An iterative solver for cone complementarity problems of nonsmooth multibody dynamics—and other DVI

Mihai ANITESCU,

Argonne National Laboratories

Hybrid System Workshop, Koç University, May 2008
Team

- Mihai Anitescu
- Alessandro Tasora
  - University of Parma,
  - Author of ChronoEngine
- Dan Negrut,
  - University of Wisconsin
  - Former ADAMS developer
Nonsmooth contact dynamics—what is it?

- Differential problem with variational inequality constraints – DVI

\[
\begin{align*}
M \frac{dv}{dt} &= \sum_{j=1,2,\ldots,p} \left( c_n^{(j)} n^{(j)} + \beta_1^{(j)} t_1^{(j)} + \beta_2^{(j)} t_2^{(j)} \right) + f_c(q,v) + k(t,q,v) \\
\frac{dq}{dt} &= \Gamma(q)v \\
c_n^{(j)} &\geq 0 \quad \perp \quad \Phi^{(j)}(q) \geq 0, \quad j = 1,2,\ldots,p \\
(\beta_1^{(j)}, \beta_2^{(j)}) &= \text{argmin} \quad \mu^{(j)} c_n^{(j)} \geq \sqrt{\beta_1^{(j)^2} + \beta_2^{(j)^2}} \left[ \left( v^T t_1^{(j)} \right) \beta_1 + \left( v^T t_2^{(j)} \right) \beta_2 \right]
\end{align*}
\]

- Truly, a Differential Problem with Equilibrium Constraints

**Friction Model**

**Newton Equations**

**Non-Penetration Constraints**

**Generalized Velocities**
Differential Variational Inequalities— why do it?

- Contact Dynamics.
  - Rigid-Bodies: Differential Operator is ODE.
  - Deformable Bodies: Differential Operator is PDE.
  - Granular Flow, Masonry Stability, Rock Dynamics…

- Finance: Option Pricing-- American Options. PDE-based.

- Dynamics of multicristalline materials: evolution of the boundary between phases.

- Porous Media Flow.

See Luo, Pang et al, and Kinderlehrer and Stampacchia Monographs..
Or, just for fun .... Physics-based VR

This “fun” is serious business in the US,
One of the main drivers of new architectures (GPU, Ageia); huge user community

Implication: Speed and Stability more weight than of accuracy.

Note: real-time simulation
Question 1: Should we do smoothing?

- Recall, DVI (for C=R+): 
  \[ \dot{x} = f(t, x(t), u(t)); \]
  \[ u \geq 0 \perp F(t, x(t), u(t)) \geq 0 \]

- Smoothing: 
  \[ \dot{x} = f(t, x(t), u(t)); \]
  \[ u_i F_i(t, x(t), u(t)) = \varepsilon, \quad i = 1, 2, \ldots n_u \]

- Followed by forward Euler. Easy to implement!!
  \[ x^{n+1} = x^n + hf(t^n, x^n, u^n); \]

- Compare with the complexity of time-stepping
  \[ x^{n+1} = x^n + hf(t^{n+1}, x^{n+1}, u^{n+1}); \]
  \[ u^{n+1} \geq 0 \perp F(t^{n+1}, x^{n+1}, u^{n+1}) \geq 0 \]

- But does it give good results?
Applying ADAMS to granular flow*

- ADAMS is the workhorse of engineering dynamics.
- ADAMS/View Procedure for simulating.
- Spheres: diameter of 60 mm and a weight of 0.882 kg.
- Forces: smoothing with stiffness of 1E5, force exponent of 2.2, damping coefficient of 10.0, and a penetration depth of 0.1

* From Madsen et al.
ADAMS versus ChronoEngine 

* From Madsen et al.

