Using Splitting Methods to solve Contact Force Problems

Kenny Erleben[1]

[1]eScience Center
University of Copenhagen

October 1, 2009

The Counteracting Goals of Animation

- Fast Computations ($\approx$ Interactivity)
- Robustness (Works no matter what)
- Accuracy (Physical correct or plausible)
- Problem size (The whole world)

The Life of a Free Particle

The Lagrangian

$$L = \frac{1}{2}mv^2 - V(r) \tag{1}$$

The Action

$$S = \int_a^b L dt \tag{2}$$

By principle of least action we have the Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial L}{\partial v} - \frac{\partial L}{\partial r} = 0 \tag{3}$$

Everything is Newtonian

From the Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial L}{\partial v} - \frac{\partial L}{\partial r} = 0 \tag{4}$$

We have equation of motions (Newtons 2. Law)

$$m\frac{dv}{dt} = -\nabla V(r) \tag{5a}$$

$$\frac{dr}{dt} = v \tag{5b}$$

This is a coupled first order ordinary differential equation

Nobody has complete Freedom

- Non-penetration
- Friction
- Impacts

These are non-linear and non-smooth effects.

To gain speed over accuracy people simplify by smoothing and linearization... they loose touch with the real-world!

Example of Non-Penetration

Non-penetration constraint, the configuratin space is defined by

$$c(r) \geq 0 \tag{6}$$

Adding the constraint to the principle of least action results in the KKT condition

$$
\begin{aligned}
m\frac{dv}{dt} + \nabla V(r) - \nabla c(r)^T \lambda &= 0 &\text{(7a)}\\
\frac{dr}{dt} &= v &\text{(7b)}\\
c(r) &\geq 0 &\text{(7c)}\\
\lambda &\geq 0 &\text{(7d)}\\
c(r)^T \lambda &= 0 &\text{(7e)}
\end{aligned}
$$

Physical Interpretation

$$m\frac{dv}{dt} + \nabla V(r) - \nabla c(r)^T \lambda = 0 \tag{8a}$$

$$\frac{dr}{dt} = v \tag{8b}$$

$$c(r) \geq 0 \tag{8c}$$

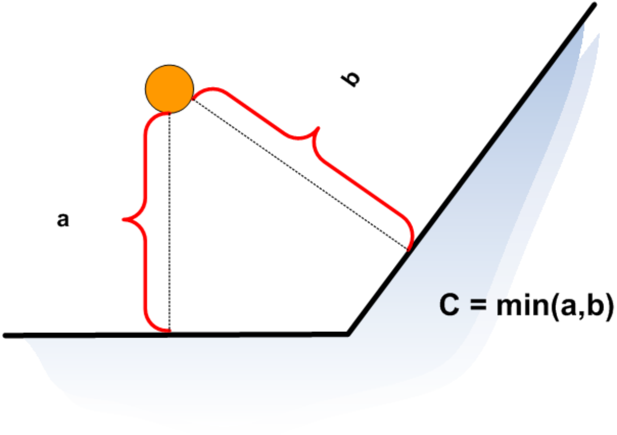
$$\lambda \geq 0 \tag{8d}$$

$$c(r)^T \lambda = 0 \tag{8e}$$

- λ is normal force
- $\nabla c(r)$ is contact normal
- $\lambda \geq 0$ means that normal forces are non-attractive
- $c(r)^T \lambda = 0$ means we only can have a normal force at a contact that is where $c(r) = 0$

The Constraint Function

In general $c(r)$ is non-smooth so $\nabla c(r)$ may not make sense



Ill-posed Problem

The mathematical model is ill-posed... we fix it by replacing positional constraint with velocity level constraints

$$\text{if} \quad c(r) = 0 \quad \text{then} \quad \nabla c(r)^T v \geq 0 \tag{9}$$

So

$$m \frac{dv}{dt} + \nabla V(r) - \nabla c(r)^T \lambda = 0 \tag{10a}$$
$$\frac{dr}{dt} = v \tag{10b}$$
$$\nabla c(r)^T v \geq 0 \tag{10c}$$
$$\lambda \geq 0 \tag{10d}$$
$$\lambda^T (\nabla c(r)^T v) = 0 \tag{10e}$$

Time Discretization

Use first order finite differences and make constraints hold at v^{n+1} and r^n then

$$m\frac{v^{n+1} - v^n}{\Delta t} + \nabla V(r) - \nabla c(r)^T \lambda = 0 \tag{11a}$$

$$\frac{r^{n+1} - r^n}{\Delta t} = v^{n+1} \tag{11b}$$

$$\nabla c(r^n)^T v^{n+1} \geq 0 \tag{11c}$$

$$\lambda \geq 0 \tag{11d}$$

$$\lambda^T (\nabla c(r^n)^T v^{n+1}) = 0 \tag{11e}$$

Clever Manipulation

From

$$m\frac{v^{n+1} - v^n}{\Delta t} + \nabla V(r) - \nabla c(r)^T \lambda = 0 \tag{12}$$

