Position-based Dynamics

for ODEs with inequality constraints

Steffen Plunder
Supervisor: Sara Merino-Aceituno

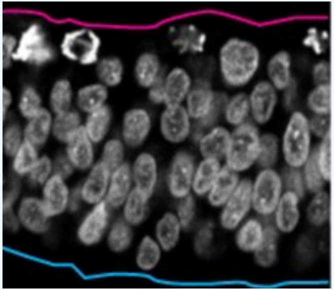
PDE Afternoon

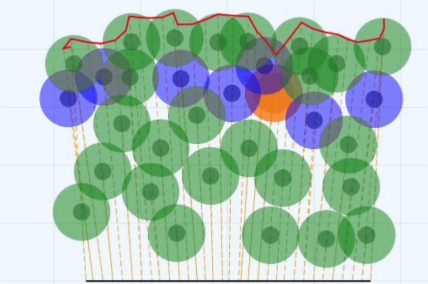
10 Nov 2021

WWTF
Vienna Science and Technology Fund
universität wien

Motivation

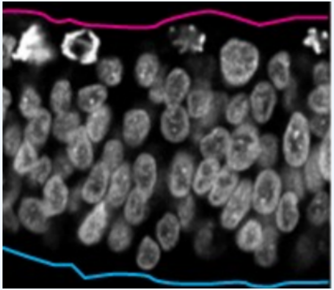
Simulation of epithelial cells

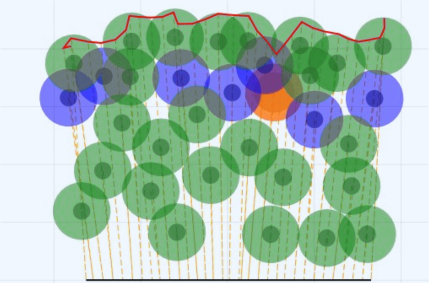




What I want to do ☺

Simulation of epithelial cells





But: There are inequality constraints in the model:

- **non-overlapping constraints** between nuclei cores,
- black line is a **chain of links** with fixed maximal length.

What I have to do ☹

1. Solve an ODE

$$\dot{x} = f(x) + \dots$$

with (many) inequality constraints

$$g_k(x(t)) \geq 0 \quad \text{for all } k.$$

What I have to do ☹

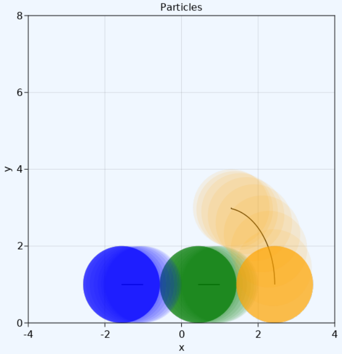
1. Solve an ODE

$$\dot{x} = f(x) + \dots$$

with (many) inequality constraints

$$g_k(x(t)) \geq 0 \quad \text{for all } k.$$

2. Do it fast...



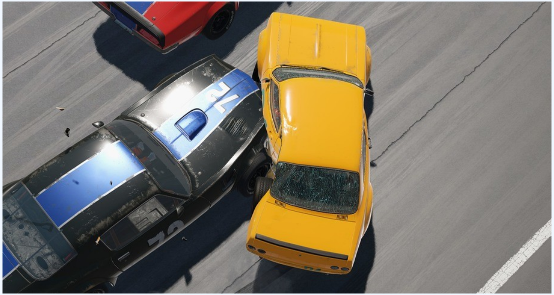
What I found (while not doing my PhD)

I needed to wait overnight for simulation studies with a model of just 30 cells.

What I found (while not doing my PhD)

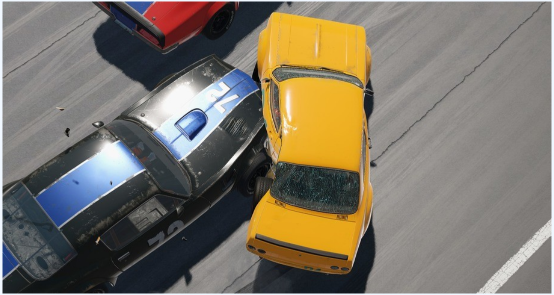
I needed to wait overnight for simulation studies with a model of just 30 cells.

At the same time, I can crash into inequality constraints in real-time...



I needed to wait overnight for simulation studies with a model of just 30 cells.

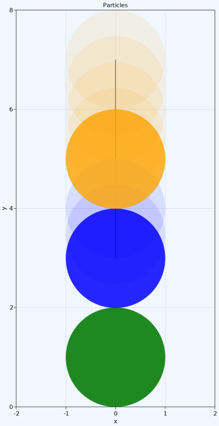
At the same time, I can crash into inequality constraints in real-time...