**Conclusion 1:** Often, time stepping is more promising,

---

**Table 1:** Number of rigid bodies v. CPU time in ADAMS

<table>
<thead>
<tr>
<th>Number of Spheres</th>
<th>Max Number of Mutual Contacts [-]</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.41</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3.3</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>7.75</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>25.36</td>
</tr>
<tr>
<td>16</td>
<td>152</td>
<td>102.78</td>
</tr>
<tr>
<td>32</td>
<td>560</td>
<td>644.4</td>
</tr>
</tbody>
</table>

The following graph shows the nonlinear increase in the CPU time as the number of colliding bodies increases.

**Table 2:** Number of rigid bodies v. CPU time in ChronoEngine

<table>
<thead>
<tr>
<th>Number of Spheres</th>
<th>Max Number of Mutual Contacts [-]</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.70</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.73</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>0.73</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>0.76</td>
</tr>
<tr>
<td>16</td>
<td>152</td>
<td>0.82</td>
</tr>
<tr>
<td>32</td>
<td>560</td>
<td>1.32</td>
</tr>
<tr>
<td>64</td>
<td>2144</td>
<td>2.65</td>
</tr>
<tr>
<td>128</td>
<td>8384</td>
<td>6.17</td>
</tr>
<tr>
<td>256</td>
<td>33152</td>
<td>15.30</td>
</tr>
</tbody>
</table>

**Table 2** graph shows a similar trend with ChronoEngine, indicating the CPU time increases as the number of spheres increases.

---

**Conclusion 1:** Often, time stepping is more promising,
**Nonsmooth contact dynamics**

- Differential problem with equilibrium constraints – DPEC.

\[
M \frac{dv}{dt} = \sum_{j=1,2,\ldots,p} \left( c_n^{(j)} n^{(j)} + \beta_1^{(j)} t_1^{(j)} + \beta_2^{(j)} t_2^{(j)} \right) + f_c(q,v) + k(t,q,v)
\]

\[
\frac{dq}{dt} = v
\]

\[
c_n^{(j)} \geq 0 \quad \Phi^{(j)}(q) \geq 0, \quad j = 1,2,\ldots,p
\]

\[
\left( \beta_1^{(j)}, \beta_2^{(j)} \right) = \arg\min_{\mu^{(j)} c_n^{(j)} \geq \sqrt{\left( \beta_1^{(j)} + \beta_2^{(j)} \right)^2}} \left[ \left( v^T t_1^{(j)} \right) \beta_1 + \left( v^T t_2^{(j)} \right) \beta_2 \right]
\]

**Friction Model**
Where is the switching?

- When bodies enter contact (collision, plastic in the previous formulation)
- Stick-Slip transition.
Options and challenges for methods with no smoothing

- Piecewise DAE (Haug, 86)
  - Plus: Uses well understood DAE technology
  - Minus: The density of switches, switching consistency, and Painleve are problems.

- Acceleration-force time-stepping (Glocker & Pfeiffer, 1992, Pang & Trinkle, 1995)
  - Plus: No consistency problem.
  - Minus: Density of switches and Painleve.

  - Plus: No consistency, or Painleve. Some have fixed time stepping (Moreau, 198*, Anitescu & Hart 04, Anitescu, 06).
  - Minus: Nonzero restitution coefficient is tough—but its value is disputable in any case
Conic Complementarity IS NATURAL in Coulomb Models.

- Coulomb model.
  \[
  \left( \beta_1^{(j)}, \beta_2^{(j)} \right) = \operatorname{argmin}_{\mu^{(j)} c_n^{(j)} \geq \sqrt{\beta_1^{(j)} + \beta_2^{(j)}}^2} \left[ \left( v^T t_1^{(j)} \right) \beta_1 + \left( v^T t_2^{(j)} \right) \beta_2 \right]
  \]

  \[
  K = \left\{ (x, y, z) \mid \mu^{(j)} z \geq \sqrt{y^2 + x^2} \right\} \quad K^* = \left\{ (x, y, z) \mid z \geq \mu^{(j)} \sqrt{y^2 + x^2} \right\}
  \]

  \[
  \left( \begin{array}{c}
  c_n^{(j)} \\
  \beta_1^{(j)} \\
  \beta_2^{(j)} 
  \end{array} \right) \in K \quad \perp \quad \left( \begin{array}{c}
  \mu^{(j)} \sqrt{\left( v^T t_1^{(j)} \right)^2 + \left( v^T t_2^{(j)} \right)^2} \\
  v^T t_1^{(j)} \\
  v^T t_2^{(j)} 
  \end{array} \right) \in K^*
  \]

