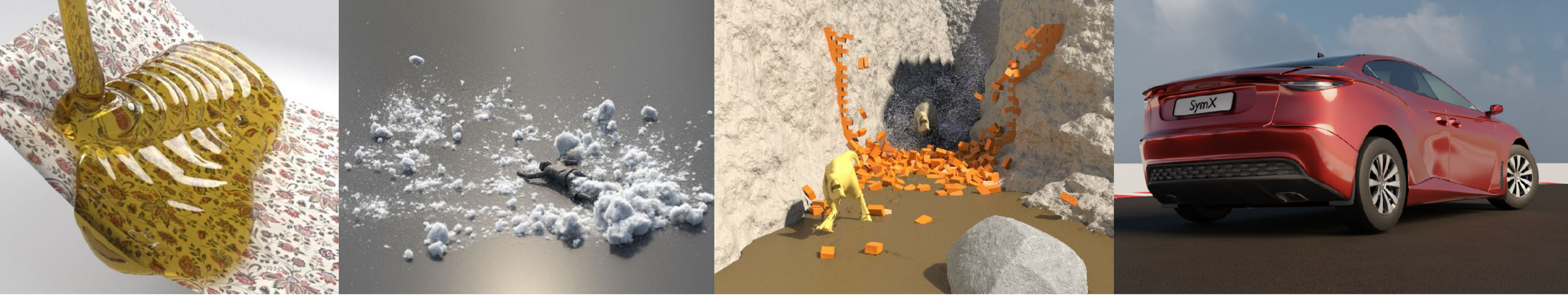
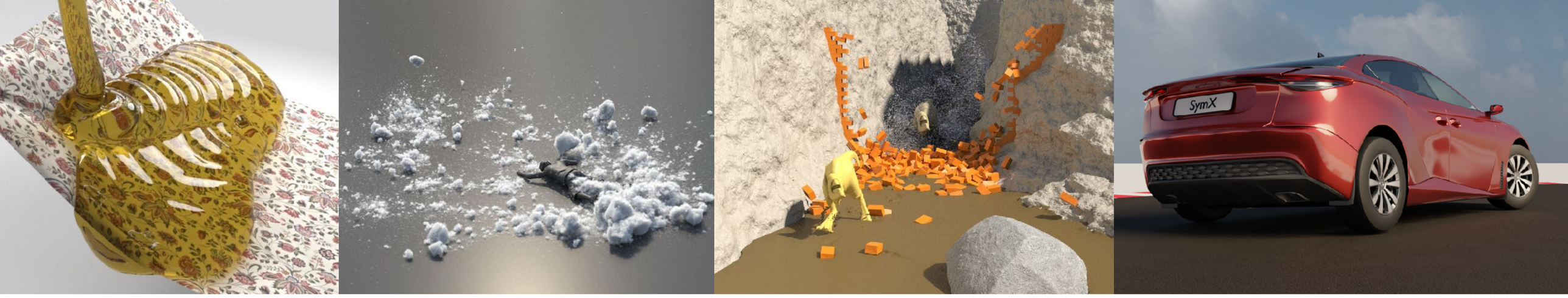


Simulation Methods for Multiphysics Phenomena in Visual Computing

Fabian Löschner, Stefan Rhys Jeske, José Antonio Fernández-Fernández, Jan Bender

Computer Animation Group, RWTH Aachen University

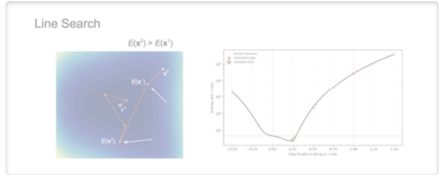


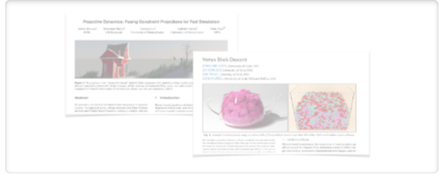
Part II
Energy-based Multiphysics Models

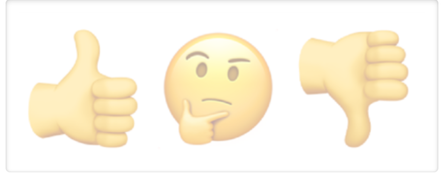
Fabian Löschner, Stefan Rhys Jeske, José Antonio Fernández-Fernández, Jan Bender

Computer Animation Group, RWTH Aachen University

Outline







1. Mathematical foundations

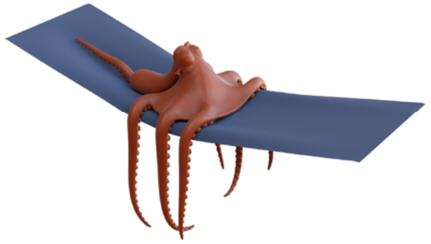
2. Physical models and coupling

3. Related methods (VBD, PD)

4. Summary: Models & Properties

Elastic Energies in Computer Graphics

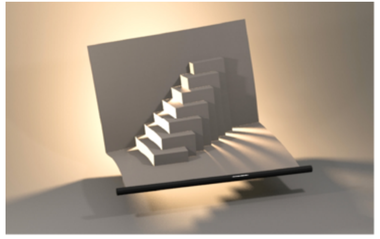
Deformable Solids



Dynamic Deformables [Kim, Eberle 2022]

Cloth & Shells

Progressive Dynamics for Cloth and Shell Animation
[Zhang et al. 2024]



Second-Order Finite Elements for Deformable Surfaces
[Le et al. 2023]

Rods

Discrete Elastic Rods
[Bergou et al. 2008]

Surface Parametrization

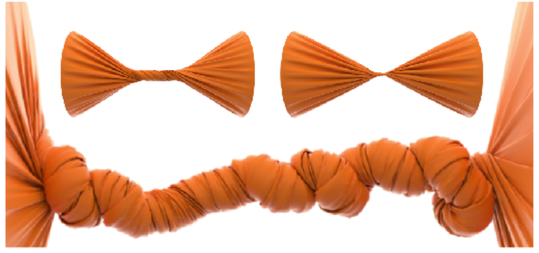
Analytic Eigensystems [Smith et al. 2019]

Shape Stylization & Optimization

Normal-Driven Spherical Shape Analogies
[Liu, Jacobson 2021]

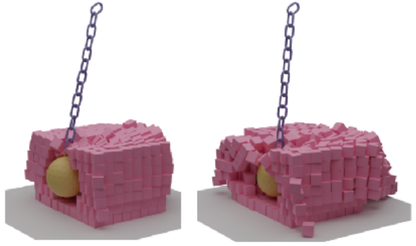
Multiphysics Energies

Contact



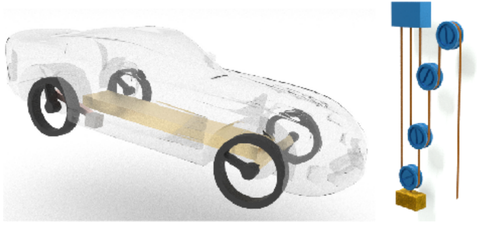
Codimensional Incremental Potential Contact [Li et al. 2021]

Rigid Bodies



Intersection-free Rigid Body Dynamics [Ferguson et al. 2021]

Multibody Systems

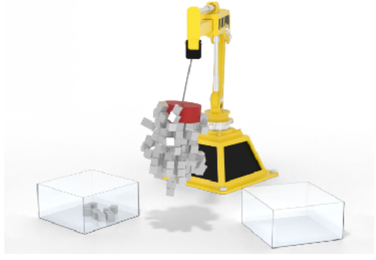


A Unified Newton Barrier Method for Multibody Dynamics
[Chen et al. 2022]