Computer graphics uses **Position-based Dynamics (PBD)**. Let's try that!

Good news: PBD is very stable!

It has not problems to simulate a stack of objects, like this...



...many mathematically more rigorous methods would lead to jittering and a colapsing stack!

Video of PBD

(embedding videos in LaTeX is annoying)

Position-based Dynamics (PBD) **is not a convergent** method.

Bad news...

Position-based Dynamics (PBD) **is not a convergent** method.

The End.

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

1. **numerical error $\ll$ model error**
   - Visually identical to other numerical methods.

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

1. **numerical error $\ll$ model error**
   - Visually identical to other numerical methods.
2. **Speed:** Our simulations are almost 100 times faster now.

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

1. **numerical error $\ll$ model error**
   - Visually identical to other numerical methods.
2. **Speed:** Our simulations are almost 100 times faster now.
3. **Stability:** No cases anymore in which an internal solver fails.

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

1. **numerical error $\ll$ model error**
   - Visually identical to other numerical methods.
2. **Speed:** Our simulations are almost 100 times faster now.
3. **Stability:** No cases anymore in which an internal solver fails.
4. **Simplicity:** It is super easy to implement (even for PDEs).

Bad news?

Position-based Dynamics **is probably not a convergent** method, but **very stable and fast**.

For our application

1. **numerical error $\ll$ model error**
   - Visually identical to other numerical methods.
2. **Speed:** Our simulations are almost 100 times faster now.
3. **Stability:** No cases anymore in which an internal solver fails.
4. **Simplicity:** It is super easy to implement (even for PDEs).

Goal (work in progress):
Find rigorous mathematical arguments to justify use of Position-based Dynamics (PBD).

Overview of this talk

1. Position-based Dynamics for first order systems,
2. Filippov ODEs and numerical integration,
3. ...attempts to get error bounds.

Position-based Dynamics for first order systems

The toy model: First order hard-sphere model

We consider N particles (in 2D) with radius $R = 1$ and with positions $\boldsymbol{X} = (X_1, \ldots, X_N) \in \mathbb{R}^{2N}$.

[1] $k = 1, \ldots, m$ corresponds to all pairs $\{1, 2\}, \{1, 3\}, \ldots, \{N-1, N\}$.

We consider N particles (in 2D) with radius $R = 1$ and with positions $\boldsymbol{X} = (X_1, \ldots, X_N) \in \mathbb{R}^{2N}$. We consider this **complementarity system**

$$\begin{cases} \dot{X}_i = f_i(\boldsymbol{X}) + \sum_{k=1}^{m} \lambda_k \nabla g_k(\boldsymbol{X}) & \text{for all } i = 1, \ldots, N, \\ g_k \geq 0, \quad \lambda_k \geq 0 \quad \text{and} \quad g_k \lambda_k = 0 & \text{for all } k = 1, \ldots, m, \\ X_i(0) = X_i^{\text{init}} & \text{for all } i = 1, \ldots, N \end{cases}$$

where

$$g_k(\boldsymbol{X}) := \|X_i - X_j\| - 2$$

are the $m = \binom{2}{N}$ constraints for non-overlapping spheres. [1]

[1] $k = 1, \ldots, m$ corresponds to all pairs $\{1, 2\}, \{1, 3\}, \ldots, \{N-1, N\}$.

Incredients of Position-based Dynamics

1. **Explicit Euler:** Numerical flow map

$$\Phi_h^f(\boldsymbol{X}) = \boldsymbol{x} + h\boldsymbol{f}(\boldsymbol{X})$$

Incredients of Position-based Dynamics

1. **Explicit Euler:** Numerical flow map

$$\Phi_h^f(\boldsymbol{X}) = \boldsymbol{x} + h\boldsymbol{f}(\boldsymbol{X})$$

2. **Proximal maps:** For a given constraint g_k, the proximal operator is

$$\mathrm{prox}^{g_k}(x) = \operatorname*{argmin}_{y \in \mathbb{R}^{2N}, g_k(y) \geq 0} \|x - y\|$$