- Most previous approaches discretize friction cone to use LCP…
- Question 2: Can we still get convergence but not do that?
A measure differential inclusion solution can be obtained by time-stepping (Stewart, 1998, Anitescu 2006)

\[
M(\mathbf{v}^{(l+1)} - \mathbf{v}^l) = \sum_{i \in \mathcal{A}(\mathbf{q}^{(l)}, \epsilon)} (\gamma_n^i D_n^i + \gamma_u^i D_u^i + \gamma_v^i D_v^i) + \\
+ \sum_{i \in \mathcal{G}_B} (\gamma_b^i \nabla \psi^i) + h f_t(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)})
\]

\[
0 = \frac{1}{h} \psi^i(\mathbf{q}^{(l)}) + \nabla \psi^i T \mathbf{v}^{(l+1)} + \frac{\partial \psi^i}{\partial t}, \quad i \in \mathcal{G}_B
\]

\[
0 \leq \frac{1}{h} \Phi^i(\mathbf{q}^{(l)}) + \nabla \Phi^i T \mathbf{v}^{(l+1)}
\]

\[
\gamma_n^i \geq 0, \quad i \in \mathcal{A}(\mathbf{q}^{(l)}, \epsilon)
\]

\[
(\gamma_u^i, \gamma_v^i) = \arg\min_{\mu \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in \mathcal{A}(\mathbf{q}^{(l)}, \epsilon)
\]

\[
[\mathbf{v}^T (\gamma_u D_u^i + \gamma_v D_v^i)] = \arg\min \frac{1}{h} \nabla \psi^i T \mathbf{v}^{(l+1)}
\]

\[
\mathbf{q}^{(l+1)} = \mathbf{q}^{(l)} + h \mathbf{v}^{(l+1)},
\]
Pause: Constraint Stabilization

- Compared to original scheme

\[ \nabla \Phi(q^{(l)})^T v^{(l+1)} \geq 0 \iff \Phi^{(j)}(q^{(l)}) + \gamma h_l \nabla \Phi(q^{(l)})^T v^{(l+1)} \geq 0. \]

\[ \nabla \Theta(q^{(l)})^T v^{(l+1)} = 0 \iff \Theta^{(j)}(q^{(l)}) + \gamma h_l \nabla \Theta(q^{(l)})^T v^{(l+1)} = 0. \]

- Allows fixed time steps for plastic collisions.
- How do we know it is achieved? Infeasibility is one order better than accuracy (O(h^2))
Time Stepping -- Convex Relaxation

A modification (relaxation, to get convex QP with conic constraints):

\[
M(v^{(l+1)} - v^l) = \sum_{i \in A(q^{(l)}, \epsilon)} \left( \gamma_n^i D_n^i + \gamma_u^i D_u^i + \gamma_v^i D_v^i \right) +
\]
\[
+ \sum_{i \in G_B} \left( \gamma_b^i \nabla \psi^i \right) + h f_t(t^{(l)}, q^{(l)}, v^{(l)})
\]

\[
0 = \frac{1}{h} \psi^i(q^{(l)}) + \nabla \psi^i^T v^{(l+1)} + \frac{\partial \psi^i}{\partial t}, \quad i \in G_B
\]

\[
0 \leq \frac{1}{h} \Phi^i(q^{(l)}) + \nabla \Phi^i^T v^{(l+1)} - \mu_i \sqrt{(D_u^i T v)^2 + (D_v^i T v)^2}
\]

\[
\perp \quad \gamma_n^i \geq 0, \quad i \in A(q^{(l)}, \epsilon)
\]

\[
(\gamma_u^i, \gamma_v^i) = \arg \min_{\mu_i, \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} \quad i \in A(q^{(l)}, \epsilon)
\]

\[
q^{(l+1)} = q^{(l)} + h v^{(l+1)}
\]

(For small \( \mu \) and/or small speeds, almost no one-step differences from the Coulomb theory)

But In any case, converges to same MDI as unrelaxed scheme.