Fluids

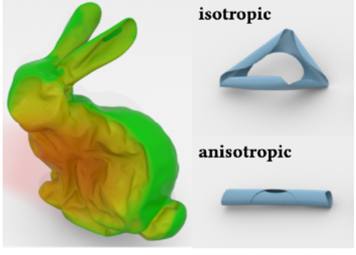
A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling
[Xie et al. 2023]

Magnetism



Strongly Coupled Simulation of Magnetic Rigid Bodies
[Westhofen et al. 2024]

Heating & Wetting



Physical Simulation of Environmentally Induced Thin Shell Deformation
[Chen et al. 2018]

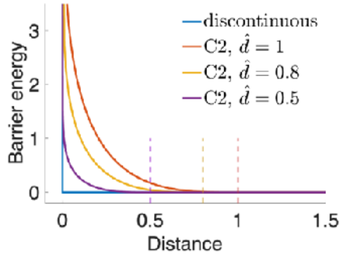
Energy-Based Multiphysics Modeling

Unifying mathematical formulation

- Physical systems modeled by **scalar potentials**
- Constraints modeled by **barrier potentials** or penalties

Typically

- …leads to fully implicit, **primal** formulation
- …solved as **unconstrained optimization** problem



Incremental Potential Contact [Li et al. 2020]

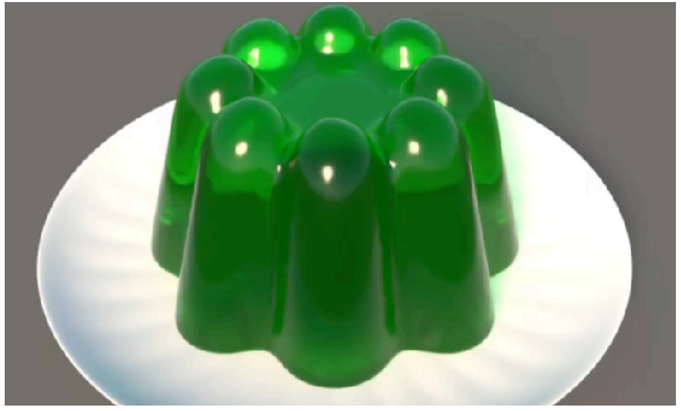
Elastic Deformables

- Naturally described using energies
- Change in shape → change in internal energy
- Simple spring-based models

  → e.g. $\phi_{\text{spring}}(\boldsymbol{x}) = \frac{1}{2}\kappa\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - L_0\right)^2$

- Continuum-based models: **strain energy densities**

  → e.g. Stable Neohookean: $\Psi_{\text{SNH}} = \frac{\mu}{2}(\text{tr}(\mathbf{F}^\top\mathbf{F}) - 3) - \mu(\det\mathbf{F} - 1) + \frac{\lambda}{2}(\det\mathbf{F} - 1)^2$



Higher-Order Time Integration for Deformable Solids [Löschner et al. 2020]

Mass Spring Systems

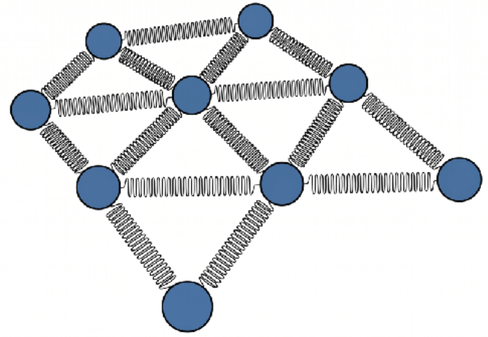
- Needs mesh of geometry (e.g. cloth → tri mesh, solid → tet mesh)
  - Vertices $i \in \mathscr{V}$ become **point masses**: position $\boldsymbol{x}_i$ and mass m_i
  - Edges $(i,j) \in \mathscr{E}$ become **springs**: stiffness κ_{ij} and rest length L_{ij}
- Energy potential per edge: $\phi_{ij} = \frac{1}{2}\kappa_{ij}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - L_{ij}\right)^2$

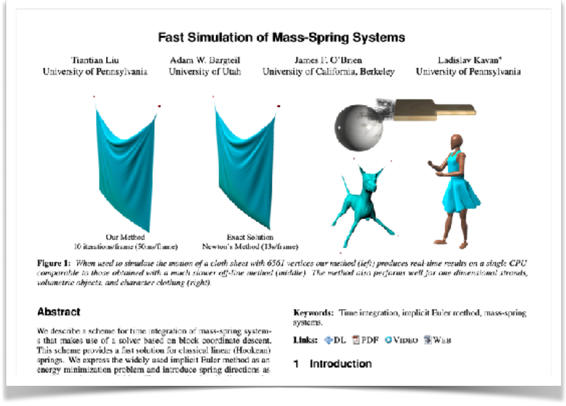
Properties:

- ✅ Easy to implement
- ✅ Extremely fast implementation variants
- 💥 No inherent volume conservation
- 💥 Elastic behavior extremely mesh dependent

Hooke's law:

$$\boldsymbol{f}_i = -\frac{\partial \phi_{ij}}{\partial \boldsymbol{x}_i} = -\kappa_{ij}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - L_{ij}\right)\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}$$





Fast Simulation of Mass-Spring Systems

Tiantian Liu, University of Pennsylvania
Adam W. Bargteil, University of Utah
James F. O'Brien, University of California, Berkeley
Ladislav Kavan*, University of Pennsylvania

Our Method
10 iterations/frame (50ms/frame)
Exact Solution
Newton's Method (1hr/frame)

Figure 1: *When used to simulate the motion of a cloth sheet with 6561 vertices our method (left) produces real time results on a single CPU comparable to those obtained with a much slower off-line method (middle). The method also performs well for our dimensional threads, volumetric objects, and character clothing (right).*

Abstract

We describe a scheme for time integration of mass-spring systems that makes use of a solver based on block coordinate descent. This scheme provides a fast solution for classical linear (Hookean) springs. We express the widely used implicit Euler method as an energy minimization problem and introduce spring direction as

Keywords: Time integration, implicit Euler method, mass-spring systems.

Links: DL PDF Video Web

1 Introduction

Kinematic Relations

- Lagrangian description using reference configuration $\boldsymbol{X}$
- Displacement field $\boldsymbol{u}(\boldsymbol{X}, t)$ gives current configuration $\boldsymbol{x}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t)$
- Deformation gradient: local linear transformation of material around a point

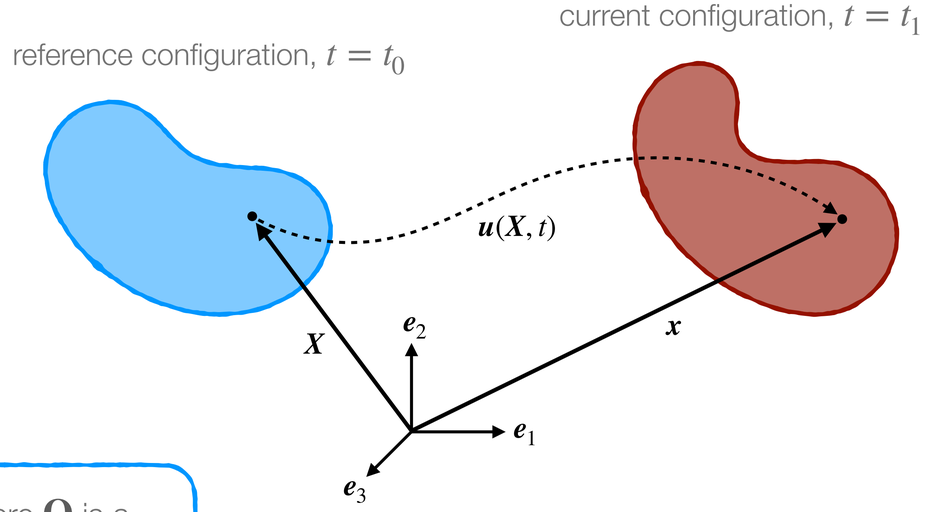
$$\mathbf{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \nabla \boldsymbol{u} + \mathbb{1}$$

