CONSTRAINT-BASED SIMULATION

Jan Bender

RWTH Aachen University

Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

Motivation

- Real-time physically-based simulation in visual computing:
  - virtual reality
  - training simulators
  - robotics
  - computer games
  - medical simulation
  - …
- Requirements:
  - high performance
  - stability
  - controllability
  - realistic results (however, high accuracy is often secondary)

Motivation

Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

Constraints

- The motion of a body can be restricted by constraints of the form

$$\mathbf{C}(\mathbf{x}) = 0,$$

  where $\mathbf{x}$ contains all particle positions.
- PBD enforces constraints by position changes.

Constraints - Example

- Distance constraint:

$$\mathbf{C}(\mathbf{x}) = \|\mathbf{x}_1 - \mathbf{x}_2\| - l_0 = 0$$

- Position changes:

PBD-Algorithm

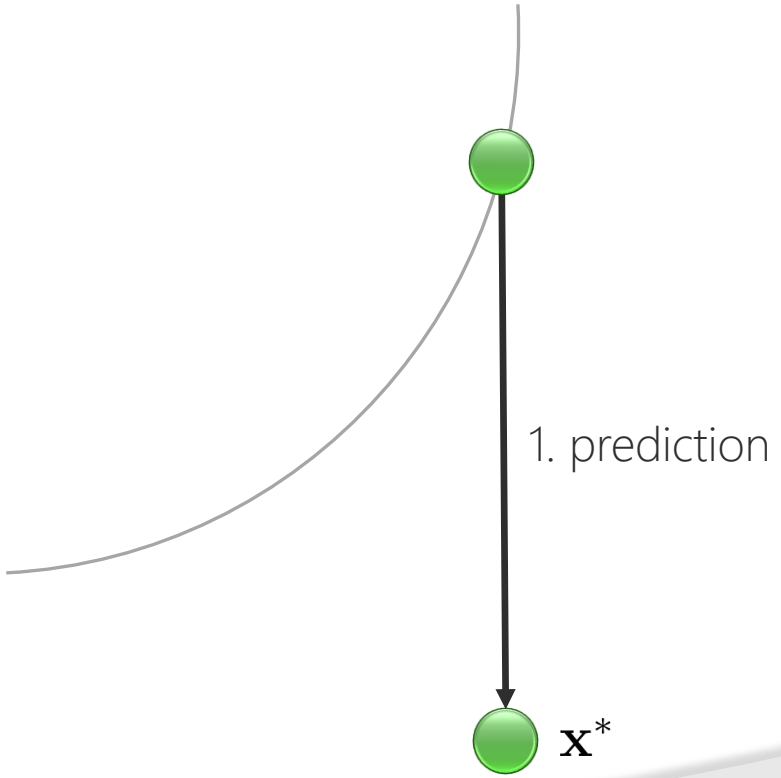
1. Prediction step:
   - Time integration only considering external forces to get candidate positions $\mathbf{x}^*$.
2. Constraint projection:
   - Check if constraints are fulfilled, otherwise modify candidate positions:

$$\mathbf{x}(t + \Delta t) = \text{constraintProjection}\left(C(\mathbf{x}), \mathbf{x}^*\right)$$