[ see M.Anitescu, “Optimization Based Simulation of Nonsmooth Rigid Body Dynamics” ]
Pause: what does convergence mean here?

We must now assign a meaning to

$$M\frac{dv}{dt} - f_c(q, v) - k(t, q, v) \in FC(q).$$

**Definition** If $\nu$ is a measure and $K(\cdot)$ is a convex-set valued mapping, we say that $\nu$ satisfies the differential inclusions

$$\frac{dv}{dt} \in K(t)$$

if, for all continuous $\phi \geq 0$ with compact support, not identically 0, we have that

$$\frac{\int \phi(t)\nu(dt)}{\int \phi(t)dt} \in \bigcup_{\tau: \phi(\tau) \neq 0} K(\tau).$$
**Pause(2) : What does convergence mean here?**

H1 The functions $n^{(j)}(q), t^{(j)}_1(q), t^{(j)}_2(q)$ are smooth and globally Lipschitz, and they are bounded in the 2-norm.

H2 The mass matrix $M$ is positive definite.

H3 The external force increases at most linearly with the velocity and position.

H4 The uniform pointed friction cone assumption holds.

Then there exists a subsequence $h_k \rightarrow 0$ where

- $q^{h_k}(\cdot) \rightarrow q(\cdot)$ uniformly.
- $v^{h_k}(\cdot) \rightarrow v(\cdot)$ pointwise a.e.
- $dv^{h_k}(\cdot) \rightarrow dv(\cdot)$ weak * as Borel measures. in $[0,T]$, and every such subsequence converges to a solution $(q(\cdot), v(\cdot))$ of MDI.
What is physical meaning of the relaxation?

- Origin

- Behavior
Further insight.

- The key is the combination between relaxation and constraint stabilization.

\[
0 \leq \frac{1}{h} \Phi^{(j)} (q^{(l)}) + \nabla q \Phi^{(j)} (q^{(l)}) v^{(l+1)} - \mu^{(j)} \sqrt{(D_{u,t}^{l} v)^2 + (D_{v,t}^{l} v)^2}
\]

- If the time step is smaller than the variation in velocity then the gap function settles at

\[
0 \approx \frac{1}{h} \Phi^{(j)} (q^{(l)}) - \mu^{(j)} \sqrt{(D_{u,t}^{l} v)^2 + (D_{v,t}^{l} v)^2}
\]

- So the solution is the same as the original scheme for a slightly perturbed gap function…..
Cone complementarity*

* Anitescu and Tasora "An iterative approach for cone complementarity problems for nonsmooth dynamics". Preprint ANL/MCS-P1413-0507

Aiming at a more compact formulation:

\[ b_A = \left\{ \frac{1}{h} \Phi i_1, 0, 0, \frac{1}{h} \Phi i_2, 0, 0, \ldots, \frac{1}{h} \Phi i_n, 0, 0 \right\} \]

\[ \gamma_A = \left\{ \gamma_{i_1}, \gamma_{i_1}, \gamma_{i_1}, \gamma_{i_1}, \gamma_{i_1}, \gamma_{i_1}, \ldots, \gamma_{i_n}, \gamma_{i_n}, \gamma_{i_n}, \gamma_{i_n} \right\} \]

