi(q,t)\) be the constraint vector with \(m\) components and each component represents an algebraic constraint. The constraint vector is represented by following Eq. (3)

\[
\Phi(q,t) = [\Phi^0(q,t), \Phi^1(q,t), \ldots, \Phi^{m-1}(q,t)]^T
\]  

(3)
here $\Phi^i$ is the individual scalar algebraic constraint equations. We can write this system of equations as

$$M\ddot{q} + \Phi^T_q(q,t)\lambda = F^A$$

$$\Phi(q,t) = 0$$

where $F^A$ is applied, gravitational, and spring forces acting on the discrete masses, $M$ is a $3n \times 3n$ diagonal matrix containing discrete nodal masses, $\lambda$ is a $m \times 1$ vector containing the Lagrange multipliers, $\Phi(q,t)$ is for constraint force.

Suppose that the simulation is initially starting with legal state in position and velocity, $\Phi(q,t) = \dot{\Phi}(q,t) = 0$, and then $\ddot{\Phi}(q,t)$ has to be minimized close to 0 with additional constraint force $F^C$. From given constraint functions $\Phi(q,t)$, $\dot{\Phi}(q,t)$ can be computed by taking derivatives. The derivatives of $\Phi(q,t)$ is $\dot{\Phi}(q,t) = \frac{\partial \Phi}{\partial q}q + \frac{\partial \Phi}{\partial \dot{q}}\dot{q} = 0$ because $\Phi$ is not an explicit function of $t$. $\Phi(q,t)$ is Jacobian matrix of $\Phi(q,t)$ which includes connectivity information of constraints. The derivative of the Jacobian matrix gives $\ddot{\Phi}(q,t) = \frac{\partial^2 \Phi}{\partial q \partial \dot{q}}\dot{q} + \frac{\partial^2 \Phi}{\partial q \partial q}q = 0$ and we can denote $\frac{\partial^2 \Phi}{\partial q \partial \dot{q}}$ by $\dot{\Phi}_q(q,t)$ as well. Thus second derivative of constraint function can be

$$\ddot{\Phi}(q,t) = \dot{\Phi}_q(q,t)\dot{q} + \Phi_q(q,t)q = 0$$

(5)

From the above Eq. (7) all other values are known except $\lambda$, so this Lagrange multipliers is computed by solving the linear system and then multiplied with Jacobian transpose matrix to obtain, $F^C = \Phi^T_q(q,t)\lambda$, constraint forces.

### 2.2 Baumgarte Stabilization Method

Once the force computation is completed, we should estimate the next states of velocity and position for each node using numerical integration. The ODE (Ordinary Differential Equation) such as Euler, Mid-point method, and 4th order Runge-Kutta method is widely used as a numerical integrator for dynamic simulations. However, there are mainly two sources of numerical error in the constrained-based dynamic simulations: computational error in Lagrange multipliers and in numerical integration. Some numerical error comes from when the Lagrange multipliers is computing, but the most unavoidable numerical error comes from when the numerical integration is computing. Therefore, constraint stabilization methods have been applied to remove this numerical error.

Many stabilization methods have been proposed for the constrained dynamic simulation systems and one of the most popular stabilization methods is Baumgarte stabilization method(Baumgarte, 1972). Baumgarte proposed to add $\alpha\Phi(q,t) + \beta\dot{\Phi}(q,t)$ terms into the constraint force computation to alleviate the numerical error. Here, $\alpha$ is spring constant and $\beta$ is damping constant. These terms are added in the constraint computation equation which is in Eq. (7)(Witkin and Baraff, 2003)

$$\Phi_q(q,t)M^{-1}\Phi^T_q(q,t)\lambda = \dot{\Phi}_q(q,t)\dot{q} + \Phi_q(q,t)M^{-1}(F^A - F^C) + \alpha\Phi(q,t) + \beta\dot{\Phi}(q,t)$$

(8)

### 2.3 Post–Stabilization Method

Cline and Pai(2003) introduce a post-stabilization approach for rigid body simulation. This approach compensates integration error at each time step to rectify the constraint error. Instead of previous Baumgarte stabilization method that adds the second order stabilization terms to reduce the constraint error, post-stabilization method modifies the position of nodes at each integration
time step to satisfy constraints. Due to numerical error, the constraint equations include constraint error, \( \Phi(q, t) \neq 0 \). Post-stabilization method attempts to find the new displacement \( q + dq \) that satisfies the constraints and enforces the constraint equations to be zero, \( \Phi(q, t) = 0 \). We can approximate this condition by following equation:

\[
\Phi(q + dq, t) \approx \Phi(q, t) + \Phi_t(q, t) dq = 0
\]  

(9)

The unknown \( dq \) can be obtained by solving the linear system in Eq. (10).

\[
\Phi_t(q, t) dq = -\Phi(q, t)
\]  

(10)

Thus post-stabilization method should additionally solve this linear system at each integration time step to reduce the numerical error.