Ingredients of Position-based Dynamics

1. **Explicit Euler:** Numerical flow map

$$\Phi_h^f(\boldsymbol{X}) = \boldsymbol{x} + h\boldsymbol{f}(\boldsymbol{X})$$

2. **Proximal maps:** For a given constraint g_k, the proximal operator is

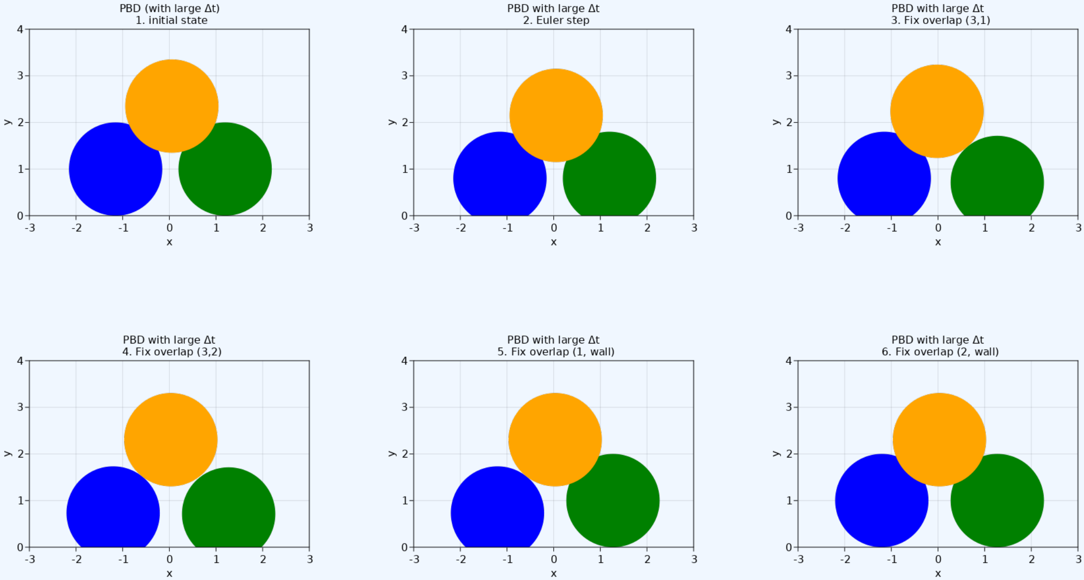
$$\text{prox}^{g_k}(x) = \underset{y \in \mathbb{R}^{2N}, g_k(y) \geq 0}{\text{argmin}} \|x - y\|$$

Numerical flow map of PBD

$$\Phi_h^{\text{PBD}}(\boldsymbol{X}) = \text{prox}^{g_m} \circ \cdots \circ \text{prox}^{g_1} \circ \Phi_h^f(\boldsymbol{X})$$

Hence, numerical solution is

$$\boldsymbol{X}^{n+1} = \Phi_h^{\text{PBD}}(\boldsymbol{X}^n)$$


PBD (with large Δt)
1. initial state
PBD with large Δt
2. Euler step
PBD with large Δt
3. Fix overlap (3,1)
PBD with large Δt
4. Fix overlap (3,2)
PBD with large Δt
5. Fix overlap (1, wall)
PBD with large Δt
6. Fix overlap (2, wall)
x
y
-3
-2
-1
0
1
2
3
4

Computational budget

Computation of $\Phi_h^{\text{PBD}}(\boldsymbol{X})$ is very fast,

Computational budget

Computation of $\Phi_h^{\mathrm{PBD}}(\boldsymbol{X})$ is very fast,

$\Rightarrow$ we can choose very small step-size h,

Computational budget

Computation of $\Phi_h^{\text{PBD}}(\boldsymbol{X})$ is very fast,

$\Rightarrow$ we can choose very small step-size h,

$\Rightarrow$ within each step the constraint violation remains small.

Computational budget

Computation of $\Phi_h^{\mathrm{PBD}}(\boldsymbol{X})$ is very fast,

$\Rightarrow$ we can choose very small step-size h,

$\Rightarrow$ within each step the constraint violation remains small.

But, we have no **error bounds** for the accuracy of the Lagrangian multipliers!

Computational budget

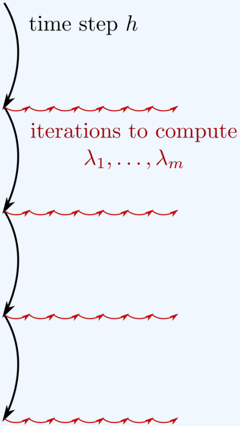
Computation of $\Phi_h^{PBD}(\boldsymbol{X})$ is very fast,

$\Rightarrow$ we can choose very small step-size h,

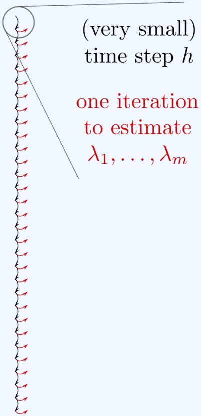
$\Rightarrow$ within each step the constraint violation remains small.

