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

Mihai ANITESCU
Argonne National Laboratory

UIC Math Applications Seminar
Feb 11, 2009
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

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

\[ \frac{dq}{dt} = \Gamma(q) v \]

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

\[ \left( \beta_1^{(j)}, \beta_2^{(j)} \right) = \text{argmin} \mu^{(j)} c_n^{(j)} \sqrt{\beta_1^{(j)} + \beta_2^{(j)}} \left[ (v^T t_1^{(j)}) \beta_1 + (v^T t_2^{(j)}) \beta_2 \right] \]

- Truly, a Differential Problem with Equilibrium Constraints

Friction Model

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

Note: real-time simulation

Implication:
Speed and Stability more weight than of accuracy.
Question 1: Should we do smoothing?

- Recall, DVI (for C=R+)

\[
\dot{x} = f(t, x(t), u(t)); \\
0 \leq u \perp F(t, x(t), u(t)) \geq 0
\]

- 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
ADAMS versus ChronoEngine

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>

The graph shows the CPU time in ChronoEngine as a function of the number of spheres.

CPU time v. Number of spheres in ChronoEngine

\[ y = 0.0563x + 0.0446 \]

\[ R^2 = 0.9782 \]
Nonsmooth contact dynamics

Differential problem with equilibrium constraints – DPEC.

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

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

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

\[
\left( \beta_1^{(j)}, \beta_2^{(j)} \right) = \text{argmin} \left\{ \mu^{(j)} c_n^{(j)} \sqrt{\beta_1^{(j)} + \beta_2^{(j)}} \left[ \left( v^T t_1^{(j)} \right) \beta_1 + \left( v^T t_2^{(j)} \right) \beta_2 \right] \right\}
\]

Friction Model
It is a hybrid system – 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.

\[
\begin{align*}
\left( \beta_1^{(j)}, \beta_2^{(j)} \right) &= \arg\min_{\mu^{(j)} c_n^{(j)} \geq \sqrt{\left( \beta_1^{(j)} + \beta_2^{(j)} \right)^2}} \left[ \left( v^T t_1^{(j)} \right) \beta_1 + \left( v^T t_2^{(j)} \right) \beta_2 \right] \\
K &= \left\{ (x, y, z) \bigg| \mu^{(j)} z \geq \sqrt{y^2 + x^2} \right\} \\
K^* &= \left\{ (x, y, z) \bigg| z \geq \mu^{(j)} \sqrt{y^2 + x^2} \right\} \\
\begin{pmatrix} c_n^{(j)} \\ \beta_1^{(j)} \\ \beta_2^{(j)} \end{pmatrix} &\in K \\
\begin{pmatrix} \mu^{(j)} \sqrt{\left( v^T t_1^{(j)} \right)^2 + \left( v^T t_2^{(j)} \right)^2} \\ v^T t_1^{(j)} \\ v^T t_2^{(j)} \end{pmatrix} &\in K^*
\end{align*}
\]

- Most previous approaches discretize friction cone to use LCP…
- Question 2: Can we still get convergence but not do that?
Time stepping scheme -- original

A measure differential inclusion solution can be obtained by time-stepping (Stewart, 1998, Anitescu 2006)

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

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

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

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

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

\[
\left[ \mathbf{v}^T (\gamma_u D_u^i + \gamma_v D_v^i) \right]
\]

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

- Compared to original scheme

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

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

- 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):

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

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

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

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

\[
(\gamma_u^i, \gamma_v^i) = \arg \min_{\mu^i \gamma_n^i \geq \sqrt{(\gamma_u^i)^2 + (\gamma_v^i)^2}} (\gamma_u D_u^i + \gamma_v D_v^i)
\]

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

(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 $n^{(j)}(q), t_1^{(j)}(q), t_2^{(j)}(q)$ 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 $h_k \to 0$ where

- $q^{h_k}(\cdot) \to q(\cdot)$ uniformly.
- $v^{h_k}(\cdot) \to v(\cdot)$ pointwise a.e.
- $dv^{h_k}(\cdot) \to dv(\cdot)$ 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.