Encodes transformation, not just deformation:

- Stretching 👍
- Shearing 👍
- Rotation (and reflection) 👎

Set $\tilde{\boldsymbol{x}}(\boldsymbol{X}) := \mathbf{Q}\,\boldsymbol{x}(\boldsymbol{X})$, where $\mathbf{Q}$ is a rotation matrix. Then:

$$\tilde{\mathbf{F}} = \frac{\partial \tilde{\boldsymbol{x}}}{\partial \boldsymbol{X}} = \frac{\partial (\mathbf{Q}\boldsymbol{x})}{\partial \boldsymbol{X}} = \mathbf{Q}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \mathbf{Q}\mathbf{F}$$

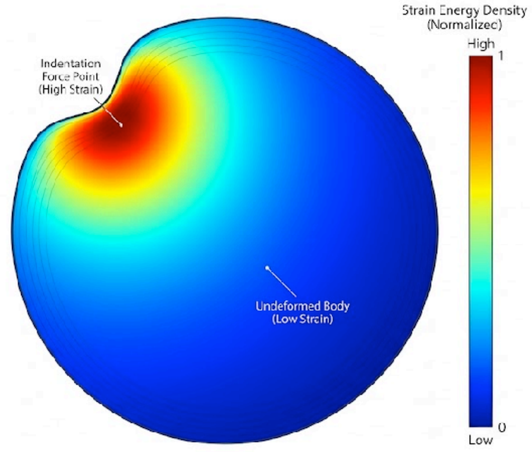


Constitutive Models

- Material defined by scalar **strain energy density $\Psi(\mathbf{F})$** at every point
- Simplest classic material: **Linear Elasticity**
  - Isotropic generalization of Hooke's Law
  - $\Psi_{\text{lin}}(\boldsymbol{\epsilon}) = \frac{\lambda}{2}\text{tr}(\boldsymbol{\epsilon})^2 + \mu\boldsymbol{\epsilon} : \boldsymbol{\epsilon}$ with infinitesimal strain tensor $\boldsymbol{\epsilon} = \frac{1}{2}\left(\mathbf{F} + \mathbf{F}^\top\right) - \mathbf{I}$

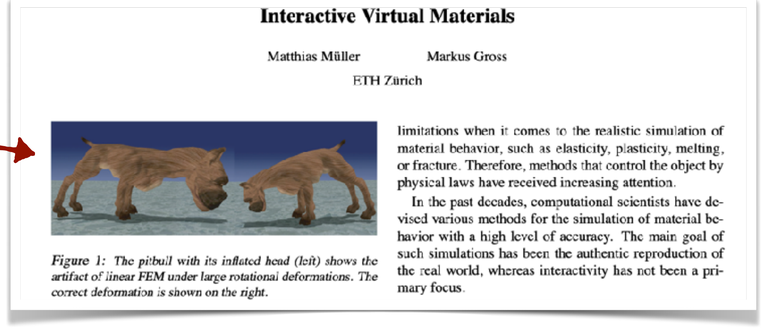
volume conservation

shape change



Properties: → Constant Hessian 🥳

→ Artifacts for finite rotations 💥



Interactive Virtual Materials

Matthias Müller Markus Gross

ETH Zürich

Figure 1: The pitbull with its inflated head (left) shows the artifact of linear FEM under large rotational deformations. The correct deformation is shown on the right.

limitations when it comes to the realistic simulation of material behavior, such as elasticity, plasticity, melting, or fracture. Therefore, methods that control the object by physical laws have received increasing attention.

In the past decades, computational scientists have devised various methods for the simulation of material behavior with a high level of accuracy. The main goal of such simulations has been the authentic reproduction of the real world, whereas interactivity has not been a primary focus.

Nonlinear Constitutive Models

- Using nonlinear strain measure, e.g. Green Strain: $\mathbf{E} = \frac{1}{2}(\mathbf{F}^\top\mathbf{F} - \mathbb{1})$

Polar decomposition: $\mathbf{F} = \mathbf{RS}$

$$\mathbf{E} = \frac{1}{2}((\mathbf{RS})^\top\mathbf{RS} - \mathbb{1})$$
$$= \frac{1}{2}(\mathbf{S}^\top\mathbf{R}^\top\mathbf{RS} - \mathbb{1})$$
$$= \frac{1}{2}(\mathbf{S}^\top\mathbf{S} - \mathbb{1})$$

Discards rotation

- e.g. **St. Venant Kirchhoff** model:

$$\Psi_{\text{StVK}} = \frac{\lambda}{2}\text{tr}(\mathbf{E})^2 + \mu\mathbf{E} : \mathbf{E}$$

(Recall: $\Psi_{\text{lin}}(\boldsymbol{\epsilon}) = \frac{\lambda}{2}\text{tr}(\boldsymbol{\epsilon})^2 + \mu\boldsymbol{\epsilon} : \boldsymbol{\epsilon}$)

- Using nonlinear invariants, e.g. from [Smith et al. 2019]: $I_1 = \text{tr}(\mathbf{S}),\ I_2 = \text{tr}(\mathbf{F}^\top\mathbf{F}),\ I_3 = \det\mathbf{F}$

- e.g. **Stable Neo-Hookean** model:

$$\Psi_{\text{SNH}} = \frac{\mu}{2}(I_2 - 3) - \mu(I_3 - 1) + \frac{\lambda}{2}(I_3 - 1)^2$$

- Interpretation: change in energy density due to local length/area/volume changes


Stable Neo-Hookean Flesh Simulation

BREANNAN SMITH, Pixar Animation Studios
FERNANDO DE GOES, Pixar Animation Studios
THEODORE KIM, Pixar Animation Studios

Elastic Deformables using FEM

- Continuum-based models: **strain energy densities**
  - → e.g. Stable Neohookean: $\Psi_{\mathrm{SNH}} = \frac{\mu}{2}(\mathrm{tr}(\mathbf{F}^\top\mathbf{F}) - 3) - \mu(\det\mathbf{F} - 1) + \frac{\lambda}{2}(\det\mathbf{F} - 1)^2$
  - → Problem: How to discretize and integrate this?

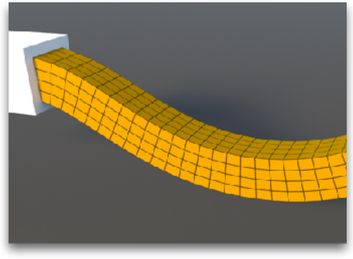
Finite Element Method for discretization

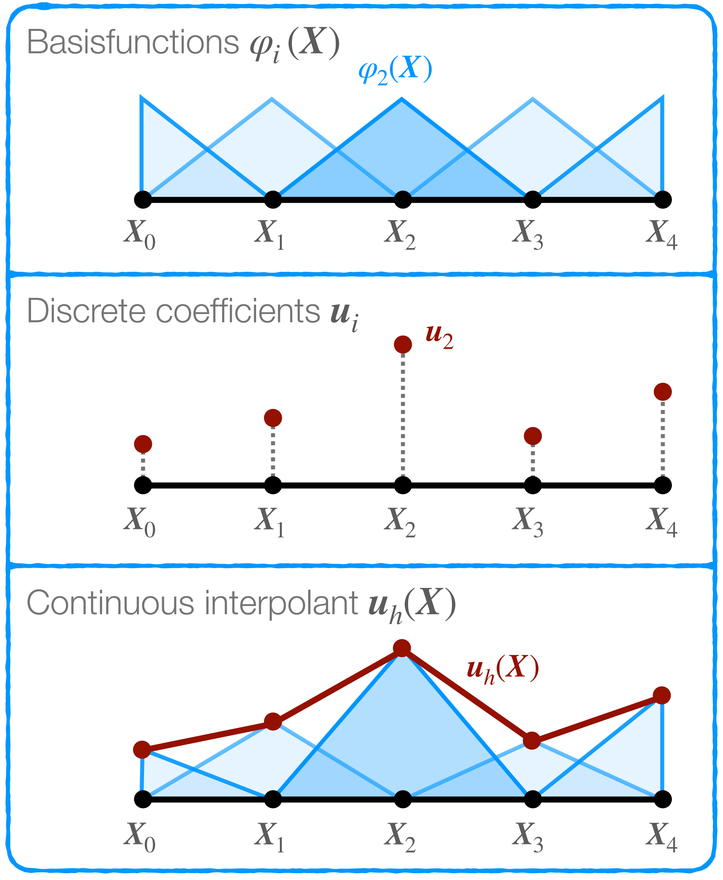
→ Decompose geometry into elements (tetrahedrons, hexahedrons)