But, we have no **error bounds** for the accuracy of the Lagrangian multipliers!

Time-stepping with internal LCP solver



Position-based Dynamics



Filippov ODEs and numerical integration

Consider $\dot{x} = f(x)$.

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerics 101

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

- **Consistency:**
  $\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2$.

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

- **Consistency:**
  $\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2$.
- **Stability:**
  $\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + Lh)\|x - y\|$.

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

- **Consistency:**
  $\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2$.
- **Stability:**
  $\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + Lh)\|x - y\|$.
- **Convergence:** For fixed $T > 0$,

$$\|\varphi_{nh}(x) - \Phi_h^n(x)\| \leq Mh \quad \text{for all } n, h \text{ with } nh < T.$$

Numerics 101

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

- **Consistency:**
  $\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2$.
- **Stability:**
  $\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + Lh)\|x - y\|$.
- **Convergence:** For fixed $T > 0$,

  $\|\varphi_{nh}(x) - \Phi_h^n(x)\| \leq Mh \quad$ for all n, h with $nh < T$.

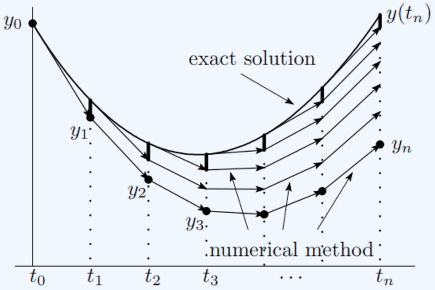


Figure 3: Lady Windermere's fan.

Consider $\dot{x} = f(x)$.

Exact flow:
$\varphi_t(x) = e^t x$.

Numerical flow (explicit Euler):
$\Phi_h(x) = x + hf(x)$.

- **Consistency:**
  $\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2$.
- **Stability:**
  $\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + Lh)\|x - y\|$.
- **Convergence:** For fixed $T > 0$,

  $\|\varphi_{nh}(x) - \Phi_h^n(x)\| \leq Mh \quad$ for all n, h with $nh < T$.

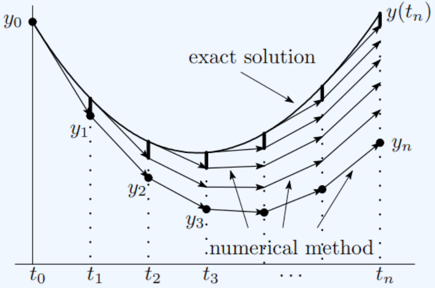


Figure 3: Lady Windermere's fan.

A typical result is **consistency + stability ⇒ convergence.**

In which sense do exact solutions even exists?

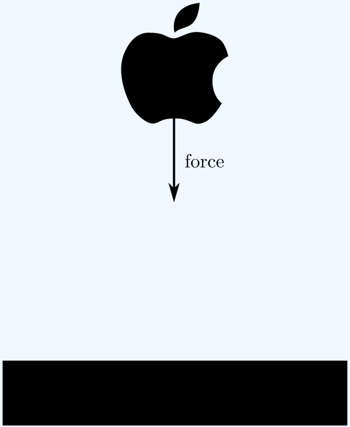
Example:

$$\begin{aligned} \dot{y} &= -1 + \lambda, \\ g(y) &= y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0. \end{aligned}$$

Example:

$$\dot{y} = -1 + \lambda,$$
$$g(y) = y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0.$$

(Think of y as the height of the apple over the ground.)

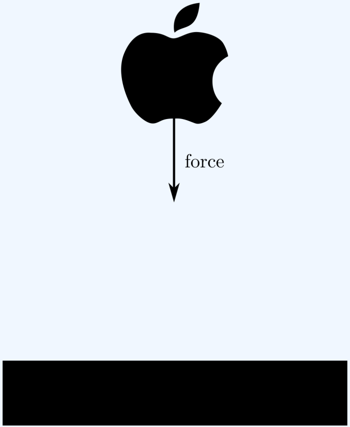


Example:

$$\dot{y} = -1 + \lambda,$$
$$g(y) = y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0.$$

(Think of y as the height of the apple over the ground.) Here:

$$\lambda = \begin{cases} 0 & \text{before impact,} \\ 1 & \text{after impact.} \end{cases}$$



On the ground, the complementary condition implies (if $\dot{y}$ exists):

$$\begin{aligned} 0 &= \dot{y}\lambda + y\dot{\lambda} \\ &= (-1 + \lambda)\lambda. \end{aligned}$$