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

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

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

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

- Aiming at a more compact formulation:

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

\[ \gamma_A = \left\{ \gamma^i_1, \gamma^i_1, \gamma^i_1, \gamma^i_2, \gamma^i_2, \gamma^i_2, \ldots, \gamma^i_{n_A}, \gamma^i_{n_A}, \gamma^i_{n_A} \right\} \]

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

\[ \gamma_B = \left\{ \gamma^b_1, \gamma^b_2, \ldots, \gamma^b_{n_B} \right\} \]

\[ D_A = [D^i_1 | D^i_2 | \ldots | D^i_{n_A}], \quad i \in A(q^l, \epsilon) \]

\[ D^i = [D^i_n | D^i_u | D^i_v] \]

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

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

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

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

Also define:

\[ \tilde{k}^{(l)} = Mv^{(l)} + hf_t(t^{(l)}, q^{(l)}, v^{(l)}) \]

\[ N = D^T_\mathcal{E}M^{-1}D_\mathcal{E} \]

\[ r = D^T_\mathcal{E}M^{-1}\tilde{k} + b_\mathcal{E} \]

Then:

\[ \sum_{i \in A(q^{(l)}, \epsilon)} (\gamma_n^i D^i_n + \gamma_u^i D^i_u + \gamma_v^i D^i_v) + \sum_{i \in G^e} (\gamma_n^i \nabla \Psi^i) + hf_t(t^{(l)}, q^{(l)}, v^{(l)}) \]

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

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

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

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

\[ [v^T(\gamma_n^i D^i_n + \gamma_v^i D^i_v)] \]

This becomes...

\[ (N\gamma_\mathcal{E} + r) \in -\gamma^\circ \perp \gamma_\mathcal{E} \in \gamma \]

This is a CCP, CONE COMPLEMENTARITY PROBLEM
Cone complementarity—Decomposable cones.

Here we introduced the convex cone

\[ \mathcal{Y} = \left( \bigoplus_{i \in A(q^l, \epsilon)} \mathcal{FC}^i \right) \bigoplus \left( \bigoplus_{i \in G_B} \mathcal{BC}^i \right) \]

..and its polar cone:

\[ \mathcal{Y}^o = \left( \bigoplus_{i \in A(q^l, \epsilon)} \mathcal{FC}^{i\circ} \right) \bigoplus \left( \bigoplus_{i \in G_B} \mathcal{BC}^{i\circ} \right) \]

\[ \mathcal{FC}^i \quad \text{In } \mathbb{R}^3 \text{ is } i\text{-th friction cone} \]
\[ \mathcal{BC}^i \quad \text{is } \mathbb{R} \]

CCP:

\[ (N\gamma \epsilon + r) \in -\mathcal{Y}^o \perp \gamma \epsilon \in \mathcal{Y} \]
Question 3: How to efficiently solve the Cone Complementarity Problem for large-scale systems?

Our method: use a fixed-point iteration

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

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

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

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

\[
\Pi_{\mathcal{Y}} : \mathbb{R}^{n_{\mathcal{E}}} \rightarrow \mathbb{R}^{n_{\mathcal{E}}}
\]
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 $r$, $r = 0, 1, 2, \ldots$, we have that $B^r \succ \alpha I$.
  
  A3 There exists a positive number, $\beta > 0$ such that, at any iteration $r$, $r = 0, 1, 2, \ldots$, 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

\[
\begin{align*}
(OC) \quad & \min_{x} \quad f(x) = \frac{1}{2} x^T N x + r^T x \\
& \text{s.t.} \quad x_i \in \mathcal{Y}^i, \quad i = 1, 2, \ldots, n_k.
\end{align*}
\]

**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:
  \[ \Pi_Y = \left\{ \Pi_{Y_1}(\gamma_1)^T, \ldots, \Pi_{Y_n}(\gamma_n)^T, \Pi_{b_1}^1(\gamma_b^1), \ldots, \Pi_{b_n}^n(\gamma_b^n) \right\}^T \]
  
- For each bilateral constraint, simply do nothing.