→ Define discrete FEM interpolation of DOF: $\boldsymbol{u}_h(\boldsymbol{X}) = \sum_i^N \boldsymbol{u}_i\,\varphi_i(\boldsymbol{X})$

→ Per element: numerical integration of discrete strain energy density

→ Minimize sum of all element energies

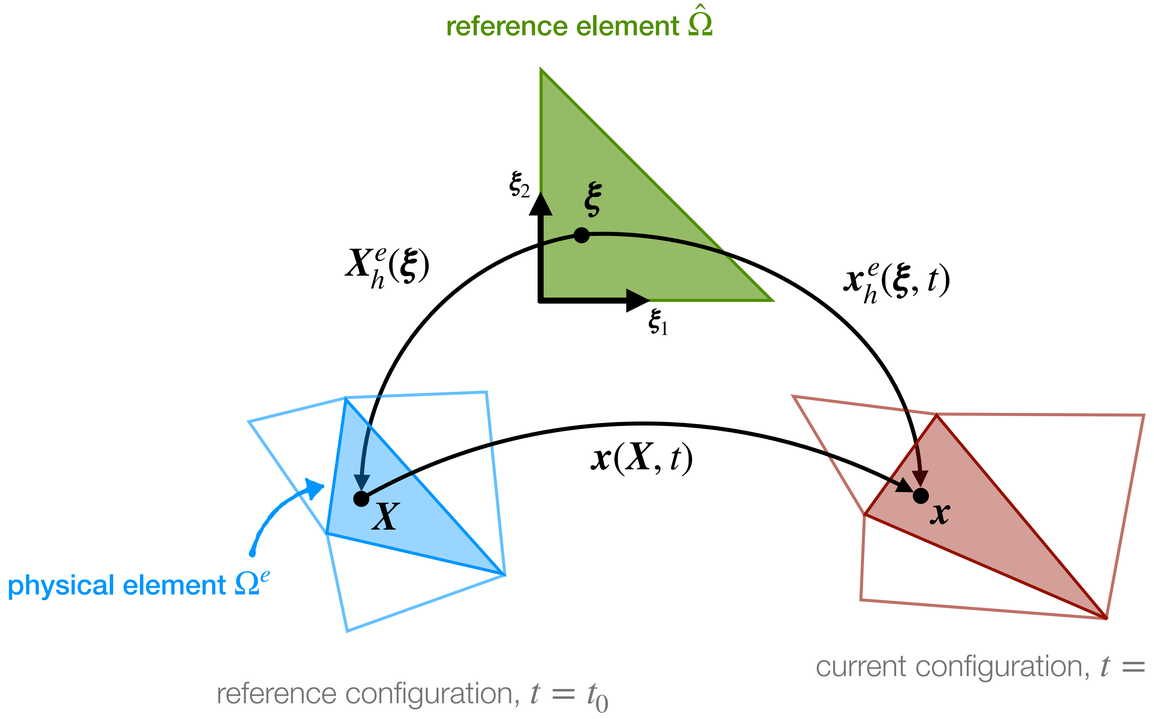




FEM: Evaluating the Strain Energy

- Input: continuous $\Psi(\mathbf{F})$, discrete DOF $\boldsymbol{u}_i = \boldsymbol{x}_i - \boldsymbol{X}_i$
- Goal: numerical integration of strain energy: $\phi^e = \int_{\Omega^e} \Psi(\mathbf{F})\,\mathrm{d}\boldsymbol{X}$
  - → Would require own quadrature rule per "physical element" Ω^e
- Introduce "reference element" $\hat{\Omega} \subset [0,1] \times [0,1]$ with coordinates $\boldsymbol{\xi} \in \hat{\Omega}$
  - → Evaluate quantities and integrals on $\hat{\Omega}$
- **Isoparametric element**: same basis functions for element geometry ($\boldsymbol{X}_h$) and unknowns ($\boldsymbol{x}_h$ or $\boldsymbol{u}_h$)
  - Rest coordinates: $\boldsymbol{X}_h^e(\boldsymbol{\xi}) = \sum_i \boldsymbol{X}_i^e \hat{\varphi}_i(\boldsymbol{\xi})$
  - Deformed coordinates: $\boldsymbol{x}_h^e(\boldsymbol{\xi}) = \sum_i \boldsymbol{x}_i^e \hat{\varphi}_i(\boldsymbol{\xi})$
- Deformation gradient on $\hat{\Omega}$:

$$\mathbf{F}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{x}_h^e}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}_h^e} = \mathbf{j}^e (\mathbf{J}^e)^{-1}$$



FEM: Evaluating the Strain Energy

- Integrate strain energy density $\Psi(\mathbf{F})$ and transform integral to reference element $\hat{\Omega}$:

$$\phi^e = \int_{\Omega^e} \Psi(\mathbf{F}(\boldsymbol{X}))\,\mathrm{d}\boldsymbol{X} = \int_{\hat{\Omega}} \Psi(\mathbf{F}(\boldsymbol{\xi}))\,\det(\mathbf{J}^e)\,\mathrm{d}\boldsymbol{\xi}$$

- Apply quadrature rule with weights w_i and points $\hat{\boldsymbol{p}}_i \in \hat{\Omega}$

$$\phi^e \approx \sum_i^{N_{\mathrm{QP}}} w_i \Psi(\mathbf{F}(\hat{\boldsymbol{p}}_i))\,\det(\mathbf{J}^e)$$

