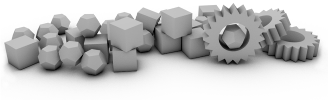
Nonsmooth Nonlinear Conjugate Gradient Method for Interactive Contact Force Computation

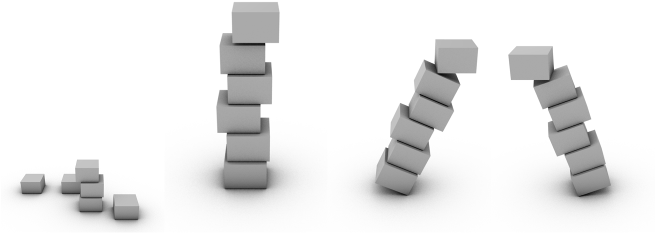
Morten Silcowitz **Sarah Niebe** **Kenny Erleben**

Department of Computer Science
University of Copenhagen
Denmark

June 2010



What is it all about?

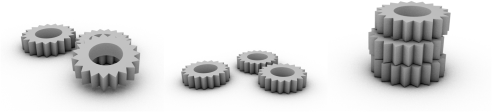


An Interactive Goal Oriented Task

The Interactive Requirements

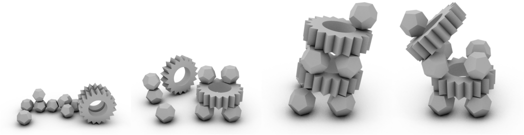
Must

- Be responsive
- Be interactive
- Have high fidelity



Why is it a Hard Problem?

- Computational very heavy
- Limited time to compute result
- End user is unpredictable
- End user has divine power $\approx$ very unphysical
- "Traditionally" proof of existence of multiple solutions

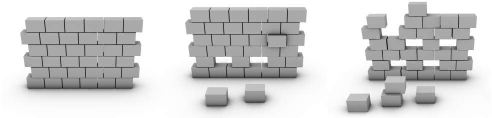


Solution: Iterative methods!

The State–of–the–Art Iterative Method

Projected Gauss–Seidel (PGS) from ODE, Bullet etc..

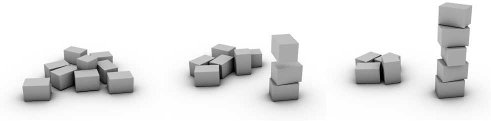
- Linear Convergence Rate
- Inaccurate Friction Forces
- Incredible Robust



Consequences: Great for blow them up physics but simulation is an art form!

Facts of Life and Hypotheses

- Better accuracy improves fidelity
- With better convergence rate one gets better accuracy in fewer iterations
- A Newton method is great but
  - Lack robustness!
  - Iteration cost is high!
  - Performance is unpredictable!



Desire: Something that behaves like PGS but with better convergence!

The Contact Force Problem

Deriving it. . .

- Start with classical mechanics
- Add some approximations
- Do a whole bunch of mathemagical tricks

then our final problem is to find λ where

$$\mathbf{y} = \mathbf{A}\lambda + \mathbf{b}$$

and for each i we have

$$\begin{aligned} y_i < 0 &\Rightarrow \lambda_i = \mathbf{u}_i(\lambda), \\ y_i > 0 &\Rightarrow \lambda_i = \mathbf{l}_i(\lambda), \\ y_i = 0 &\Rightarrow \mathbf{l}_i(\lambda) \leq \lambda_i \leq \mathbf{u}_i(\lambda). \end{aligned}$$

where $\mathbf{l}_i$ and $\mathbf{u}_i$ are affine functions modeling friction bounds and non-negative normal forces.

An Quick Observation

The Contact Force Problem is a Nonlinear Complementarity Problem (NCP)

Oh no! If only we had a mixed Linear Complementary Problem then

We could use a Projected Conjugate Gradient method on a Quadratic Programming Problem reformulation

That would give us

- Same iteration cost as PGS
- But quadratic convergence rate instead of linear

Projected Gauss-Seidel Method (PGS)

Using a splitting of $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$

and a whole lot of math later ...

we have the Projected Gauss–Seidel Method

$$\lambda^{k+1} \leftarrow \min(\mathbf{u}(\lambda^k), \max(\mathbf{l}(\lambda^k), (\mathbf{D} + \mathbf{L})^{-1} (\mathbf{U}\lambda^k - \mathbf{b})))$$