### 2.4 Implicit Constraint Method

Motivated by implicit integration method (Baraff and Witkin, 1998), Hong et al. (2005) propose implicit method for constraint satisfaction. The equations of motion along with the kinematics relationship between \( q \) and \( \dot{q} \) can be discretized to first order integration as

\[
\dot{q}(t + \Delta t) = \dot{q}(t) - \Delta t M^{-1} \Phi_q^T(q, t) \lambda + \Delta t M^{-1} F \quad (11)
\]

\[
q(t + \Delta t) = q(t) + \Delta t \dot{q}(t + \Delta t) \quad (12)
\]

The constraint equations can be written at new time thus they are treated implicitly

\[
\Phi(q(t + \Delta t), t + \Delta t) = 0 \quad (13)
\]

Eq. (13) can be approximated using a truncated first order Taylor series

\[
\Phi(q(t) + \Phi_t(q, t)(q(t + \Delta t) - q(t)) + \Phi_q(q, t) \Delta t = 0 \quad (14)
\]

Note that the subscript \( t \) indicates partial differentiation with respect to \( t \). Substituting \( \dot{q}(t + \Delta t) \) from Eq. (11) into Eq. (12) we obtain

\[
q(t + \Delta t) = q(t) + \Delta t \dot{q}(t) + \Delta t \left[ \Delta t M^{-1} F - \Delta t M^{-1} \Phi_q(q, t) \lambda \right] \quad (15)
\]

Substitution of this result into Eq. (14) eliminates \( q(t + \Delta t) \) and it returns the following linear system with unknown \( \lambda \).

\[
\Phi_q(q, t) M^{-1} \Phi_q^T(q, t) \lambda = \frac{1}{\Delta t^2} \Phi(q, t) + \frac{1}{\Delta t} \Phi_t(q, t) + \Phi_q(q, t) \left[ \frac{1}{\Delta t} q(t) + M^{-1} F \right] \quad (16)
\]

Note that the coefficient matrix in left side for this implicit method is the same as the coefficient matrix for Baumgarte stabilization method in Eq. (7). Thus the computational complexity of both methods is asymptotically same. This system is solved for Lagrange multipliers then Eq. (11) and (12) are used to update the generalized velocities and coordinates. The matrix \( \Phi_q(q, t) M^{-1} \Phi_q^T(q, t) \) is usually symmetric and positive definite. Thus this system can be solved by a preconditioned conjugated gradient method.

### 2.5 Fast Projection Method

Recently Goldenthal et al. (2007) propose the fast projection method with Constrained Lagrangian Mechanics to efficiently enforce inextensibility for cloth simulation. This method works as a velocity filter that maintains the given constraints. It starts with the unconstrained position and tries to find a close position on the constraint manifold. For fast projection, the following linear system should be solved at each integration time step.

\[
\Delta q = \Phi_q(q, t) M^{-1} \Phi_q^T(q, t) \lambda \quad (17)
\]

The increment \( \Delta q = \Phi_q^T(q, t) \lambda \) can be calculated using \( \lambda \) which is computed by Eq. (17). Using \( \Delta q \), we can estimate the position and can calculate the constraint-enforcing velocity to update the new velocity and position. This fast projection method is successfully applied to the cloth simulation to obtain very low strain along the warp and weft direction.

### 3. Analysis and Evaluation of Traditional methods

#### 3.1 Constraint Error Comparison

Fig. 1 shows an experimental one-link simulation fall-
ing down under gravity. Initially the link is in horizontal position such as Fig. 1(a) and one end-node (red color) is fixed and then freely falling down according to the gravity. There is no air drag or any external forces in the simulation.