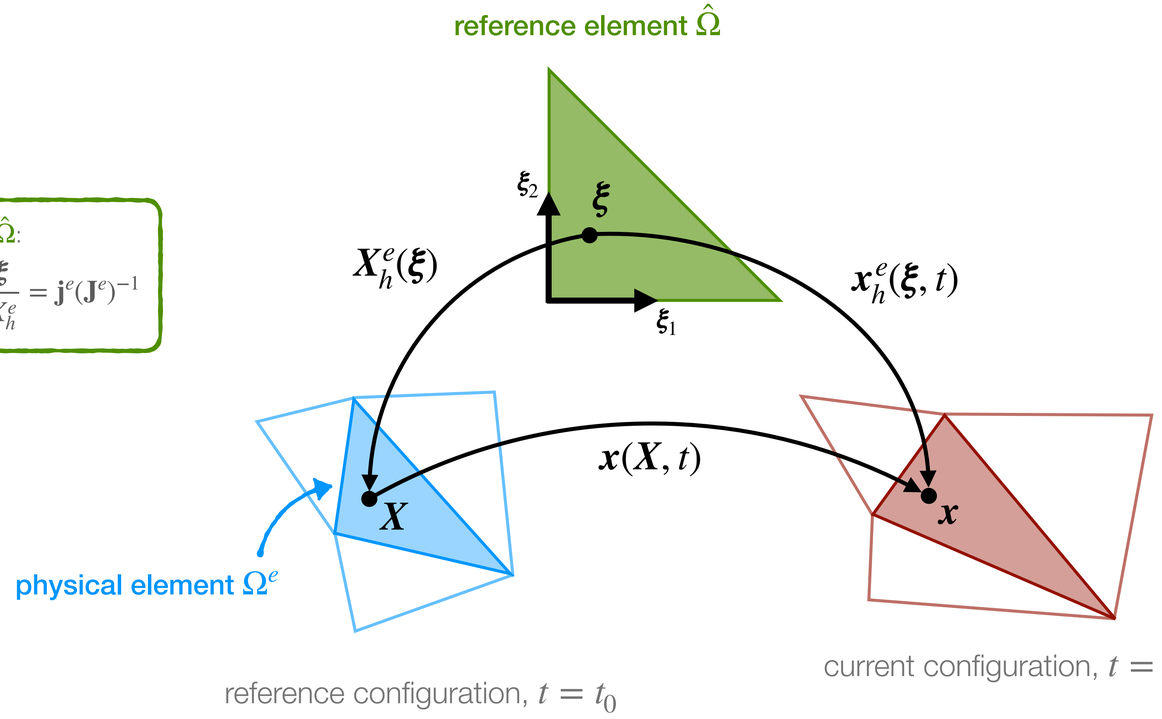
- In particular, for **linear tetrahedra**:
  - Derivatives constant, one quadrature point sufficient
  - $\mathbf{J}^e = [X_1 - X_0, X_2 - X_0, X_3 - X_0]$
  - $\mathbf{j}^e = [x_1 - x_0, x_2 - x_0, x_3 - x_0]$
  - $\det(\mathbf{J}^e)$ is **volume** V^e of physical element Ω^e

  ↳ This yields: $\phi^e = V^e \Psi\left(\mathbf{j}^e(\mathbf{J}^e)^{-1}\right)$

- Last step: **sum over all elements** and add to incremental potential

Deformation gradient on $\hat{\Omega}$:

$$\mathbf{F}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{x}_h^e}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}_h^e} = \mathbf{j}^e(\mathbf{J}^e)^{-1}$$



Modelling Dynamic Contact

- Incremental Potential Contact [Li et al. 2020]

  ↳ Goal: intersection-free simulations in unconstrained optimization

- Introduce smoothly clamped barriers:

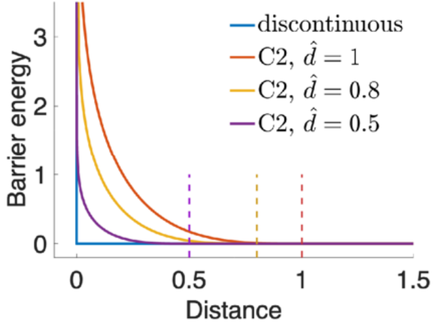
$$b(d,\hat{d}) = \begin{cases} -(d-\hat{d})^2 \ln\left(\frac{d}{\hat{d}}\right), & 0 < d < \hat{d} \\ 0 & d \geq \hat{d} \end{cases}$$

  With contact potential $\phi_C(\boldsymbol{x}) = \kappa \sum_{k \in C} b(d_k(\boldsymbol{x}))$

- Apply barriers to all triangle-vertex and edge-edge pairs $k \in C$

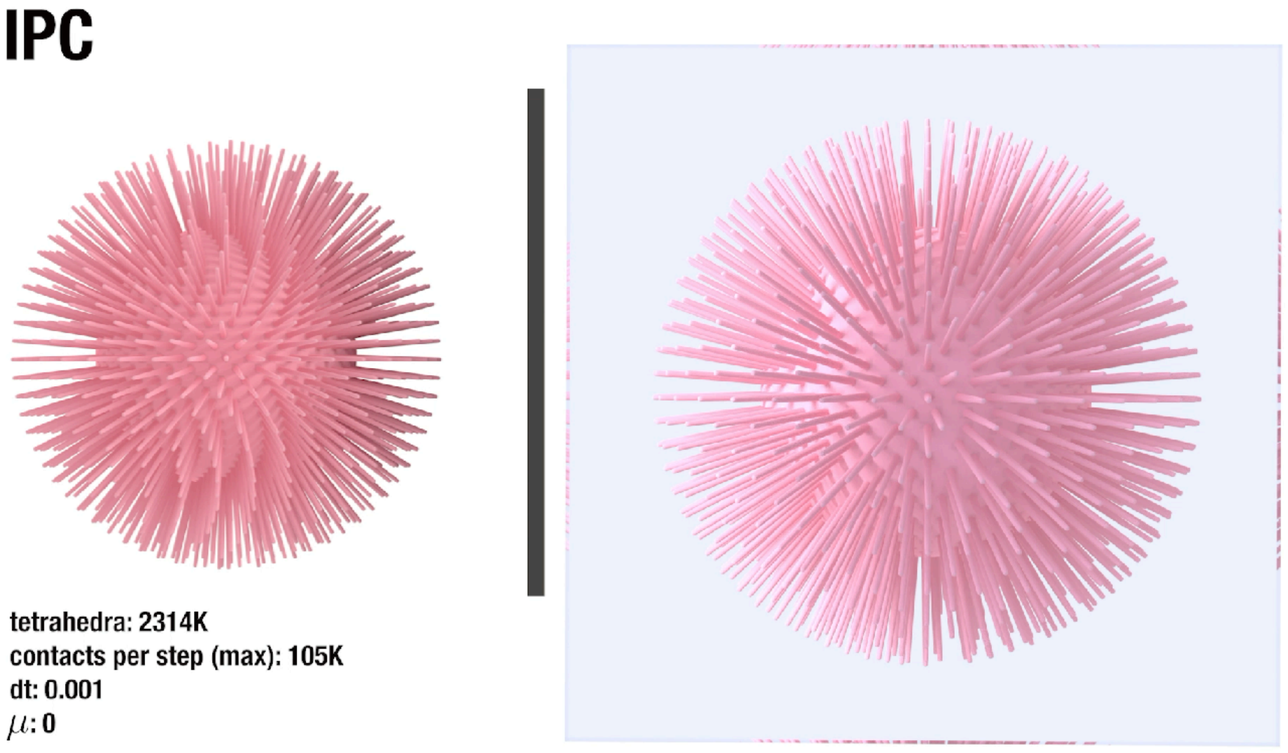
- Filtered line search: CCD determines upper bound for step

Incremental Potential Contact [Li et al. 2020]



Incremental Potential Contact [Li et al. 2020]

Intersection-free Dynamic Contact



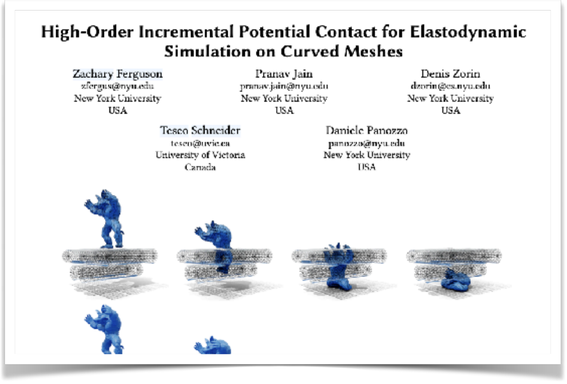
Incremental Potential Contact [Li et al. 2020]