\[ b_B = \left\{ \frac{1}{h} \Psi^1 + \frac{\partial \Psi^1}{\partial t}, \frac{1}{h} \Psi^2 + \frac{\partial \Psi^2}{\partial t}, \ldots, \frac{1}{h} \Psi^{n_B} + \frac{\partial \Psi^{n_B}}{\partial t} \right\} \]

\[ \gamma_B = \left\{ \gamma_{b_1}, \gamma_{b_1}, \ldots, \gamma_{b_{n_B}} \right\} \]

\[ D_A = \begin{bmatrix} D_{i_1} & D_{i_2} & \ldots & D_{i_n} \end{bmatrix}, \quad i \in A(\Theta, \epsilon) \quad D^i = \begin{bmatrix} D_{i_n}^i & D_{i_u}^i & D_{i_v}^i \end{bmatrix} \]

\[ D_B = \begin{bmatrix} \nabla \Psi^{i_1} | \nabla \Psi^{i_2} | \ldots | \nabla \Psi^{i_{n_B}} \end{bmatrix}, \quad i \in G_B \]

\[ b_\epsilon \in \mathbb{R}^{n_\epsilon} = \{ b_A, b_B \} \]

\[ \gamma_\epsilon \in \mathbb{R}^{n_\epsilon} = \{ \gamma_A, \gamma_B \} \]

\[ D_\epsilon = [D_A | D_B] \]
**Cone complementarity**

- Also define:

\[
\tilde{k}^{(l)} = M \mathbf{v}^{(l)} + h \mathbf{f}_t(t^{(l)}, q^{(l)}, \mathbf{v}^{(l)})
\]

\[
N = D_\xi^T M^{-1} D_\xi
\]

\[
r = D_\xi^T M^{-1} \tilde{k} + b_\xi
\]

- Then:

\[
M(\mathbf{v}^{(l+1)} - \mathbf{v}^l) = \sum_{i \in A(q^{(l)}, \epsilon)} (\gamma_n^i D_n^i + \gamma_u^i D_u^i + \gamma_v^i D_v^i) + \sum_{i \in \mathcal{G}_\xi} (\gamma_n^i \nabla \Psi^i + h \mathbf{f}_t(t^{(l)}, q^{(l)}, \mathbf{v}^{(l)}) + \frac{\partial \Psi^i}{\partial t}, i \in \mathcal{G}_\xi
\]

\[
0 = \frac{1}{h} \Psi^i(q^{(l)}) + \nabla \Psi^i \mathbf{v}^{(l+1)} + \frac{\partial \Psi^i}{\partial t}, i \in \mathcal{G}_\xi
\]

\[
0 \leq \frac{1}{h} \Psi^i(q^{(l)}) + \nabla \Psi^i \mathbf{v}^{(l+1)}
\]

\[
\perp \quad \gamma_n^i \geq 0, \quad \gamma_u^i \geq 0, \quad i \in A(q^{(l)}, \epsilon)
\]

\[
(\gamma_n^i, \gamma_u^i) = \arg\min_{u^T (\gamma_u^i D_u^i + \gamma_v^i D_v^i)} \left\{ u^T (\gamma_u^i D_u^i + \gamma_v^i D_v^i) \right\}
\]

\[
(N \gamma_\xi + r) \in -\gamma^\circ \perp \gamma_\xi \in \gamma
\]
Cone complementarity—Decomposable cones.

Here we introduced the convex cone

\[ \mathcal{Y} = \bigoplus_{i \in \mathcal{A}(q^l, \epsilon)} \mathcal{F}C^i \bigoplus \left( \bigoplus_{i \in \mathcal{G}_B} \mathcal{B}C^i \right) \]

..and its polar cone:

\[ \mathcal{Y}^\circ = \bigoplus_{i \in \mathcal{A}(q^l, \epsilon)} \mathcal{F}C^{i^\circ} \bigoplus \left( \bigoplus_{i \in \mathcal{G}_B} \mathcal{B}C^{i^\circ} \right) \]

CCP:

\( (N\gamma^\epsilon + r) \in -\mathcal{Y}^\circ \perp \gamma^\epsilon \in \mathcal{Y} \)

\( \mathcal{F}C^i \) is the i-th friction cone.