The complete operator:

For all \( i \in A(q^{(l)}, c) \):
- \( \gamma_r < \mu_i \gamma_n \)
- \( \gamma_r < -\frac{1}{\mu_i} \gamma_n \)
- \( \gamma_r > \mu_i \gamma_n \land \gamma_r > -\frac{1}{\mu_i} \gamma_n \)

- \( \Pi_i = \gamma_i \)
- \( \Pi_i = \{0, 0, 0\} \)
- \( \Pi_{i,n} = \frac{\gamma_r \mu_i + \gamma_n}{\mu_i^2 + 1} \)
- \( \Pi_{i,u} = \gamma_u \frac{\mu_i \Pi_{i,n}}{\gamma_r} \)
- \( \Pi_{i,v} = \gamma_v \frac{\mu_i \Pi_{i,n}}{\gamma_r} \)
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) \( s^i_a = M^{-1} D^i \)
(4) \( g^i_a = D^i, T s^i_a \)
(5) \( \eta^i_a = \frac{1}{\text{Trace}(g^i_a)} \)
(6) // Pre-compute some data for bilateral constraints
(7) for \( i := 1 \) to \( n_B \)
(8) \( s^i_b = M^{-1} \nabla \Psi^i \)
(9) \( g^i_b = \nabla \Psi^i, T s^i_b \)
(10) \( \eta^i_b = \frac{1}{g^i_b} \)
(11)
(12) // Initialize impulses
(13) if warm start with initial guess \( \gamma^*_\mathcal{E} \)
(14) \( \gamma^0_\mathcal{E} = \gamma^*_\mathcal{E} \)
(15) else
(16) \( \gamma^0_\mathcal{E} = 0 \)
(17)
(18) // Initialize speeds
(19) \( v = \sum_{i=1}^{n_A} s^i_a \gamma_a^{i,0} + \sum_{i=1}^{n_B} s^i_b \gamma_b^{i,0} + M^{-1} \ddot{k} \)
(20)
(21) // Main iteration loop
(22) for \( r := 0 \) to \( r_{\text{max}} \)
(23)
(24)
(25) \( \delta^{i,r} = (\gamma_a^{i,r} - \omega \eta_a^{i} \left( D^{i,T} v^r + b^i_a \right)) \)
(26) \( \gamma_a^{i,r+1} = \lambda \Pi_T \left( \delta^{i,r} \right) + (1 - \lambda) \gamma_a^{i,r} \)
(27) \( \Delta \gamma_a^{i,r+1} = \gamma_a^{i,r+1} - \gamma_a^{i,r} \)
(28) \( v := v + s^i_a \Delta \gamma_a^{i,r+1} \)
(29)
(30)
(31) \( \delta^{i,r} = (\gamma_b^{i,r} - \omega \eta_b^{i} \left( \nabla \Psi^{i,T} v^r + b^i_b \right)) \)
(32) \( \gamma_b^{i,r+1} = \lambda \Pi_T \left( \delta^{i,r} \right) + (1 - \lambda) \gamma_b^{i,r} \)
(33) \( \Delta \gamma_b^{i,r+1} = \gamma_b^{i,r+1} - \gamma_b^{i,r} \)
(34) \( v := v + s^i_b \Delta \gamma_b^{i,r+1} \)
(35)
(36) return \( \gamma_\mathcal{E}, v \)
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 $k_z = \frac{2}{7} k_n$ and choose $k_n = 2 \times 10^5 \text{ gml/d}$. While this is significantly less than would be realistic for graphite pebbles, where we expect $k_n > 10^{10} \text{ gml/d}$, 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 nuclear reactor

- **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.

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

GFLOPS

Jan-03 Jul-03 Feb-04 Aug-04 Mar-05 Sep-05 Apr-06 Nov-06

Intel CPU
NVIDIA GeForce GPU

Argonne National Laboratory
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!!!
Brick Wall Example…

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

<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>
Preliminary Results for GS on large problems on GPU
Future work

- N non symmetric, but positive semidefinite.
- Parallelizing the algorithms: block Jacobi with Gauss Seidel blocks, or coloring GS (50% there).
- Huge scale simulation. Multigrid for rigid multibody dynamics?
- Including a good collision model—here we are at a loss with rigid body theory—it may need some measure of deformability—convolution complementarity.
- Involving other physics … fluid flow.
- Compare with experimental data.
Comparison with experimental data PBR