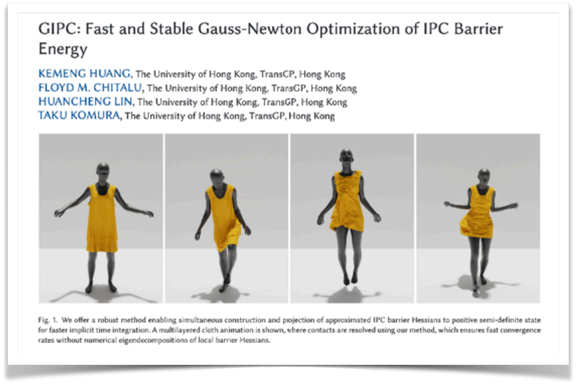
Intersection-free Dynamic Contact

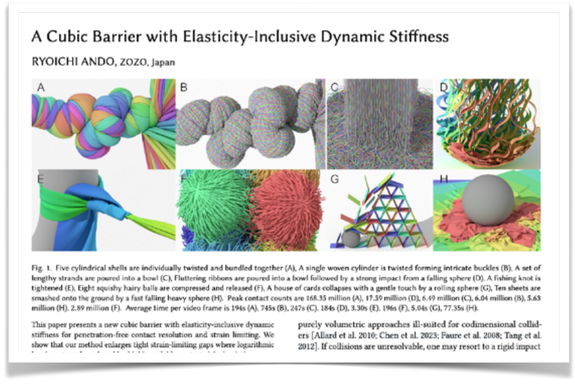
tetrahedra: 181K
contacts per step (max): 277K
dt: 0.01
μ: 0

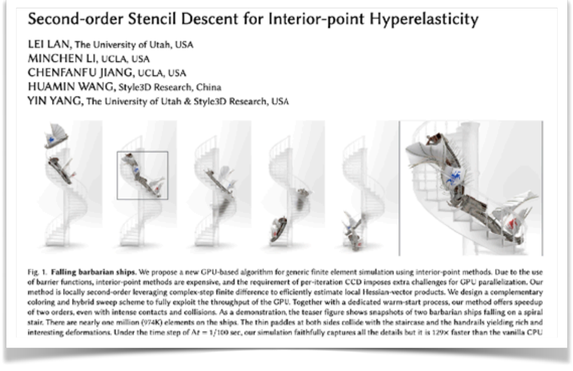
Incremental Potential Contact [Li et al. 2020]

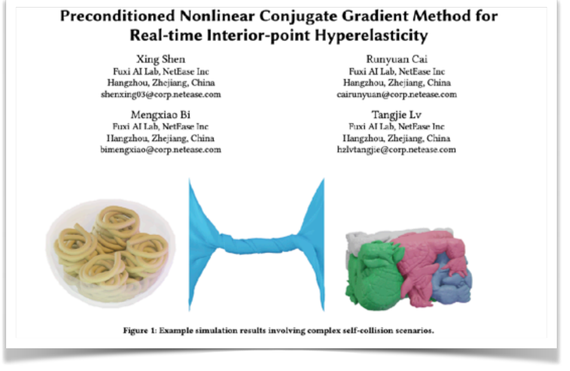
Intersection-free Dynamic Contact

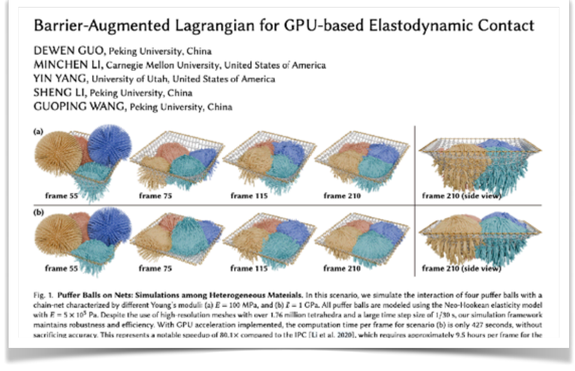






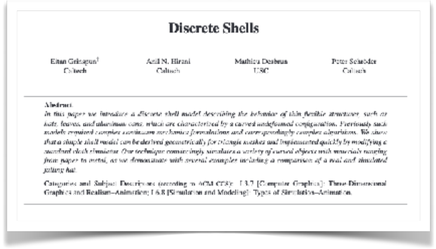


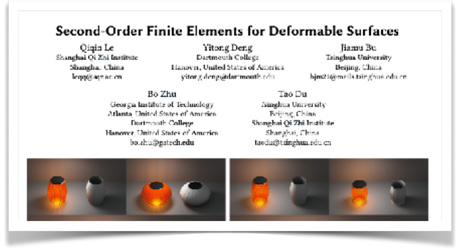




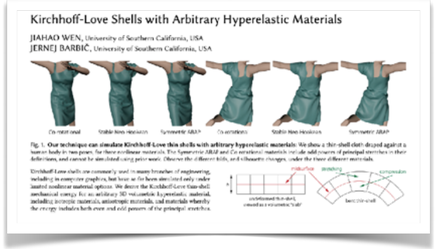
Cloth, Shells & Rods

Shell models (examples…)



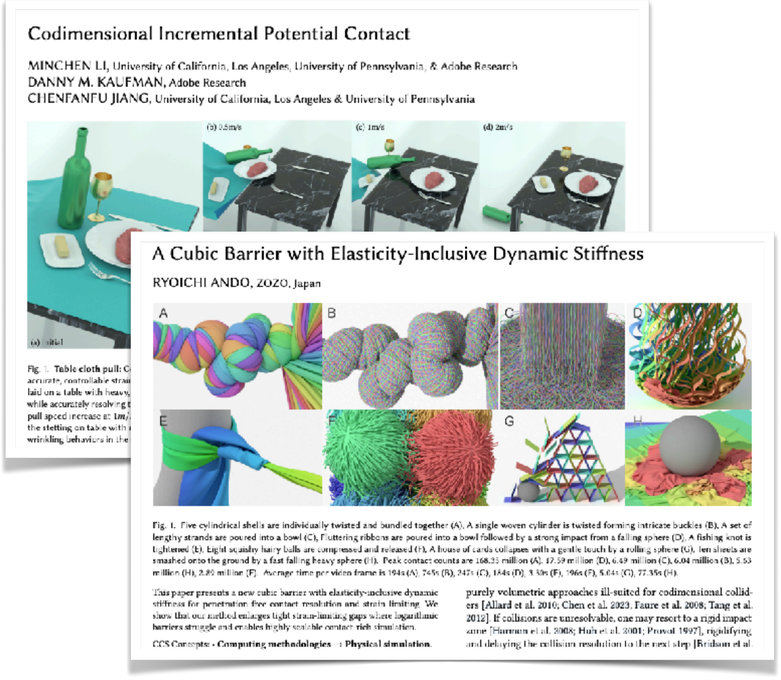


Geometrically motivated



Continuous models

Intersection-free contact



Ingredients for basic shell simulation

- Input: Non-planar triangle mesh in 3D
- **In-plane stresses:** 2D elasticity
  - Neo-Hookean: $\Psi_{\mathrm{NH}} = \frac{\mu}{2}(\mathrm{tr}(\mathbf{C}) - 2) - \mu \ln(J) + \frac{\lambda}{2} \ln(J)^2$
    with $\mathbf{C} = \mathbf{F}^\mathsf{T}\mathbf{F}$ and $J = \sqrt{\det(\mathbf{C})}$
  - Deformation gradient: $\mathbf{F} \in \mathbb{R}^{3\times 2}$ such that $\mathbf{j}^e = \mathbf{F}\mathbf{J}^e$
    (maps vectors from 2D reference to 3D current configuration)
  - Current configuration: $\mathbf{j^e} = [\boldsymbol{x_1} - \boldsymbol{x_0}, \boldsymbol{x_2} - \boldsymbol{x_0}]$
  - For reference configuration, rotate triangle Ω^e into planar configuration:
    (only edge lengths relevant, model Ψ_{NH} is rotationally invariant)
    - Translate to origin: $\boldsymbol{u}_0 = [0, 0]^\mathsf{T}$
    - Align first edge with x-axis: $\boldsymbol{u}_1 = [\|\boldsymbol{X}_1 - \boldsymbol{X}_0\|, 0]^\mathsf{T}$
    - Rotate third vertex into x-y plane:

$$\boldsymbol{u}_{2,x} = (\boldsymbol{X}_1 - \boldsymbol{X}_0) \cdot (\boldsymbol{X}_2 - \boldsymbol{X}_0) \, \|\boldsymbol{X}_1 - \boldsymbol{X}_0\|^{-1}$$

$$\boldsymbol{u}_{2,y} = \sqrt{\|\boldsymbol{X}_2 - \boldsymbol{X}_0\|^2 - \boldsymbol{u}_{2,x}{}^2}$$

$$\boldsymbol{u}_2 = [\boldsymbol{u}_{2,x}, \boldsymbol{u}_{2,y}]^\mathsf{T}$$

    - Then: $\mathbf{J^e} = [\boldsymbol{u_1} - \boldsymbol{u_0}, \boldsymbol{u_2} - \boldsymbol{u_0}]$