\( \mathcal{B}C^i \) is \( R \).

General: The iterative method

- Question 3: How to efficiently solve the Cone Complementarity Problem for large-scale systems?

\[(N\gamma_e + r) \in -\gamma^\circ \perp \gamma_e \in \gamma\]

- Our method: use a fixed-point iteration

\[\gamma^{r+1} = \lambda \Pi_\gamma \left(\gamma^r - \omega B^r \left(N\gamma^r + r + K^r \left(\gamma^{r+1} - \gamma^r\right)\right)\right) + (1 - \lambda) \gamma^r\]

- with matrices:
- ..and a non-extensive orthogonal projection operator onto feasible set

\[B^r = \begin{bmatrix}
\eta_1 I_{n_1} & 0 & \cdots & 0 \\
0 & \eta_2 I_{n_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \eta_{nk} I_{n_{nk}}
\end{bmatrix}\]

\[N^r = \begin{bmatrix}
0 & K_{12} & \cdots & K_{1nk} \\
0 & 0 & K_{23} & \cdots & K_{2nk} \\
0 & 0 & 0 & \cdots & K_{3nk} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}\]

\[\Pi_\gamma : \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_e}\]
General: The iterative method

ASSUMPTIONS

A1 The matrix $N$ of the problem (CCP) is symmetric and positive semi-definite.
A2 There exists a positive number, $\alpha > 0$ such that, at any iteration $r = 0, 1, 2, \ldots$, we have that $B^r \succ \alpha I$.
A3 There exists a positive number, $\beta > 0$ such that, at any iteration $r = 0, 1, 2, \ldots$, we have that $(x^{r+1} - x^r)^T \left( (\lambda \omega B^r)^{-1} + K^r - \frac{N}{2} \right) (x^{r+1} - x^r) \geq \beta \|x^{r+1} - x^r\|^2$.

Under the above assumptions, we can prove THEOREMS about convergence.

The method produces a bounded sequence with an unique accumulation point.
General: Theory

\[(OC) \quad \min \quad f(x) = \frac{1}{2} x^T N x + r^T x \]
\[s.t. \quad x_i \in \mathcal{Y}_i, \quad i = 1, 2, \ldots, n_k.\]

**Theorem** Assume that \(x^0 \in \mathcal{Y}\) and that the sequences of matrices \(B^r\) and \(K^r\) are bounded. Then we have that

\[f(x^{r+1}) - f(x^r) \leq -\beta \| x^{r+1} - x^r \|^2\]

for any iteration index \(r\), and any accumulation point of the sequence \(x^r\) is a solution of (CCP).

**Corollary** Assume that the friction cone of the configuration is pointed The algorithm produces a bounded sequence, and any accumulation point results in the same velocity solution

■ **Answer 2:** Simple, but first result of this nature for conic constraints—and highly efficient
The projection operator is easy and separable

- For each frictional contact constraint:

$$\Pi_T = \left\{ \Pi_{T_1}(\gamma_1)^T, \ldots, \Pi_{T_{n_A}}(\gamma_{n_A})^T, \Pi_{T_b}^1(\gamma_b)^T, \ldots, \Pi_{T_b}^{n_B}(\gamma_b^{n_B})^T \right\}^T$$

- For each bilateral constraint, simply do nothing.

