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

Mihai ANITESTCU ,
Argonne National Laboratory

With D. Negrut (Wisconsin) and A. Tasora (Parma)
Nonsmooth contact dynamics—what is it?

- Differential problem with variational inequality constraints – DVI
  - **Newton Equations**
    \[ M \frac{dv}{dt} = \sum_{j=1,2,...,p} \left( c_n^{(j)} n^{(j)} + \beta_1^{(j)} t_1^{(j)} + \beta_2^{(j)} t_2^{(j)} \right) + f_c(q,v) + k(t,q,v) \]
  - **Non-Penetration Constraints**
    \[ \frac{dq}{dt} = \Gamma(q)v \]
    \[ c_n^{(j)} \geq 0 \perp \Phi^{(j)}(q) \geq 0, \quad j = 1,2,...,p \]
    \[ \beta_1^{(j)}, \beta_2^{(j)} = \text{argmin} \quad \mu^{(j)} c_n^{(j)} z\left( \beta_1^{(j)} \right) \beta_1^{(j)} + \left( v^T t_1^{(j)} \right) \beta_1^{(j)} + \left( v^T t_2^{(j)} \right) \beta_2^{(j)} \]
  - **Generalized Velocities**
  - **Friction Model**

- Truly, a Differential Problem with Equilibrium Constraints
Contact dynamics applications

- Granular Flow,
- Masonry Stability,
- Rock Dynamics.
- Agent-Based Modeling (Pedestrian Evacuation Dynamics).
- Physics-based graphics simulations.
- Most common approaches by far are smoothing approaches (DEM)
Question 1: Should we do smoothing?

Recall, DVI (for C=R+)

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

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!!

\[ u_i^n F_i(t^{n-1}, x^{n-1}, u^{n-1}) = \varepsilon, \quad i = 1, 2, \ldots n_u \]
\[ x^{n+1} = x^n + hf(t^n, x^n, u^n); \]

Compare with the complexity of time-stepping

\[ x^{n+1} = x^n + hf(t^{n+1}, x^{n+1}, u^{n+1}); \]
\[ u^{n+1} \geq 0 \quad \nabla F(t^{n+1}, x^{n+1}, u^{n+1}) \geq 0 \]

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**

**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>

**Conclusion 1: Often, time stepping is more promising,**
Time stepping scheme -- original

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

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

- \[0 = \frac{1}{h} \Psi^i(q^{(l)}) + \nabla \Psi^i T \mathbf{v}^{(l+1)} + \frac{\partial \Psi^i}{\partial t}, \quad i \in \mathcal{G}_B\]
- \[0 \leq \frac{1}{h} \Phi^i(q^{(l)}) + \nabla \Phi^i T \mathbf{v}^{(l+1)}\]
- \[\gamma_n^i \geq 0, \quad i \in \mathcal{A}(q^{(l)}, \epsilon)\]

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

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

\[q^{(l+1)} = q^{(l)} + h \mathbf{v}^{(l+1)},\]
**Time Stepping -- Convex Relaxation**

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

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

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

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

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

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

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

(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” ]
Cone complementarity

- Aiming at a more compact formulation:

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

\[ \gamma_A = \left\{ \gamma_{n_1}^{i_1}, \gamma_{u_1}^{i_1}, \gamma_{v_1}^{i_1}, \gamma_{n_2}^{i_2}, \gamma_{u_2}^{i_2}, \gamma_{v_2}^{i_2}, \ldots, \gamma_{n_A}^{i_n_A}, \gamma_{u_A}^{i_n_A}, \gamma_{v_A}^{i_n_A} \right\} \]

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

\[ \gamma_B = \left\{ \gamma_1^1, \gamma_2^1, \ldots, \gamma_{n_B}^1 \right\} \]

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

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

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

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

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

\[ D_\epsilon = [D_A | D_B] \]
Cone complementarity problem at each step

Also define:

\[ \tilde{k}^{(l)} = M \nu^{(l)} + h f_t(t^{(l)}, q^{(l)}, \nu^{(l)}) \]
\[ N = D_E^T M^{-1} D_E \]
\[ r = D_E^T M^{-1} \tilde{k} + b_E \]

Then:

\[ (N \gamma_E + r) \in \mathcal{Y}^\circ \perp \gamma_E \in \mathcal{Y} \]
Cone complementarity—Decomposable cones.