3. Velocity update:
   - Update velocities according to position changes to get a consistent system.

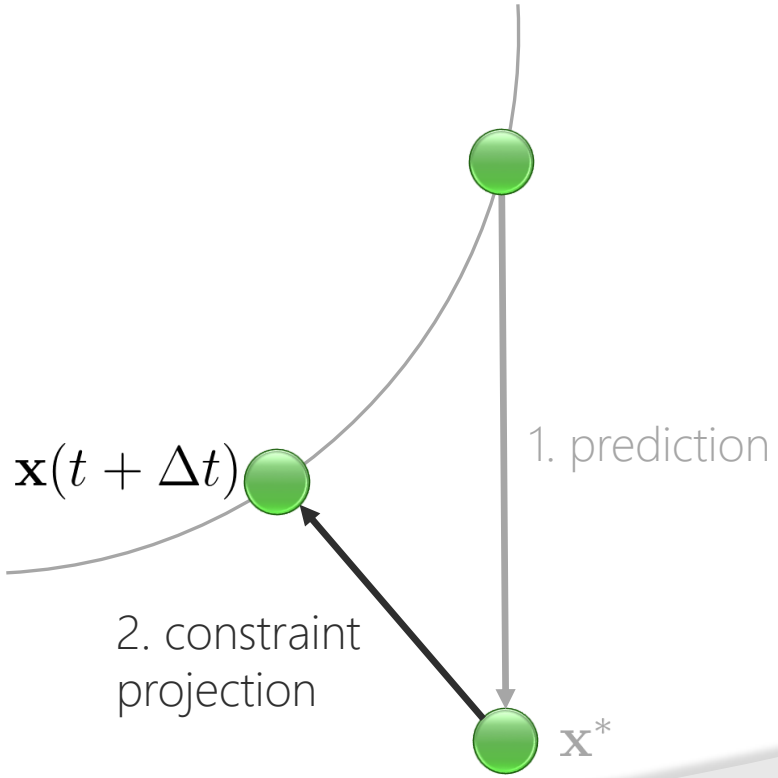
Example

- Position correction for a particle on a circle



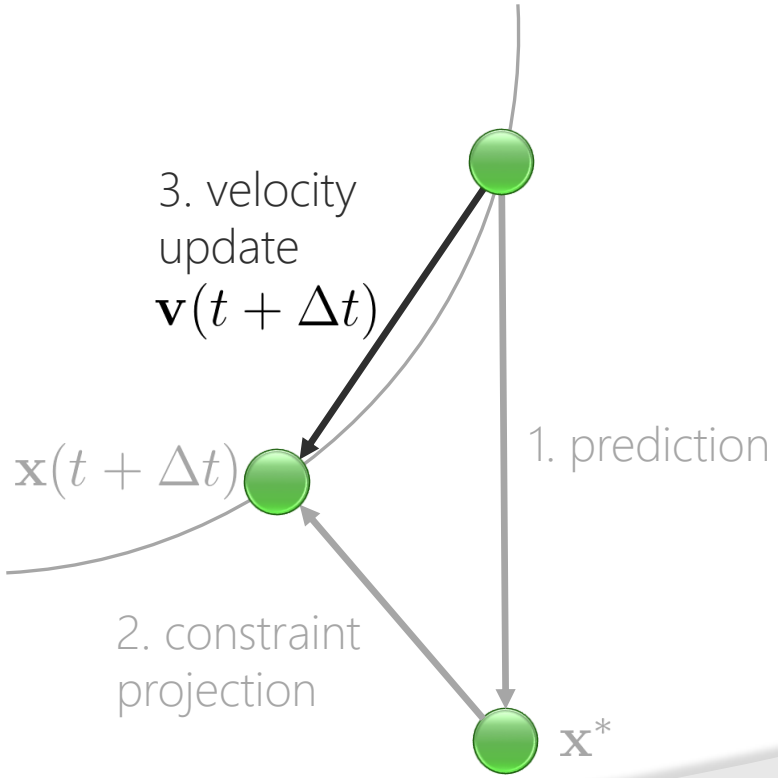
Example

- Position correction for a particle on a circle



Example

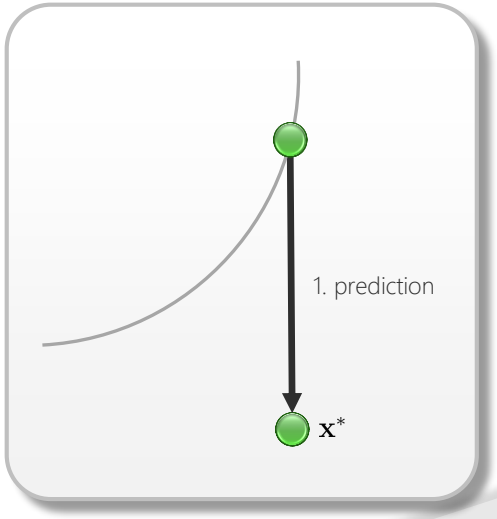
- Position correction for a particle on a circle



1. Prediction Step

- Symplectic Euler integration to get candidate positions:

$$\mathbf{v}^* = \mathbf{v}^n + \frac{\Delta t}{m}\mathbf{F}^{\text{ext}}$$
$$\mathbf{x}^* = \mathbf{x}^n + \Delta t \mathbf{v}^*$$



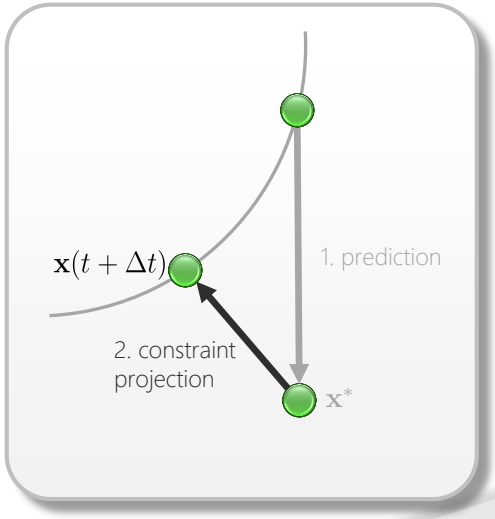
2. Constraint Projection

- Linearize the constraint by a first-order Taylor approximation:

$$\mathbf{C}(\mathbf{x}^* + \Delta\mathbf{x}) \approx \mathbf{C}(\mathbf{x}^*) + \mathbf{J}\Delta\mathbf{x} = 0$$

$$\mathbf{J} = \nabla\mathbf{C}^T$$

- However, the resulting linear system is underdetermined.



Constraint Projection

- Solution: we restrict position correction to be in the direction of the gradient (constraint space):

$$\Delta \mathbf{x} = \mathbf{J}^T \boldsymbol{\lambda}$$

- This guarantees momentum conservation.
- $\boldsymbol{\lambda}$ is a Lagrange multiplier which defines the position correction in constraint space.

Constraint Projection

- Finally, we weight the corrections with the inverse particle masses:

$$\Delta \mathbf{x} = \mathbf{M}^{-1}\mathbf{J}^{T}\boldsymbol{\lambda}$$

Constraint Projection

- Substituting this position correction

$$\Delta \mathbf{x} = \mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda}$$

in

$$\mathbf{C}(\mathbf{x}^*) + \mathbf{J}\Delta\mathbf{x} = 0$$

yields

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} = -\mathbf{C}(\mathbf{x}^*)$$

which is the final system of linear equations that has to be solved.

Interpretation

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} = -\mathbf{C}(\mathbf{x}^*)$$

- Mass-weighted position correction (CS): $\boldsymbol{\lambda}$
- Mass-weighted position correction (WS): $\mathbf{J}^T\boldsymbol{\lambda}$
- Position correction (WS): $\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda}$
- Position correction (CS): $\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda}$
- Predicted constraint error (CS): $-\mathbf{C}(\mathbf{x}^*)$

WS = world space, CS = constraint space

Example: Distance Constraint

- Constraint function:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\| - l_0$$

- Gradients:

$$\frac{\partial}{\partial \mathbf{x}_1} C(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$$

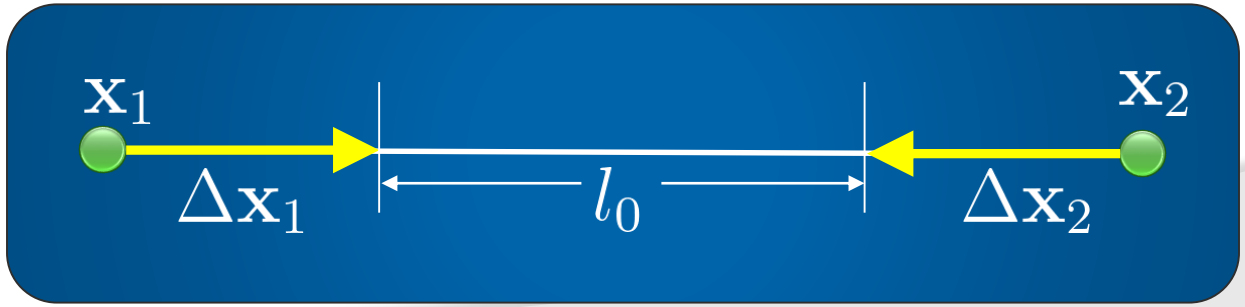
$$\frac{\partial}{\partial \mathbf{x}_2} C(\mathbf{x}_1, \mathbf{x}_2) = -\frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$$