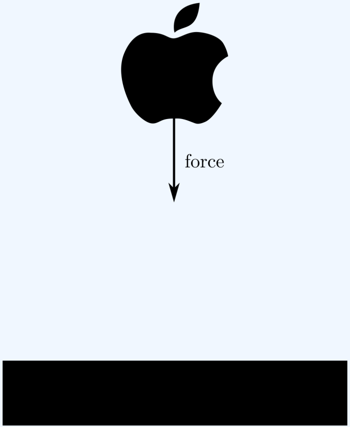
Discontinious right-hand sides

Example:

$$\dot{y} = -1 + \lambda,$$
$$g(y) = y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0.$$

(Think of y as the height of the apple over the ground.) Here:

$$\lambda = \begin{cases} 0 & \text{before impact,} \\ 1 & \text{after impact.} \end{cases}$$



...or as Filippov ODE:

$$\dot{y} \in \begin{cases} \{-1\} & y > 0 \\ [-1, 0] & y = 0 \\ \{0\} & y < 0 \end{cases}$$

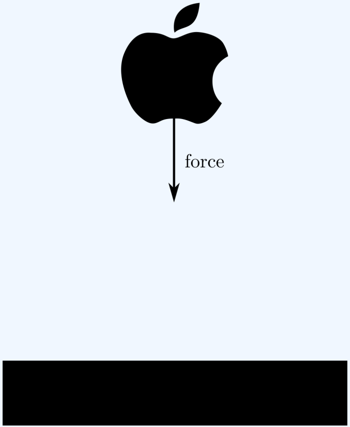
Discontinious right-hand sides

Example:

$$\dot{y} = -1 + \lambda,$$
$$g(y) = y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0.$$

(Think of y as the height of the apple over the ground.) Here:

$$\lambda = \begin{cases} 0 & \text{before impact,} \\ 1 & \text{after impact.} \end{cases}$$



...or as Filippov ODE:

$$\dot{y} \in \begin{cases} \{-1\} & y > 0 \\ [-1, 0] & y = 0 \\ \{0\} & y < 0 \end{cases}$$

- Existence theory,
- allows extension of ODE to infeasible positions.

Does numerical integration work for such systems?

Example: Sliding case

$$f^+ := \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad f^- := \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\dot{x} \in \begin{cases} f^+ & x_2 > 0 \\ \overline{\text{co}}(\{f^+, f^-\}) & x_2 = 0 \\ f^- & x_2 < 0 \end{cases}$$

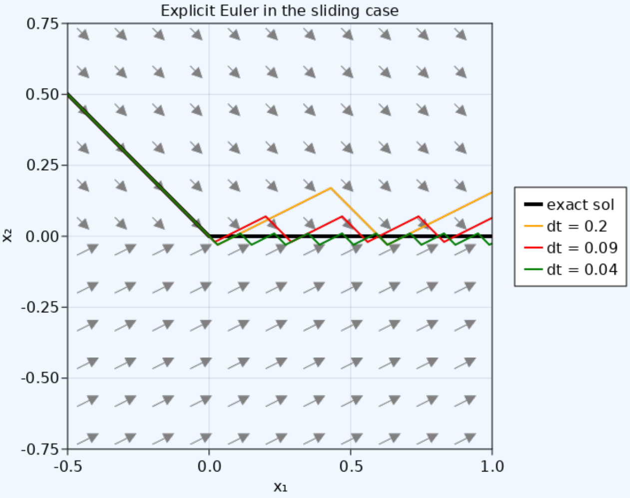
$\overline{\text{co}}\{\dots\}$ is the closure of the convex hull.

Example: Sliding case

$$f^+ := \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad f^- := \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

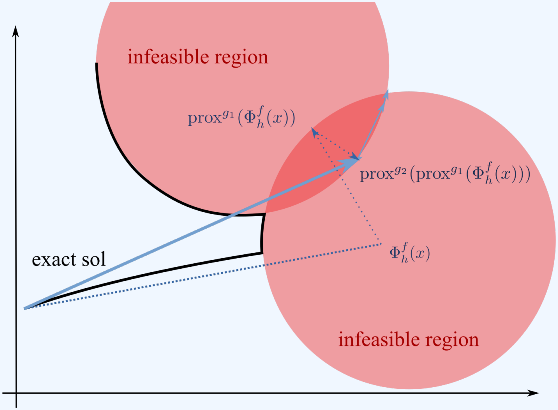
$$\dot{x} \in \begin{cases} f^+ & x_2 > 0 \\ \overline{\text{co}}(\{f^+, f^-\}) & x_2 = 0 \\ f^- & x_2 < 0 \end{cases}$$

$\overline{\text{co}}\{\dots\}$ is the closure of the convex hull.