In the one-link simulation, two nodes are connected by link which is represented by a distance constraint. Initially, the length of link is set to 10, thus it should be maintained during the whole simulation. However, the dynamic simulation contains some sources of numerical drift, the constraint is not met to 0. To fairly compare the constraint error, the constraint error of traditional methods is recorded at each integration time step under the same simulation conditions. Fig. 2 shows the constraint error graph using the simulation of the falling one-link simulation shown in Fig. 1. We compare the constraint error of Baumgarte method, post stabilization method, implicit constraint method, and fast projection method at each integration time step. In this graph, Post-stabilization method provides the best performance in constraint error point of view.

3.2 Dynamic Behavior Analysis
To equitably compare the dynamic behavior of existing methods during simulation, we also analyze the motion of traditional constraint enforcement systems with a one-link shown in Fig. 1. Again, there is no air drag or any external forces in the simulation thus the end of one-link node (green color) should be reached to the original position (yellow horizontal line) due to the energy conservation. Fig. 3 shows the trajectory of y-coordinate for the end of falling one-link in Fig. 1. The position of the end of one-link (green color) in each integration time steps is stored with traditional methods for behavior comparison. Unlike most methods which fairly well reach to yellow horizontal line, the trajectory of post-stabilization method deviates from due to energy loss at 0.01 integration time step. In addition, when we increase the integration time step to 0.05, the trajectory of Baumgarte method with obviously becomes unstable and trajectory severely oscillates due to numerical error.

Fig. 4 shows the upper limit position of traditional methods. The trajectory of implicit constraint method exactly overlaps with fast projection method and they are converged to 0. Therefore, it shows that there are
4. Evaluation of Traditional Methods

Generally Baumgarte method is widely used due to its simplicity, but the solution with Baumgarte stabilization methods is sensitively bounded by the integration time step. In most case, the integration time step should be small to guarantee the stability of dynamic simulation. In addition, another the difficulty is a selection of the parameter to enforce the constraint in a lower error tolerance. This selection of parameters is ad-hoc and problem dependent and it is not easy to choose optimal parameters. The difficulty of selection of proper parameter values for Baumgarte stabilization method and improved constraint stabilization techniques are explained in Ascher et al. (1994).

Although post-stabilization method preserves a tight error bound in the constraint error point of view, it can not guarantee the correct physical behavior since the constraint error reduction is independently performed from the conforming dynamic motion of the objects. Therefore, unnatural behavior of object is showed in dynamic behavior analysis section. Besides, this method requires an additional expensive linear system solution.

Implicit constraint method does not require to select any ad-hoc parameters and there is no additional computation cost with good accuracy. Also there is no energy variation which causes abnormal dynamic motion of object.

Table 1 shows the upper limit position of the end of one-link for traditional methods. The result shows that implicit constraint method most closely reaches to yellow horizontal line and it's absolute value of sum and average of upper limit position is also mostly close to 0.

Table 2 summarizes the performance comparison between traditional methods. Accuracy column shows the ranking for constraint error and post-stabilization method performs in the lowest error bound. In summary, post-stabilization method returns the best performance, implicit constraint method provides better performance.
Table 2. Performance comparison for traditional methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (Rank)</th>
<th>Dynamic Behavior</th>
<th>Computational Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baumgarte</td>
<td>4</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>Post-stabilization</td>
<td>1</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>Implicit constraint</td>
<td>2</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Fast projection</td>
<td>3</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>

than fast projection method, and Baumgarte method provides the worst performance in constraint error.

For dynamic behavior, implicit constraint method and fast projection method bring correct motion of object with good energy conservation. However, the energy variation which causes the abnormal movement of object occurs in other methods.

On the computational cost point of view, all traditional methods basically require to solve the linear system once to estimate next position and velocity of object. However, post-stabilization method requires to solve the linear system twice, it can cause slow refresh rate for complex and detailed objects.

In conclusion, although the formation is somewhat troublesome to implement, this paper highly recommend to use implicit constraint method to seamlessly integrate the constraint system into any physically-based dynamic systems.

5. Conclusion

Most physically-based dynamic simulation systems require robust constraint enforcement system to control the simulations. Although traditional constraint methods can be seamlessly integrated into existing dynamic simulation systems, they have their own characteristics and formulations to enforcing given constraints. This paper provides the comparison and evaluation of constraint error, dynamic behavior, and computational cost for traditional methods using the simple example. Therefore, the user who wants to use constraint enforcement system can be readily understood the features of each method and can be helpful to select it.

References