- The complete operator:

$$\forall i \in A(q^{(l)}, \epsilon)$$

$$\begin{align*}
\gamma_r < \mu_i \gamma_n & \quad \Pi_i = \gamma_i \\
\gamma_r < -\frac{1}{\mu_i} \gamma_n & \quad \Pi_i = \{0, 0, 0\} \\
\gamma_r > \mu_i \gamma_n \land \gamma_r > -\frac{1}{\mu_i} \gamma_n & \quad \Pi_{i,n} = \frac{\gamma_r \mu_i + \gamma_n}{\mu_i^2 + 1} \\
\Pi_{i,u} = \gamma_u \frac{\mu_i \Pi_{i,n}}{\gamma_r} & \\
\Pi_{i,v} = \gamma_v \frac{\mu_i \Pi_{i,n}}{\gamma_r} 
\end{align*}$$
The algorithm

Development of an efficient algorithm for fixed point iteration:

- avoid temporary data, exploit sparsity. Never compute explicitly the $N$ matrix!

- implemented in incremental form. Compute only deltas of multipliers.

- $O(n)$ space requirements and supports premature termination

- for real-time purposes: $O(n)$ time
The algorithm is specialized, for minimum memory use!

(1) // Pre-compute some data for friction constraints
(2) for i := 1 to n_A
(3)  \( s_a^i = M^{-1} D^i \)
(4)  \( g_a^i = D_{i,T} s_a^i \)
(5)  \( \eta_i^a = \frac{1}{3} \text{Trace}(g_a^i) \)
(6) // Pre-compute some data for bilateral constraints
(7) for i := 1 to n_B
(8)  \( s_b^i = M^{-1} \nabla \Psi^i \)
(9)  \( g_b^i = \nabla \Psi^i, T s_b^i \)
(10)  \( \eta_b^i = \frac{1}{g_b^i} \)
(11) // Initialize impulses
(12) if warm start with initial guess \( \gamma^*_E \)
(13)  \( \gamma_E^0 = \gamma^*_E \)
(14) else
(15)  \( \gamma_E^0 = 0 \)
(16) // Initialize speeds
(17)  \( v = \sum_{i=1}^{n_A} s_a^i \gamma_i^a + \sum_{i=1}^{n_B} s_b^i \gamma_i^b + M^{-1} \dot{k} \)
(18) // Main iteration loop
(19) for r := 0 to r_{max}
(20) // Loop on frictional constraints
(21) for i := 1 to n_A
(22)  \( \delta^{i,r} = (\gamma_i^{i,r} - \omega \eta_i^a (D^i_{i,T} v^r + b^a_i)) \)
(23)  \( \gamma_i^{i,r+1} = \lambda \Pi \chi (\delta^{i,r}) + (1 - \lambda) \gamma_i^{i,r} \)
(24)  \( \Delta \gamma_i^{i,r+1} = \gamma_i^{i,r+1} - \gamma_i^{i,r} \)
(25)  \( v := v + s_a^i \Delta \gamma_i^{i,r+1} \)
(26) // Loop on bilateral constraints
(27) for i := 1 to n_B
(28)  \( \delta^{i,r} = (\gamma_i^{i,r} - \omega \eta_i^b (\nabla \Psi^i, T v^r + b^b_i)) \)
(29)  \( \gamma_i^{i,r+1} = \lambda \Pi \chi (\delta^{i,r}) + (1 - \lambda) \gamma_i^{i,r} \)
(30)  \( \Delta \gamma_i^{i,r+1} = \gamma_i^{i,r+1} - \gamma_i^{i,r} \)
(31)  \( v := v + s_b^i \Delta \gamma_i^{i,r+1} \)
(32) return \( \gamma_E, v \)
Simulating the PBR nuclear reactor

- The PBR nuclear reactor:
  - Fourth generation design
  - Inherently safe, by Doppler broadening of fission cross section
  - Helium cooled > 1000 °C
  - Can crack water (mass production of hydrogen)
  - Continuous cycling of 360'000 graphite spheres in a pebble bed
Simulating the PBR nuclear reactor

- Problem of bidisperse granular flow with dense packing.
- Previous attempts: DEM methods on supercomputers at Sandia Labs regularization)
- 40 seconds of simulation for 440,000 pebbles needs 1 week on 64 processors dedicated cluster (Rycroft et al.)