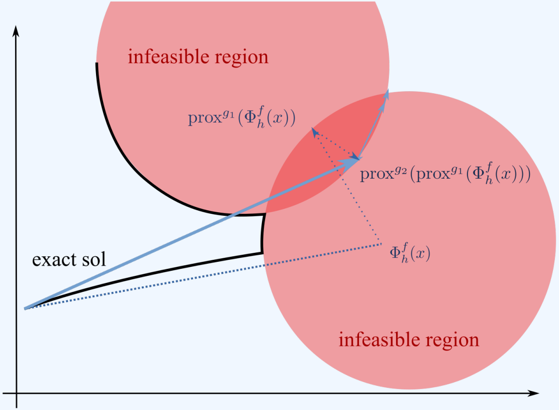


…towards error bounds for PBD

Challenge in the numerical analysis

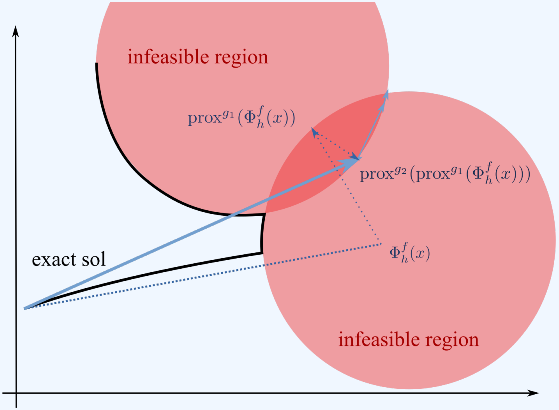


- How fast do we enter the infeasible regions?



- How fast do we enter the infeasible regions?
- What are the chain reactions of

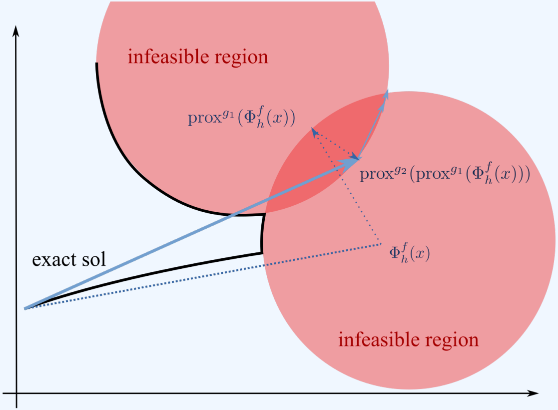
  $P(X) := \text{prox}^{g_m} \circ \cdots \circ \text{prox}^{g_1}(X).$



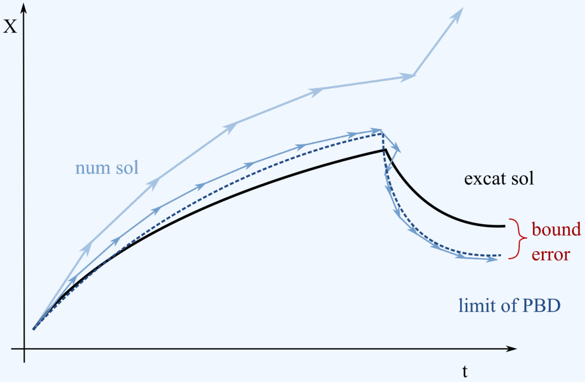
- How fast do we enter the infeasible regions?
- What are the chain reactions of

  $$P(X) := \mathrm{prox}^{g_m} \circ \cdots \circ \mathrm{prox}^{g_1}(X).$$

- How likely are bad cases?



- I want to find a global error bound.



To analyse

$$P(\boldsymbol{X}) \coloneqq \mathrm{prox}^{g_m} \circ \cdots \circ \mathrm{prox}^{g_1}(\boldsymbol{X})$$

To analyse

$$P(\boldsymbol{X}) \coloneqq \operatorname{prox}^{g_m} \circ \cdots \circ \operatorname{prox}^{g_1}(\boldsymbol{X})$$

we consider the graph

$$\begin{aligned} G &= (V, E) \quad \text{with} \\ V &= \{1, \dots, N\}, \\ E &= \{(i, j) \mid \text{if } \|X_i - X_j\| < 2R\}. \end{aligned}$$