One iteration of PGS can be implemented nicely as a forward loop we write it as

$$\lambda^{k+1} = \mathbf{PGS}(\lambda^k)$$

The Idea Part One

Assuming convergence $\lambda^k \rightarrow \lambda^*$ for $k \rightarrow \infty$

$$\lambda^* = \underbrace{\min(\mathbf{u}(\lambda^*), \max(\mathbf{l}(\lambda^*), (\mathbf{D} + \mathbf{L})^{-1} (\mathbf{U}\lambda^* - \mathbf{b})))}_{\simeq \mathbf{H}\lambda^* + \mathbf{h}},$$

Conceptually evaluation of an affine function

$$0 = (\mathbf{H} - \mathbf{I})\lambda^* + \mathbf{h}$$

Residual is given by

$$\mathbf{r}^k = (\mathbf{H} - \mathbf{I})\lambda^k + \mathbf{h} = \mathbf{PGS}(\lambda^k) - \lambda^k$$

The Idea Part Two

Define a non-smooth nonlinear quasi-quadratic function

$$f(\lambda^k) \equiv \frac{1}{2} \parallel \mathbf{r}^k \parallel^2$$

Gradient is given by

$$\nabla f(\lambda^k) = -\mathbf{r}^k$$

Oh! Gradient Descent method is equivalent to the PGS method

Nonlinear Nonsmooth Conjugate Gradients (NNCG)

Use a Fletcher–Reeves nonlinear conjugate gradient method on f. Update step,

$$\lambda^{k+1} = \lambda^k + \tau^k \mathbf{p}^k$$

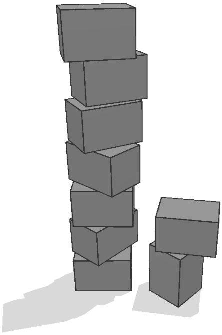
$\mathbf{p}^k$ is the search direction and and τ^k is the line step. Initially $\mathbf{p}^1 = -\nabla f^1$,

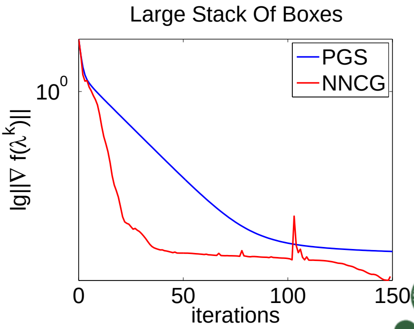
$$\beta^{k+1} = \frac{\| \nabla f^{k+1} \|^2}{\| \nabla f^k \|^2}$$
$$\mathbf{p}^{k+1} = \beta^{k+1}\mathbf{p}^k - \nabla f^{k+1}$$

Restart if $\|\nabla f^{k+1}\|^2 > \|\nabla f^k\|^2$.

Interactive Results

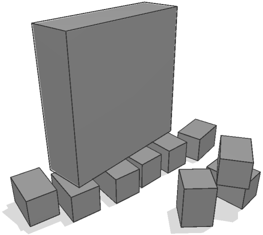
- Fast convergence for small sized stacks

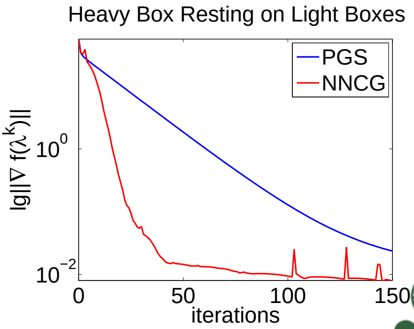




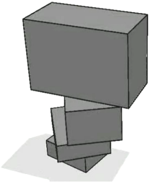
Interactive Results

- Better convergence for large mass ratios





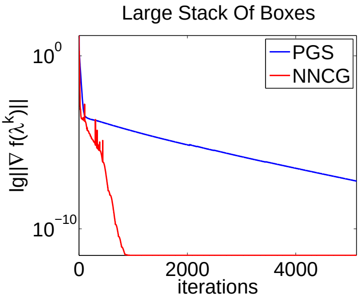
NNCG Method in Action

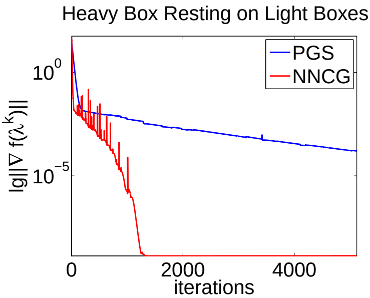


Interactive Comparisons with PGS

Non-Interactive Results

- Quadratic convergence rates was observed
- Notice restart spikes





Conclusion and Future Work

Findings

- Roughly same iteration cost as PGS
- Lower error level than PGS
- Super linear and even quadratic convergence rate
- Best for small problems with obvious structure

Speculation

- Over-determinacy causes inability to improve convergence on larger and more complex problems

Future work

- Preconditioning or multigrid techniques

Thank you for attending this talk!