- Here we introduced the convex cone
  \[ \gamma = \left( \bigoplus_{i \in A(q^i, \epsilon)} FC_i \right) \bigoplus \left( \bigoplus_{i \in G_B} BC_i \right) \]

- and its polar cone:
  \[ \gamma^\circ = \left( \bigoplus_{i \in A(q^i, \epsilon)} FC_{i^\circ} \right) \bigoplus \left( \bigoplus_{i \in G_B} BC_{i^\circ} \right) \]

CCP:
\[ (N\gamma\epsilon + r) \in -\gamma^\circ \perp \gamma\epsilon \in \gamma \]
General: The iterative method

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

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

Our method: use a fixed-point iteration

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

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

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

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

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

- ASSUMPTIONS

A1 The matrix $N$ of the problem (CPP) 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.
The projection operator is easy and separable

- For each frictional contact constraint:
  \[ \Pi_Y = \left\{ \Pi_{Y_1}(\gamma_1)^T, \ldots, \Pi_{Y_n,\sigma}(\gamma_{n,\sigma})^T, \Pi_{b_1}(\gamma_{b_1})^T, \ldots, \Pi_{b_{n_b}}(\gamma_{b_{n_b}})^T \right\}^T \]

- For each bilateral constraint, simply do nothing.
- The complete operator:

\[
\forall i \in A(q^{(t)}, e) \\
\begin{align*}
\gamma_r &< \mu \gamma_n \\
\gamma_r &< -\frac{1}{\mu} \gamma_n \\
\gamma_r &> \mu \gamma_n \wedge \gamma_r > -\frac{1}{\mu} \gamma_n \\
\Pi_i &= \gamma_i \\
\Pi_i &= \{0, 0, 0\} \\
\Pi_{i,n} &= \frac{\gamma_r + \gamma_n}{\mu + 1} \\
\Pi_{i,u} &= \gamma_u \frac{\gamma_i}{\gamma_r} \\
\Pi_{i,v} &= \gamma_v \frac{\gamma_i}{\gamma_r}
\end{align*}
\]
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)
Scaling/constraint accuracy test:

- 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.

**Graphs:**
- Number of contacts in time, 300 spheres
- CPU time per step for 300-1500 spheres
Preliminary Results for GS on large problems on GPU

<table>
<thead>
<tr>
<th>Number of Bodies</th>
<th>CCP Time</th>
<th>CD Time</th>
<th>Step Time</th>
<th>CCP Time</th>
<th>CD Time</th>
<th>Step Time</th>
<th>CCP Speedup</th>
<th>CD Speedup</th>
<th>Step Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>8000</td>
<td>5.26725</td>
<td>0.10655</td>
<td>5.47370</td>
<td>0.39600</td>
<td>0.12288</td>
<td>0.54658</td>
<td>13.3010</td>
<td>0.8671</td>
<td>10.0144</td>
</tr>
<tr>
<td>16000</td>
<td>11.18999</td>
<td>0.23601</td>
<td>11.63754</td>
<td>0.74096</td>
<td>0.15337</td>
<td>0.95315</td>
<td>15.1020</td>
<td>1.5388</td>
<td>12.2095</td>
</tr>
<tr>
<td>32000</td>
<td>23.28647</td>
<td>0.55861</td>
<td>24.28257</td>
<td>1.46889</td>
<td>0.21752</td>
<td>1.80878</td>
<td>15.8531</td>
<td>2.5680</td>
<td>13.4249</td>
</tr>
<tr>
<td>64000</td>
<td>49.16970</td>
<td>1.36559</td>
<td>51.44442</td>
<td>3.12925</td>
<td>0.36476</td>
<td>3.75152</td>
<td>15.7130</td>
<td>3.7438</td>
<td>13.7130</td>
</tr>
<tr>
<td>128000</td>
<td>103.66658</td>
<td>3.80176</td>
<td>109.34287</td>
<td>6.97682</td>
<td>0.74488</td>
<td>8.27050</td>
<td>14.8587</td>
<td>5.1038</td>
<td>13.2208</td>
</tr>
</tbody>
</table>
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—may need some measure of deformability—convolution complementarity.
- Involving other physics ... fluid flow.
- Compare with experimental data.
Comparison with experimental data PBR
Packing statistics
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 comparable to the number of stream processors.