Example: Distance Constraint

- This yields the following position corrections:

$$\Delta \mathbf{x}_1 = -\frac{\frac{1}{m_1}}{\frac{1}{m_1}+\frac{1}{m_2}}(\|\mathbf{x}_1-\mathbf{x}_2\|-l_0)\frac{\mathbf{x}_1-\mathbf{x}_2}{\|\mathbf{x}_1-\mathbf{x}_2\|}$$

$$\Delta \mathbf{x}_2 = +\frac{\frac{1}{m_2}}{\frac{1}{m_1}+\frac{1}{m_2}}(\|\mathbf{x}_1-\mathbf{x}_2\|-l_0)\frac{\mathbf{x}_1-\mathbf{x}_2}{\|\mathbf{x}_1-\mathbf{x}_2\|}$$



Constraint Solver

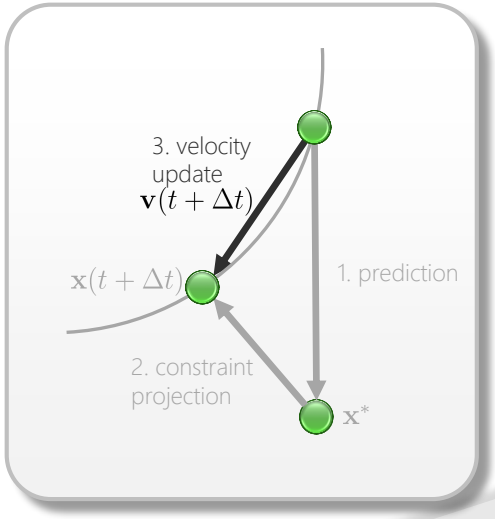
- Modified Jacobi iteration
  - In each iteration solve all constraints in parallel
  - Accumulate position corrections
  - Average corrections after each iteration

- Gauss-Seidel iteration
  - In each iteration solve constraints one by one
  - Better convergence
  - Atomic operations required for parallelization

3. Velocity Update

- Update velocities to get a consistent system as:

$$\mathbf{v}(t+\Delta t) = \frac{1}{\Delta t}(\mathbf{x}(t+\Delta t) - \mathbf{x}(t))$$



Discussion

- Advantages:
  - Fast
  - Stable
  - Simple to implement
- Disadvantages
  - Material stiffness depends on number of iterations and time step size

Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

eXtended Position-Based Dynamics

- XPBD solves the stiffness problem by using a compliant constraint formulation to minimize the energy:

$$E(\mathbf{x}) = \tfrac{1}{2}\mathbf{C}(\mathbf{x})^T \boldsymbol{\alpha}^{-1}\mathbf{C}(\mathbf{x})$$

where compliance matrix $\boldsymbol{\alpha}$ corresponds to the inverse stiffness.

PBD - XPBD

- Position Based Dynamics

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} = -C(\mathbf{x}^n)$$

$$\Delta\mathbf{x} = \mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda}$$

- eXtended Position-Based Dynamics (XPBD)

$$\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T + \frac{\boldsymbol{\alpha}}{\Delta t^2}\right)\Delta\boldsymbol{\lambda} = -C(\mathbf{x}^n) - \frac{\boldsymbol{\alpha}}{\Delta t^2}\boldsymbol{\lambda}^n$$

$$\Delta\mathbf{x} = \mathbf{M}^{-1}\mathbf{J}^T\Delta\boldsymbol{\lambda}$$

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^n + \Delta\boldsymbol{\lambda}$$

Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

PBD Energy Minimization

- To minimize an energy, we define the constraint:

$$C(\mathbf{x}) = E(\mathbf{x})$$

- For a continuous model of a deformable solid, we use the strain energy.
- In the following we will derive the constraint function and its gradient.

Deformation

- Deformation

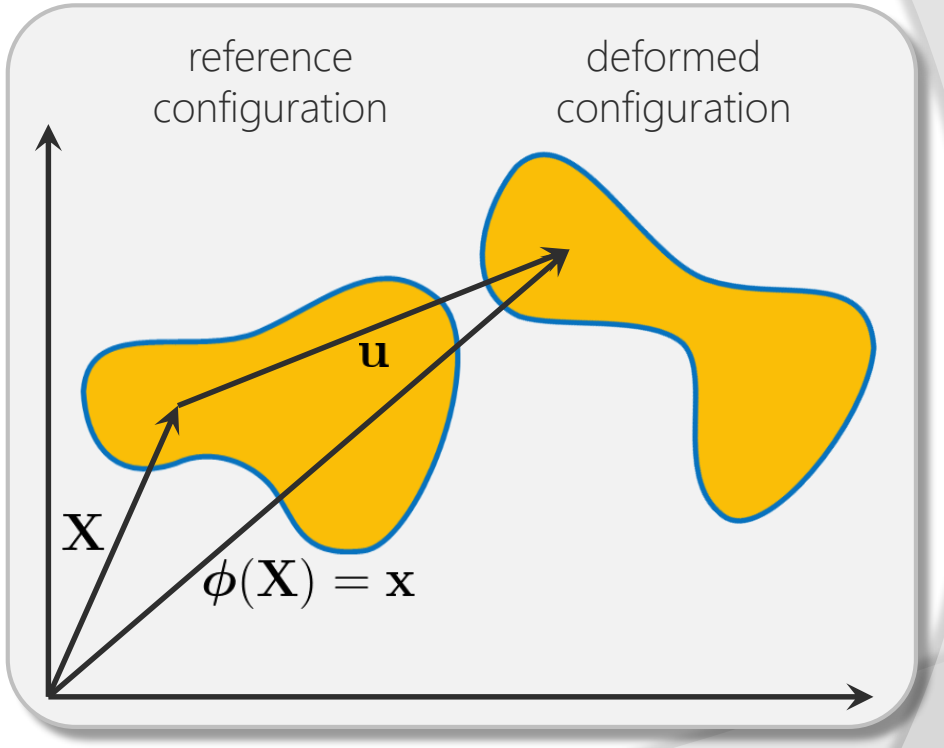
$$\boldsymbol{\phi}(\mathbf{X}) = \mathbf{X} + \mathbf{u} = \mathbf{x}$$