Alternatively:


A Finite Element Formulation of Baraff-Witkin Cloth

Theodore Kim
Yale University

- **Strain limiting:** Singular Value Barriers [Li 2021]
  - Compute singular values σ_1, σ_2 of $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbb{1})$
  - Define barrier per singular value with strain limit s and barrier activation threshold $\hat{s}$:

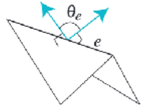
$$b_{\mathrm{SL},i}(\boldsymbol{x}) = \begin{cases} -\left(\frac{\hat{s} - \sigma_i}{s - \hat{s}}\right)^2 \ln\left(\frac{s - \sigma_i}{s - \hat{s}}\right), & \sigma_i > \hat{s} \\ 0 & \sigma_i \leq \hat{s} \end{cases}$$

  - Integrate over element volume:

$$\phi^e_{\mathrm{SL}} = \kappa_{\mathrm{SL}} \sum_i \int_{\Omega^e} b_{\mathrm{SL},i}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{X} \approx \kappa_{\mathrm{SL}} V^e \sum_i b_{\mathrm{SL},i}$$

- **Out of plane bending:** Discrete Shells [Grinspun 2003] or Quadratic Bending [Bergou 2006]
  - Potential per interior edge, depends on dihedral angle:

$$\phi_{\mathrm{B}}(\boldsymbol{x}) = \kappa_{\mathrm{B}} \frac{\|\boldsymbol{e}_0\|}{h_0} \left(\theta(\boldsymbol{x}) - \theta_0\right)^2$$



  - For rest-flat shells a quadratic formulation in vertex positions exists (constant Hessian)

Rigid Bodies & Affine Bodies

- DOF per rigid body: translation $\boldsymbol{x}_i \in \mathbb{R}^3$ and rotation vector $\boldsymbol{\theta}_i \in \mathbb{R}^3$
- Inertial term for rotation DOF:

$$E_R(\mathbf{R}(\boldsymbol{\theta})) = \sum_i^{N_{\text{rb}}} \left(\frac{1}{2}\text{tr}(\mathbf{R}_i \mathbf{J}_i \mathbf{R}_i^T) - \text{tr}(\mathbf{R}_i \mathbf{J}_i \tilde{\mathbf{R}}_i^T) \right)$$

with $\tilde{\mathbf{R}}_i = \mathbf{R}_i^{\text{prev}} + h\dot{\mathbf{R}}_i^{\text{prev}} + h^2 \tau_i \mathbf{J}_i^{-1}$

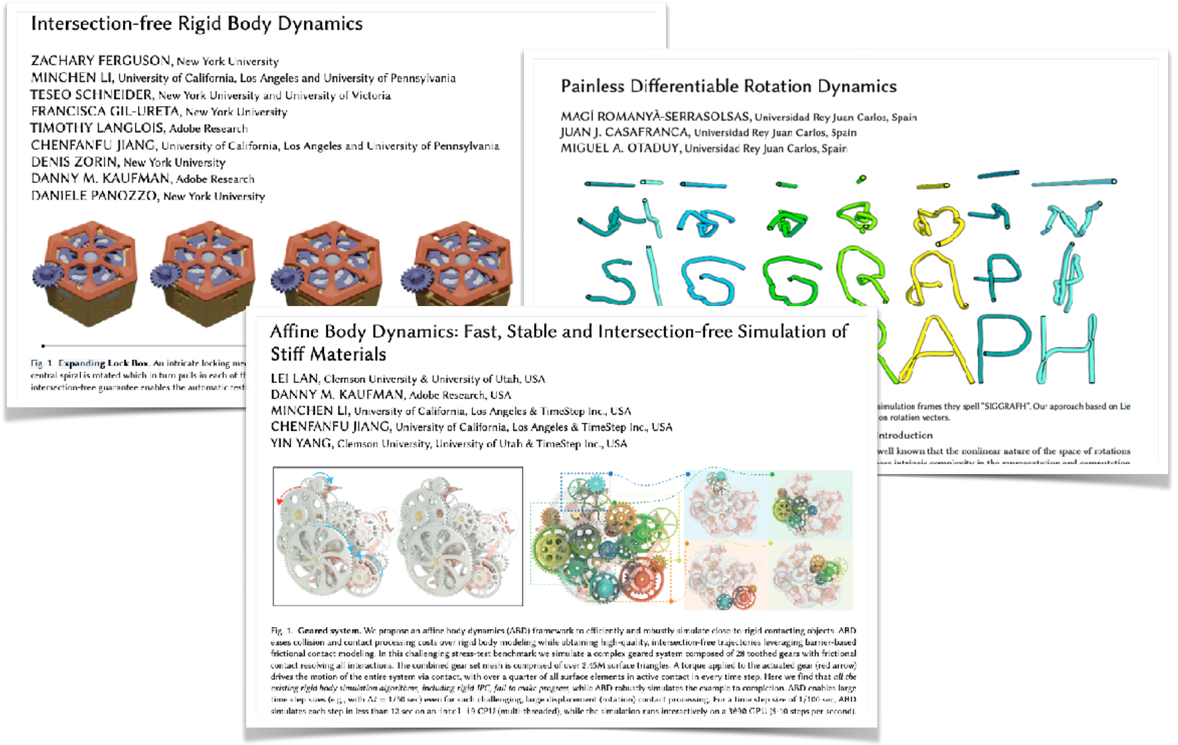
- Non-linear mapping from DOF to vertices

  → Small time steps or “curved CCD” for intersection-free contact

- Alternative: **Affine Bodies**

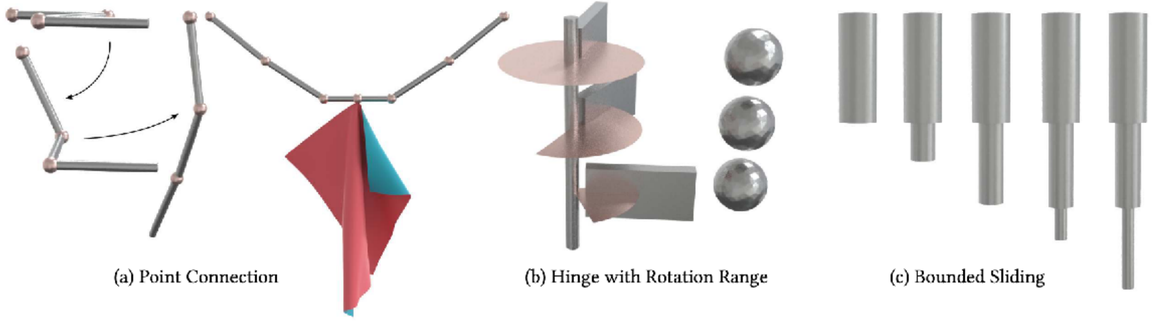
  → Body embedded in a single tetrahedron

  → Use any stiff material model ($E > 10^8 \text{Pa}$)