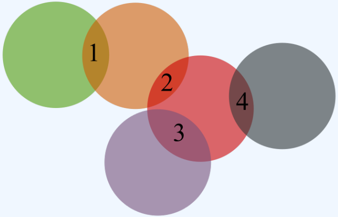
To analyse

$$P(\boldsymbol{X}) := \mathrm{prox}^{g_m} \circ \cdots \circ \mathrm{prox}^{g_1}(\boldsymbol{X})$$

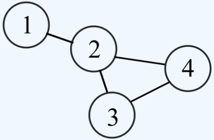
we consider the graph

$$\begin{aligned} G &= (V, E) \quad \text{with} \\ V &= \{1, \dots, N\}, \\ E &= \{(i,j) \mid \text{if } \|X_i - X_j\| < 2R\}. \end{aligned}$$

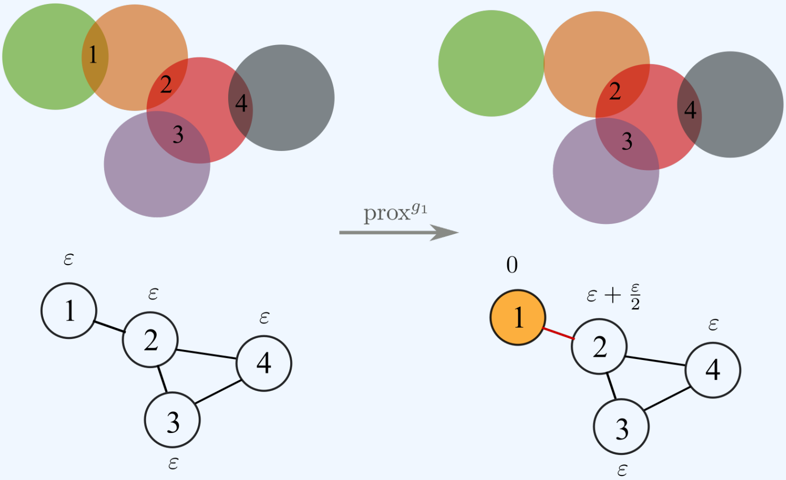
enumerated contacts



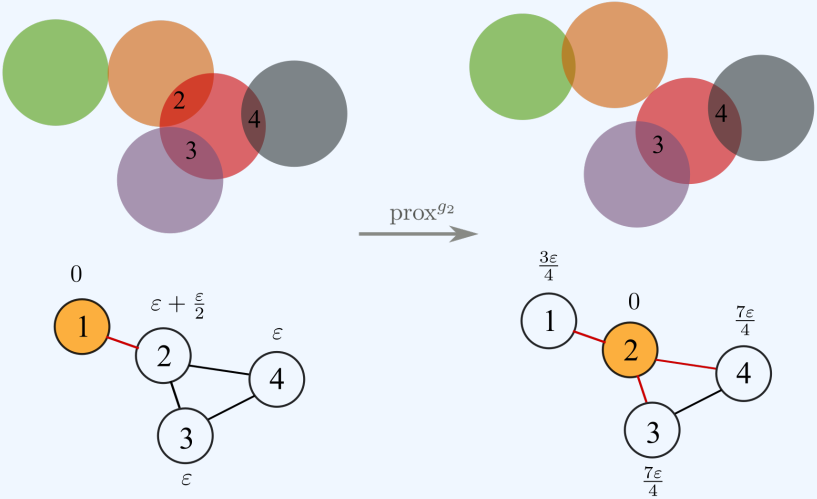
dual of unit disc graph



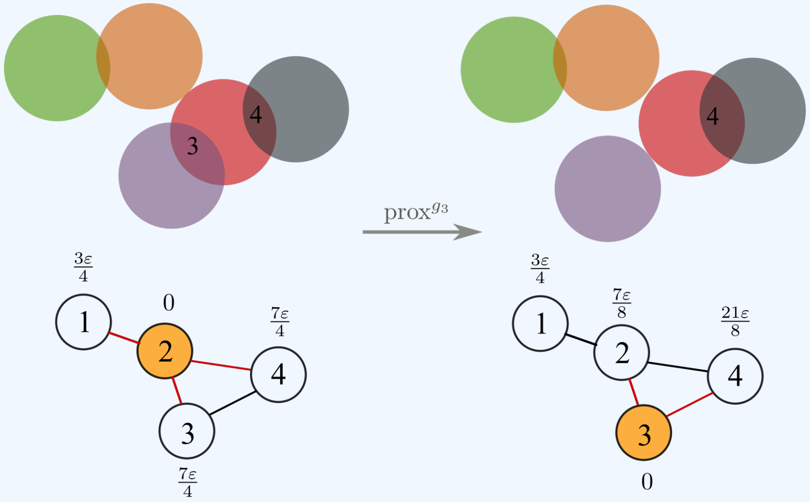
Worst-case violation of constraints



Worst-case violation of constraints



Worst-case violation of constraints



Worst-case bounds

Lemma

Given a state $\boldsymbol{X} \in \mathbb{R}^{2N}$ such that

$$g_k(\boldsymbol{X}) \geq 0 - \varepsilon \quad \text{for all } k$$

then

$$g_k(P(\boldsymbol{X})) \geq 0 - C\varepsilon \quad \text{for all } k$$

where the constant C depends on properties of the unit disk graph.