- Deformation gradient

$$\mathbf{F} = \partial \boldsymbol{\phi}(\mathbf{X}) / \partial \mathbf{X}$$

- Non-linear Green strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbb{1}\right)$$



Strain Energy

- Energy:

$$E = \int_{\Omega} \Psi d\mathbf{X}$$

- Possible constitutive models of materials:
  - Hooke's generalized law $\Psi = \frac{1}{2}\boldsymbol{\epsilon} : \mathbf{C}\boldsymbol{\epsilon}$
  - St. Venant-Kirchhoff model $\Psi = \frac{\lambda}{2}\left[\mathrm{tr}(\boldsymbol{\epsilon})\right]^2 + \mu \mathrm{tr}(\boldsymbol{\epsilon} : \boldsymbol{\epsilon})$
  - Neohookean elasticity $\Psi = \frac{\mu}{2}(\mathbf{I}_1 - 3) - \mu \log(J) + \frac{\lambda}{2}\log^2(J)$

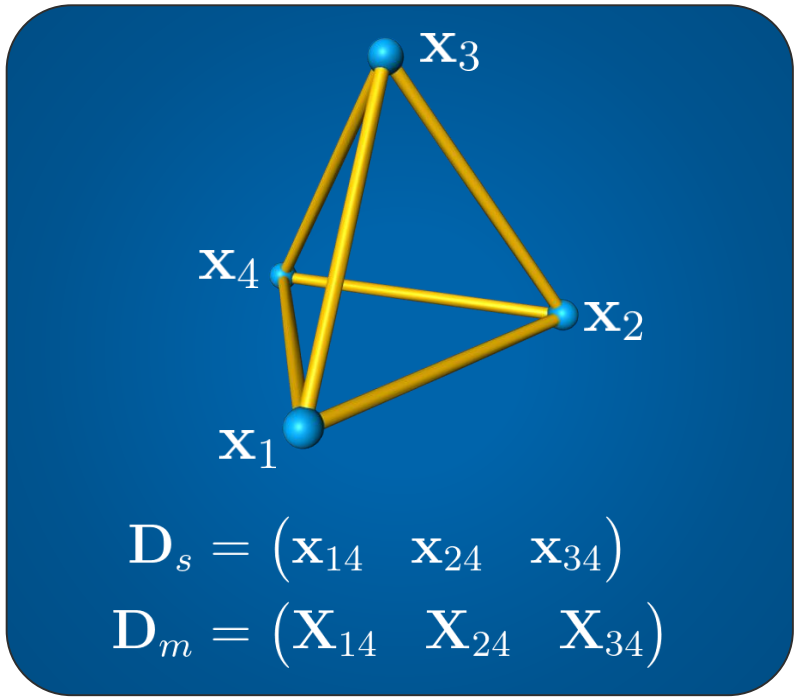
Strain Energy Gradient

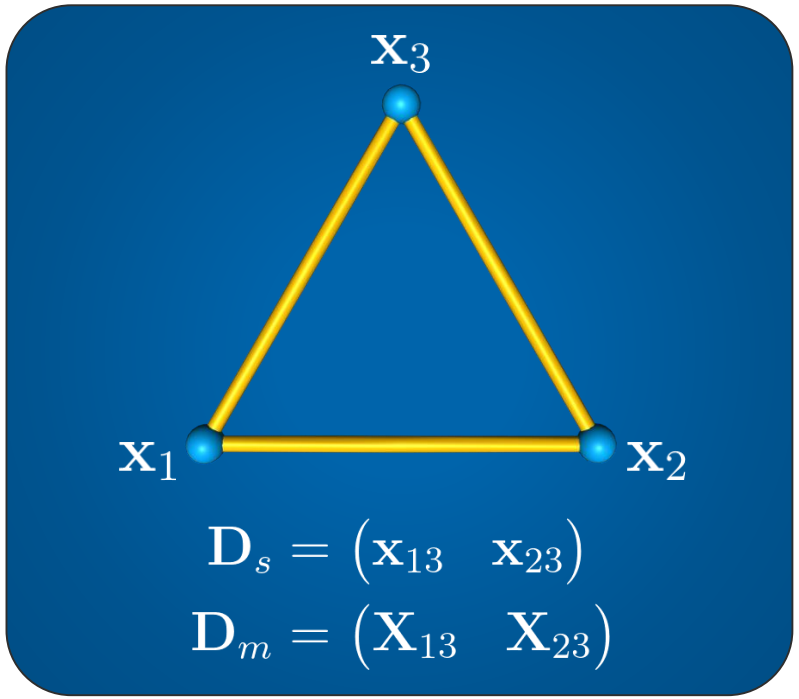
- Strain energy gradient

$$\frac{\partial E}{\partial \mathbf{x}} = \int_{\Omega} \frac{\partial \Psi}{\partial \mathbf{x}} d\mathbf{X} = \int_{\Omega} \mathbf{P}(\mathbf{F}) \frac{\partial \mathbf{F}}{\partial \mathbf{x}} d\mathbf{X}$$

where $\mathbf{P}(\mathbf{F})$ is the first Piola-Kirchhoff stress tensor.