model a frictionless wall, $\mu_w=0.0$. For the current simulations we set $k_s = \frac{2}{7}k_n$ and choose $k_n = 2 \times 10^5 \text{gm/d}$. While this is significantly less than would be realistic for graphite pebbles, where we expect $k_n > 10^{10} \text{gm/d}$, such a spring constant would be prohibitively computationally expensive, as the time step scales as $\delta t \propto k_n^{-1/2}$ for collisions to be modeled effectively. Previous simulations have shown that
Simulating the PBR nucleation process:

- 160,000 Uranium-Graphite spheres, 600,000 contacts on average
- Two millions of primal variables, six millions of dual variables
- 1 day on a Windows station…
- But we are limited by the 2GB user mode limit, 64 bit port in progress—but linear scaling..
- We estimate 3CPU days, compare with 450 CPU days for an incomplete solution in 2006 !!!
- Answer 3: Our approach is efficient for large scale!!
In addition, we can approach efficiently approach many engineering problems (see website for papers)
Examples

- Example: size-segregation in shaker, with thousands of steel spheres

Note: solution beyond reach of Lemke-type LCP solvers!
Tests

Feasibility accuracy increases with number of iterations:

- Speed violation in constraints
- Position error in constraints (penetration)

(with example of 300 spheres in shaker)
Tests: Scalability

- CPU effort per contact, since our contacts are the problem variables.
- Penetration error was uniformly no larger than 0.2% of diameter.

![Graph of number of contacts over time steps.](image1)

![Graph of average CPU time per step.](image2)

Number of contacts in time, 300 spheres

CPU time per step for 300-1500 spheres
New large scale computational opportunity
Graphical Processing Unit *

Floating Point Operations per Second for the CPU and GPU

G80 = GeForce 8800 GTX
G71 = GeForce 7900 GTX
G70 = GeForce 7800 GTX
NV40 = GeForce 6800 Ultra
NV35 = GeForce FX 5950 Ultra
NV30 = GeForce FX 5800

* NVIDIA CUDA
Compute Unified Device Architecture
Programming Guide
**IBM BlueGene/L—GPU comparison**

- Entry model: 1024 dual core nodes
- 5.7 Tflop (compare to 0.5 Tflop for NVIDIA Tesla GPU)
- Dedicated OS
- Dedicated power management solution
- Require dedicated IT support
- Price (2007): $1.4 million
- Same GPU power (2008): 7K!!!

- Of course, GPU much harder to work with at the moment, and unsuitable for general purpose computing.
Brick Wall Example *

- Times reported are in seconds for one second long simulation
- GPU: NVIDIA GeForce 8800 GTX

---

*Alessandro Tasora, Dan Negrut and Mihai Anitescu. "Large-Scale Parallel Multibody Dynamics with Frictional Contact on the Graphical Processing Unit ". Preprint ANL/MCS-P1494-0508

<table>
<thead>
<tr>
<th>Bricks</th>
<th>Sequential Version</th>
<th>GPU Co-processing Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>43</td>
<td>6</td>
</tr>
<tr>
<td>2000</td>
<td>87</td>
<td>10</td>
</tr>
<tr>
<td>8000</td>
<td>319</td>
<td>42</td>
</tr>
</tbody>
</table>
Future work

- \(N\) non symmetric, but positive semidefinite.
- Parallelizing the algorithms: block Jacobi with Gauss Seidel blocks.
- Asynchronous version of the algorithm, particularly for use with GPU.
- Including a good collision model—here we are at a loss with rigid body theory—may need some measure of deformability.
- Compare with experimental data.
Conclusions

- We have defined a new algorithm for complementarity problems with conic constraints.
- We have shown that it can solve very large problems in granular flow far faster than DEM.
- It is the first iterative algorithm that provably converges for nonsmooth rigid body dynamics.
- Its scalability is decent.
- We have created a multithreaded implementation and GPU port increases computational speed by a factor of 7-8.
References (preprints are at authors’ web site)