we get

$$v^{n+1} = h(\lambda) \tag{13}$$

By substitution we have

$$f(\lambda) = \nabla c(r^n)^T v^{n+1} = \nabla c(r^n)^T h(\lambda) \tag{14}$$

so the complementarity constraints are now

$$f(\lambda) \geq 0 \tag{15a}$$

$$\lambda \geq 0 \tag{15b}$$

$$\lambda^T f(\lambda) = 0 \tag{15c}$$

Our Discrete Model

(1) Solve for λ in

$$f(\lambda) \geq 0 \tag{16a}$$

$$\lambda \geq 0 \tag{16b}$$

$$\lambda^T f(\lambda) = 0 \tag{16c}$$

(2) Then compute v^{n+1} by

$$v^{n+1} = h(\lambda) \tag{17}$$

(3) Finally compute

$$r^{n+1} = \Delta t v^{n+1} + r^n \tag{18}$$

Step 4 Goto (1) and do another simulation step

The Archetype Model

The Complementarity Problem

$$f(\lambda) \geq 0 \tag{19a}$$
$$\lambda \geq 0 \tag{19b}$$
$$\lambda^T f(\lambda) = 0 \tag{19c}$$

How can we solve problems of this type efficiently and accurately?

A Two Stage Splitting Method

In our case $f(\lambda) = A\lambda + b$ so we introduce the splitting

$$A = M - N \tag{20}$$

Next we let $c^k = b - N\lambda^k$ then

$$M\lambda^{k+1} + c^k \geq 0 \tag{21a}$$
$$\lambda^{k+1} \geq 0 \tag{21b}$$
$$(\lambda^{k+1})^T(M\lambda^{k+1} + c^k) = 0 \tag{21c}$$

This is a fixed-point formulation where the CP subproblem might be easier to solve than the original problem.

Using the Minimum Map Reformulation

Given the CP subproblem

$$M\lambda^{k+1} + c^k \geq 0 \tag{22a}$$

$$\lambda^{k+1} \geq 0 \tag{22b}$$

$$(\lambda^{k+1})^T(M\lambda^{k+1} + c^k) = 0 \tag{22c}$$

This is equivalent to

$$\min(\lambda^{k+1}, M\lambda^{k+1} + c^k) = 0 \tag{23}$$

Why?

The Minimum Map Formulation

Say a and b are complementary, that is

$$a > 0 \Rightarrow b = 0 \tag{24a}$$

$$b > 0 \Rightarrow a = 0 \tag{24b}$$

Or more compactly we write $a \geq 0 \perp b \geq 0$. Let us look at the minimum map

$\min(a, b)$	$a > 0$	$a = 0$	$a < 0$
$b > 0$	$+$	0	$-$
$b = 0$	0	0	$-$
$b < 0$	$-$	$-$	$-$

So

$$a \geq 0 \perp b \geq 0 \quad \text{iff} \quad \min(a, b) = 0 \tag{25}$$

More Clever Manipulation

So we have

$$\min(\lambda^{k+1}, M\lambda^{k+1} + c^k) = 0 \tag{26}$$

Subtract λ^{k+1}

$$\min(0, M\lambda^{k+1} + c^k - \lambda^{k+1}) = -\lambda^{k+1} \tag{27}$$

Multiply by minus one

$$\max(0, -M\lambda^{k+1} - c^k + \lambda^{k+1}) = \lambda^{k+1} \tag{28}$$

Again a fixed-point formulation

Case-by-Case Analysis

If

$$(\lambda^{k+1} - M\lambda^{k+1} - c^k)_i < 0 \tag{29}$$

Then

$$\lambda_i^{k+1} = 0 \tag{30}$$

Otherwise

$$(\lambda^{k+1} - M\lambda^{k+1} - c^k)_i = \lambda_i^{k+1} \tag{31}$$

That is

$$(M\lambda^{k+1})_i = c_i^k \tag{32}$$

Putting it Together

For suitable choice of M and back-substitution of $c^k = b - N\lambda^k$ we have

$$\left(M^{-1}\left(N\lambda^k - b\right)\right)_i = \lambda_i^{k+1} \tag{33}$$

Combining it all we have the iterative scheme

$$\max\left(0, \left(M^{-1}\left(N\lambda^k - b\right)\right)\right) = \lambda^{k+1} \tag{34}$$

Choices of M and N

The Projected Jacobi Method

$$M = D \quad \text{and} \quad N = L + U \tag{35}$$

Gauss–Seidel Method

$$M = L + D \quad \text{and} \quad N = U \tag{36}$$

SOR Method

$$M = D + \omega L \quad \text{and} \quad N = (1 - \omega) D + \omega U \tag{37}$$

Pros and Cons

- The computation cost of each iteration is comparable to a sparse matrix-vector multiplication
- The iterative projection scheme is very robust
- It has linear convergence
- Interactivity implies maximum iteration count
- High-frequency motion is lost
- Motion appear damped and objects become elastic
- More accuracy requires a zillion iterations