Discretization – Linear Elements



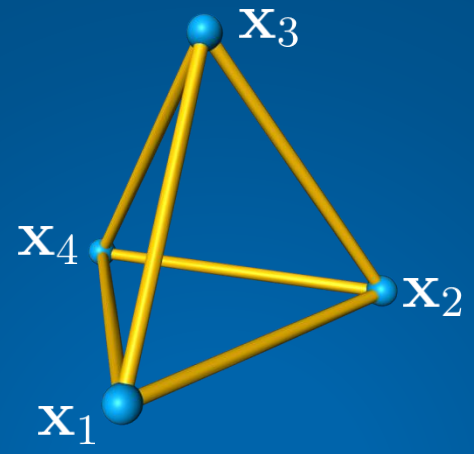


Deformation gradient:

$$\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}$$

$$\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j, \quad \mathbf{X}_{ij} = \mathbf{X}_i - \mathbf{X}_j$$

Elastic Energy

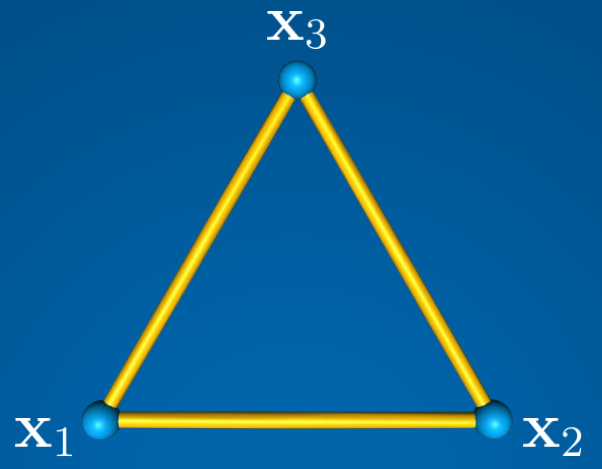


Energy: $E = V\Psi(\mathbf{F})$

Gradient:

$$\left[\frac{\partial E}{\partial \mathbf{x}_1}\ \frac{\partial E}{\partial \mathbf{x}_2}\ \frac{\partial E}{\partial \mathbf{x}_3}\right] = V\mathbf{P}(\mathbf{F})\mathbf{D}_m^{-T}$$

$$\frac{\partial E}{\partial \mathbf{x}_4} = -\sum_{i=1}^{3} \frac{\partial E}{\partial \mathbf{x}_i}$$



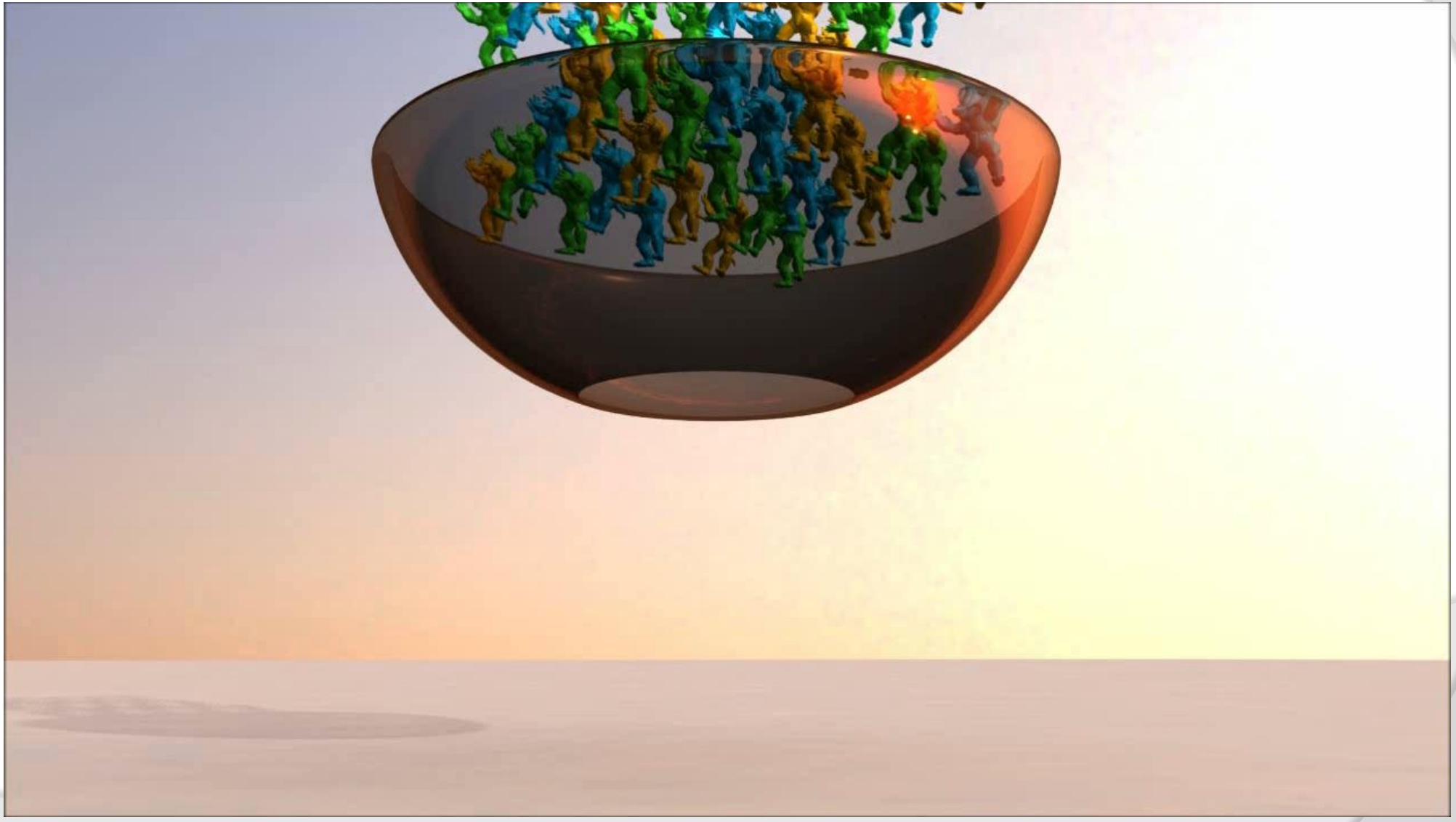
Energy: $E = A\Psi(\mathbf{F})$

Gradient:

$$\left[\frac{\partial E}{\partial \mathbf{x}_1}\ \frac{\partial E}{\partial \mathbf{x}_2}\right] = A\mathbf{P}(\mathbf{F})\mathbf{D}_m^{-T}$$

$$\frac{\partial E}{\partial \mathbf{x}_3} = -\sum_{i=1}^{2} \frac{\partial E}{\partial \mathbf{x}_i}$$