Lemma

Given a state $\boldsymbol{X} \in \mathbb{R}^{2N}$ such that

$$g_k(\boldsymbol{X}) \geq 0 - \frac{R}{4} \quad \text{for all } k$$

$$\sum_k \max(-g_k(\boldsymbol{X}), 0) \geq C \sum_k \max(-g_k(P(\boldsymbol{X})), 0)$$

where the constant C depends on properties of the unit disk graph.

(But I have no satisfying bound for C yet.)

Outlook

- **I cannot prove consistency (yet):**

$$\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2.$$

Outlook

- **I cannot prove consistency (yet):**

$$\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2.$$

- **Proving stability in this sense is possible:**

$$\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + \tilde{C}Lh)\|x - y\|.$$

Outlook

- **I cannot prove consistency (yet):**

$$\|\varphi_h(x) - \Phi_h(x)\| \leq Ch^2.$$

- **Proving stability in this sense is possible:**

$$\|\Phi_h(x) - \Phi_h(y)\| \leq (1 + \tilde{C}Lh)\|x - y\|.$$

- **Maybe I can only get this kind of convergence:** For fixed $T > 0$,

$$\|\varphi_{nh}(x) - \Phi_h^n(x)\| \leq \mathbf{C} + Mh \quad \text{for all } n, h \text{ with } nh < T.$$

References I

A. F. Filippov.
Differential Equations with Discontinuous Righthand Sides, volume 18 of *Mathematics and Its Applications*.
Springer Netherlands, Dordrecht, 1988.

E. Hairer, S. P. Nørsett, and Gerhard Wanner.
Solving Ordinary Differential Equations I: Nonstiff Problems.
Number 8 in Springer Series in Computational Mathematics. Springer, Heidelberg ; London, 2nd rev. ed edition, 2009.

Remco I. Leine and Henk Nijmeijer.
Dynamics and Bifurcations of Non-Smooth Mechanical Systems, volume 18 of *Lecture Notes in Applied and Computational Mechanics*.
Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

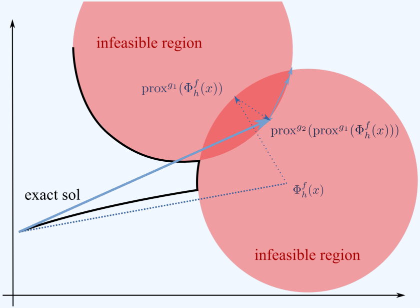
Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller.
Small steps in physics simulation.
In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '19*, pages 1–7, Los Angeles, California, 2019. ACM Press.

References II

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff.
Position based dynamics.
Journal of Visual Communication and Image Representation, 18(2):109–118, April 2007.

Peter Wriggers.
Computational Contact Mechanics.
Springer-Verlag, Berlin Heidelberg, second edition, 2006.

Thank you for your attention!



Alternatives

- **Non-smooth contact dynamics:**
  Solve a nonlinear optimisation problem in each time-step...
- **Smoothening, Repulsive potentials, Penalty method, Discrete Element method, ...**
  Replace non-smooth right-hand side with a smooth approximation or use alternative model.
  $\longrightarrow$ Might be more physical, but also leads to very stiff systems.
- **Implicit methods:**
  Use large time-steps but a nonlinear solve which usually also predicts the collision response.
- **Event time methods:**
  Predict time of collision and compute correct response exactly.

It is very hard to be faster and simpler than PBD, but these methods above are more rigorous and backed by decades of experience.

Discontinious right-hand sides

Example:

$$\dot{y} = -1 + \lambda,$$
$$g(y) = y \geq 0, \quad \lambda \geq 0, \quad y\lambda = 0.$$

Consider a state $y(t^*) = 0$.

Then, the complementary condition implies (if $\dot{y}$ exists):

$$\begin{aligned} 0 &= \dot{y}\lambda + y\dot{\lambda} \\ &= (-1 + \lambda)\lambda. \end{aligned}$$

Hence,

$$\lambda(t^*) = 1.$$