Results



Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

Prediction Step

- Translational DOFs:

$$\mathbf{v}^* = \mathbf{v}^n + \Delta t \frac{\mathbf{F}^n_{\text{ext}}}{m}$$

$$\mathbf{x}^* = \mathbf{x}^n + \Delta t \mathbf{v}^*$$

- Rotational DOFs:

$$\boldsymbol{\omega}^* = \boldsymbol{\omega}^n + \Delta t \mathbf{I}^{-1}(\boldsymbol{\tau}^n_{\text{ext}} - (\boldsymbol{\omega}^n \times (\mathbf{I}\boldsymbol{\omega}^n)))$$

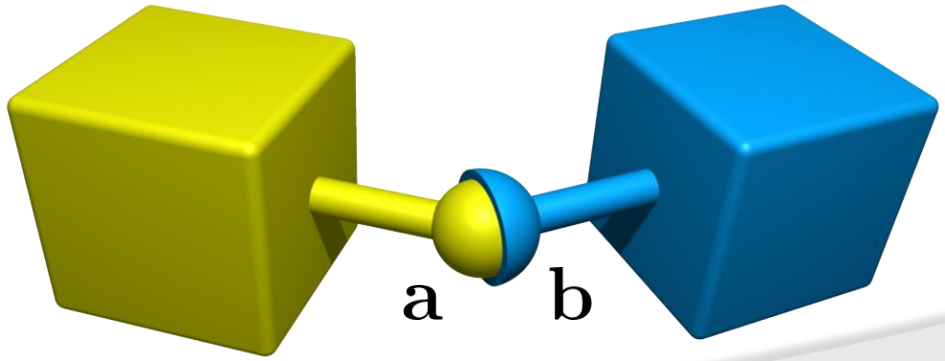
$$\mathbf{q}' = \mathbf{q}^n + \Delta t \frac{1}{2} \left[0, \omega^*_x, \omega^*_y, \omega^*_z\right] \mathbf{q}^n$$

$$\mathbf{q}^* = \frac{\mathbf{q}'}{\|\mathbf{q}'\|}$$

Position Correction

- We split up the constraint in its magnitude c and its direction $\mathbf{n}$.
- Example: ball joint

$$\mathbf{C} = \mathbf{a} - \mathbf{b} = \underbrace{\|\mathbf{a} - \mathbf{b}\|}_{c} \underbrace{\frac{\mathbf{a} - \mathbf{b}}{\|\mathbf{a} - \mathbf{b}\|}}_{\mathbf{n}} = \mathbf{0}$$

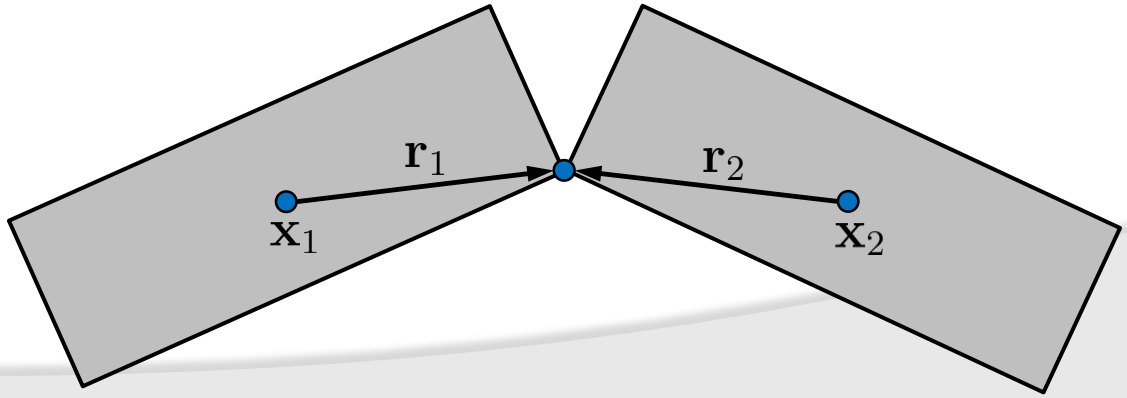


Position Correction

- To apply the mass weighted correction at the positions $\mathbf{r}_1$ and $\mathbf{r}_2$ we need the generalized inverse masses:

$$w_1 = \frac{1}{m_1} + (\mathbf{r}_1 \times \mathbf{n})^T \mathbf{I}_1^{-1} (\mathbf{r}_1 \times \mathbf{n})$$

$$w_2 = \frac{1}{m_2} + (\mathbf{r}_2 \times \mathbf{n})^T \mathbf{I}_2^{-1} (\mathbf{r}_2 \times \mathbf{n})$$



Position Correction

- The Lagrange multiplier is then determined by:

$$\lambda = \frac{-c}{w_1 + w_2}$$

- We get the mass weighted position change as:

$$\mathbf{p} = \lambda \mathbf{n}$$

Position Correction

- Finally, we apply the position change:

$$\mathbf{x}_1^* := \mathbf{x}_1^* + \frac{\mathbf{p}}{m_1}$$
$$\mathbf{x}_2^* := \mathbf{x}_2^* - \frac{\mathbf{p}}{m_2}$$
$$\mathbf{q}_1^* := \mathbf{q}_1^* + \frac{1}{2}\left(0, \mathbf{I}_1^{-1}(\mathbf{r}_1 \times \mathbf{p})\right)\mathbf{q}_1^*$$
$$\mathbf{q}_2^* := \mathbf{q}_2^* - \frac{1}{2}\left(0, \mathbf{I}_2^{-1}(\mathbf{r}_2 \times \mathbf{p})\right)\mathbf{q}_2^*$$

Velocity Update

- We update the velocities as:

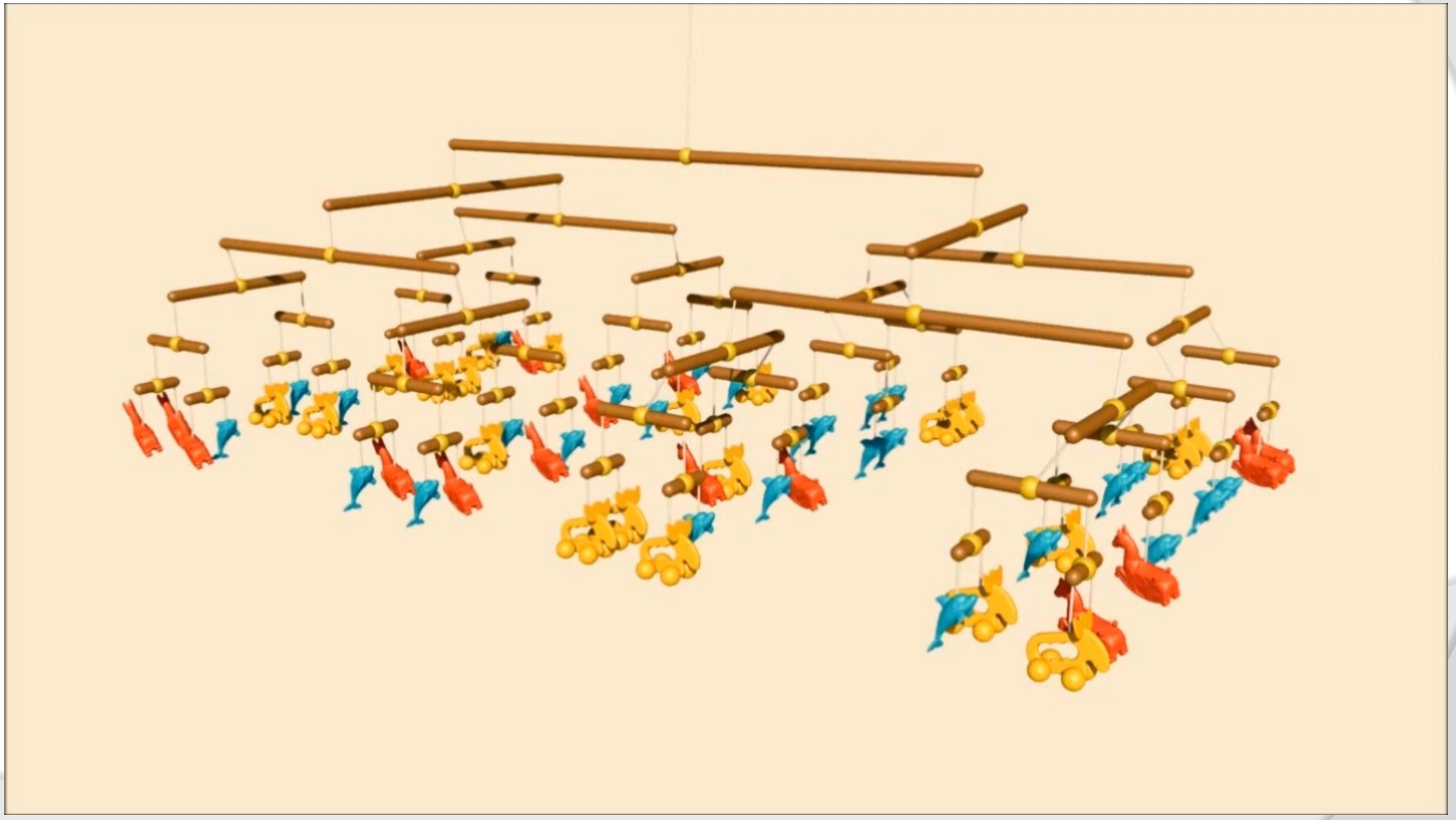
$$\mathbf{v}^{n+1} = \frac{1}{\Delta t}(\mathbf{x}^{n+1} - \mathbf{x}^{n})$$

$$\Delta\mathbf{q} = \mathbf{q}^{n+1}(\mathbf{q}^{n})^{-1}$$

$$\boldsymbol{\omega}^{n+1} = \frac{2}{\Delta t}\left(\Delta\mathbf{q}_x, \Delta\mathbf{q}_y, \Delta\mathbf{q}_z\right)$$

- To avoid a rotation flip we negate $\boldsymbol{\omega}^{n+1}$, if $\Delta\mathbf{q}_w < 0$.

Results

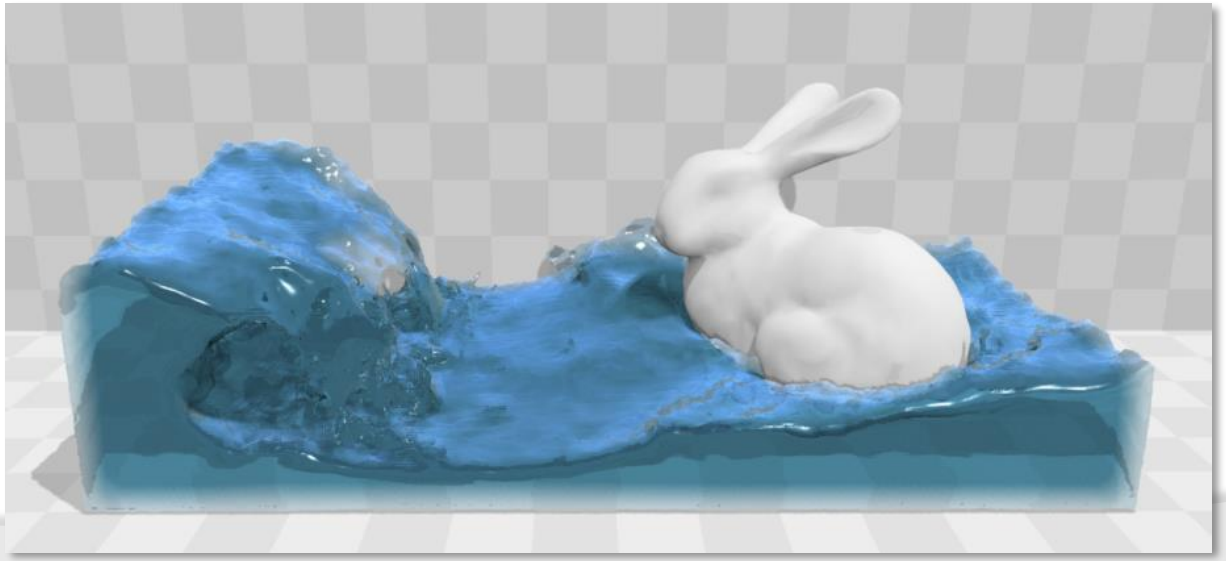


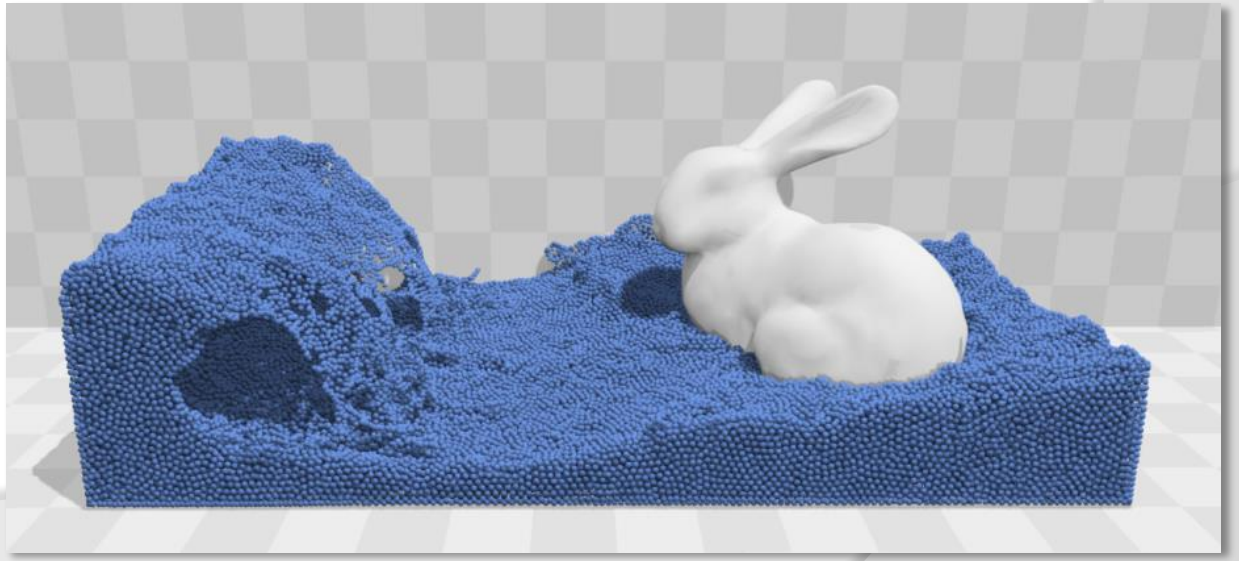
Content

- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

Position Based Fluids

- Density constraint: $C_i = \frac{\rho_i}{\rho_0} - 1 = \frac{1}{\rho_0} \sum_j m_j W_{ij} - 1 = 0$
- Gradient: $\nabla C_i = \frac{1}{\rho_0} \sum_j m_j \nabla W_{ij}$





Position Based Fluid simulation [Macklin and Müller, 2013]

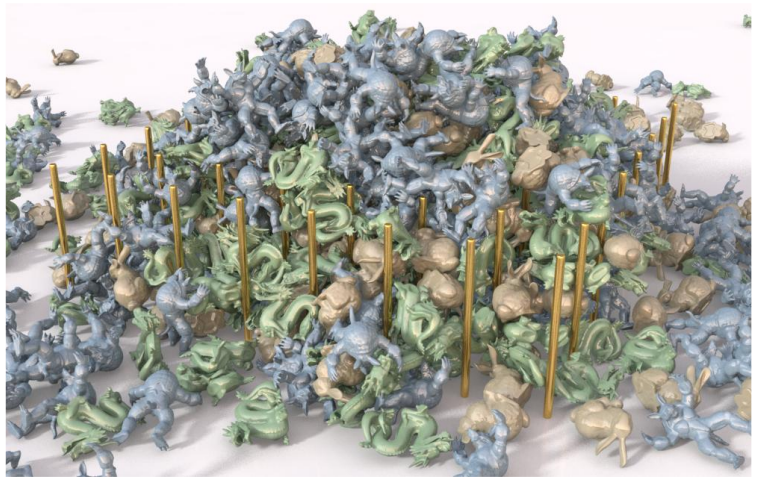
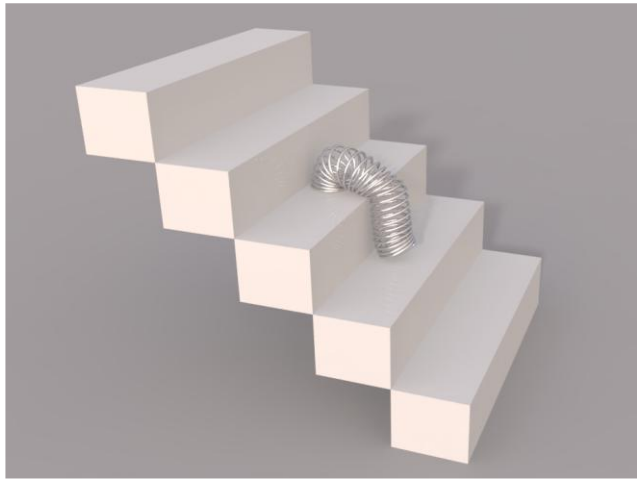
Content

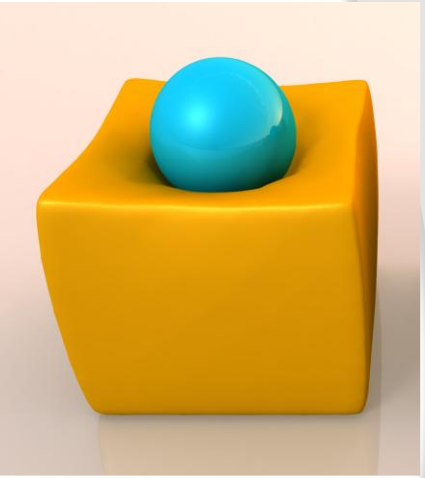
- Motivation
- Position-Based Dynamics (PBD)
- eXtended Position-Based Dynamics (XPBD)
- Deformable Solids
- Rigid Bodies
- Fluids
- Conclusion

Conclusion

- Position-based methods:
  - are fast, stable and simple to implement,
  - provide a high level of control,
  - can simulate deformable solids (1D, 2D, 3D), multibody systems, fluids and granular materials,
  - support complex physical effects like anisotropy or plasticity and
  - provide visual plausibility.

Thank you for your attention!



https://github.com/InteractiveComputerGraphics/PositionBasedDynamics