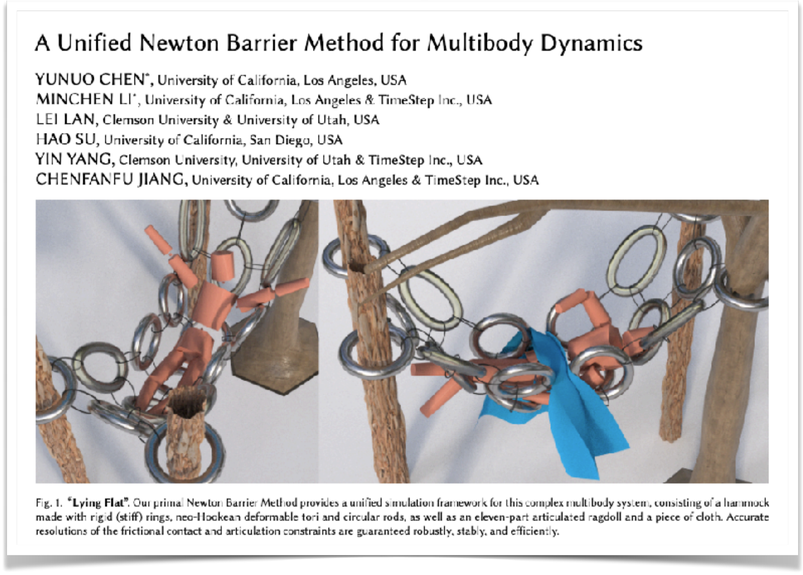


Multi-Body Systems

- Coupling deformable & rigid bodies with joints and motors
- Reformulate constraints using penalties and barriers
  - → Equality constraints: $\phi_{\mathrm{Eq}}(\boldsymbol{x}) = \frac{1}{2}k(c(\boldsymbol{x}))^2$
  - → Inequality constraints: $\phi_{\mathrm{Ineq}}(\boldsymbol{x}) = \kappa b(c(\boldsymbol{x}))$



(a) Point Connection (b) Hinge with Rotation Range (c) Bounded Sliding



A Unified Newton Barrier Method for Multibody Dynamics

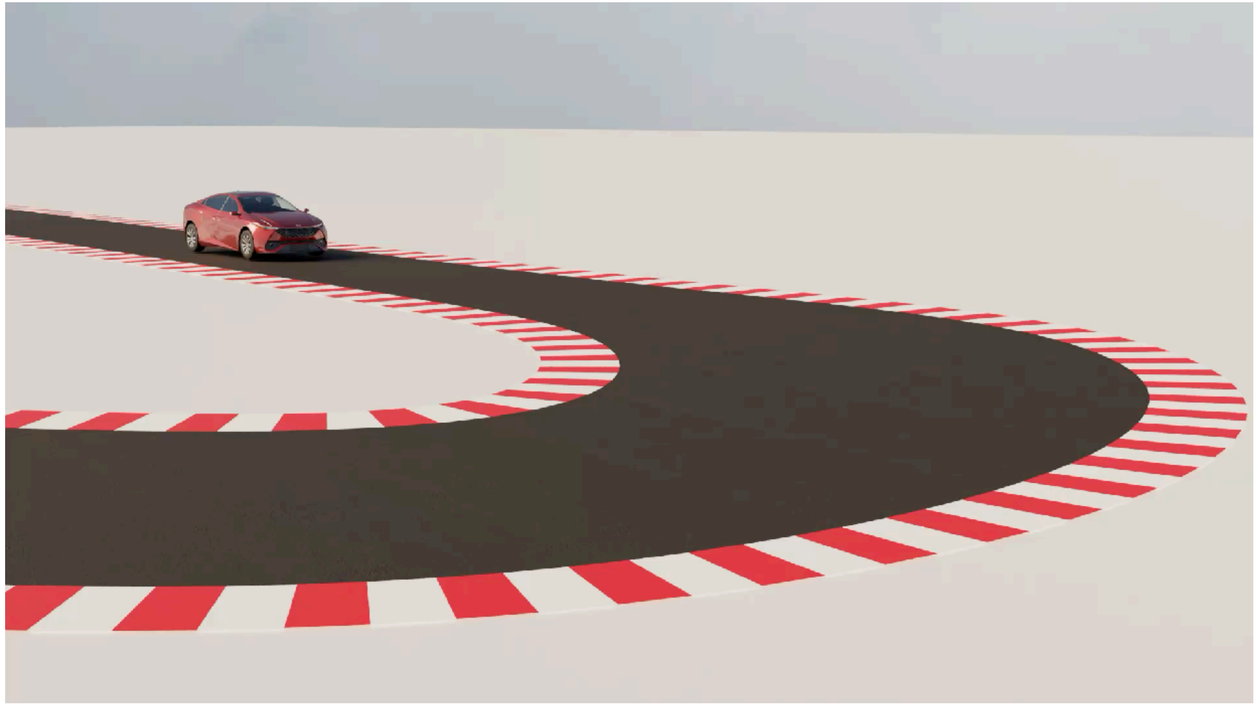
YUNUO CHEN*, University of California, Los Angeles, USA
MINCHEN LI*, University of California, Los Angeles & TimeStep Inc., USA
LEI LAN, Clemson University & University of Utah, USA
HAO SU, University of California, San Diego, USA
YIN YANG, Clemson University, University of Utah & TimeStep Inc., USA
CHENFANFU JIANG, University of California, Los Angeles & TimeStep Inc., USA

Fig. 1. **"Lying Flat".** Our primal Newton Barrier Method provides a unified simulation framework for this complex multibody system, consisting of a hammock made with rigid (stiff) rings, neo-Hookean deformable tori and circular rods, as well as an eleven-part articulated ragdoll and a piece of cloth. Accurate resolutions of the frictional contact and articulation constraints are guaranteed robustly, stably, and efficiently.

Multi-Body Systems

Strongly coupled systems:

- Rigid bodies for car body and rims
- Tires as deformable solids
- Dampened springs and slider joints for suspension
- Connected with axial joints
- Rotation limits for steering



SymX: Energy-based Simulation from Symbolic Expressions [Fernández-Fernández et al. 2025]

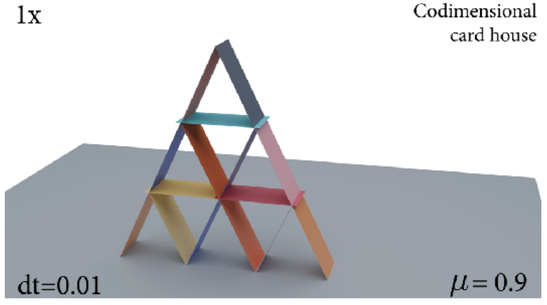
Dissipative Forces

- We need scalar potentials $\phi_i(\boldsymbol{x})$ such that $\boldsymbol{f}_i(\boldsymbol{x}) = -\nabla\phi_i(\boldsymbol{x})$

  → Possible for conservative and path-independent forces

- Adaptations required for dissipative/non-conservative forces:

Friction

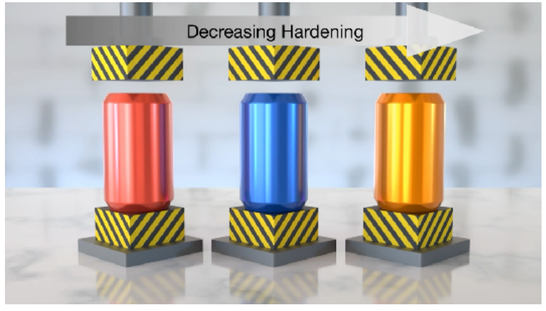


Intersection-free Rigid Body Dynamics [Ferguson et al. 2021]

Damping

Accurate Dissipative Forces in Optimization Integrators [Brown et al. 2018]

Plasticity



Energetically Consistent Inelasticity for Optimization Time Integration [Li et al. 2020]

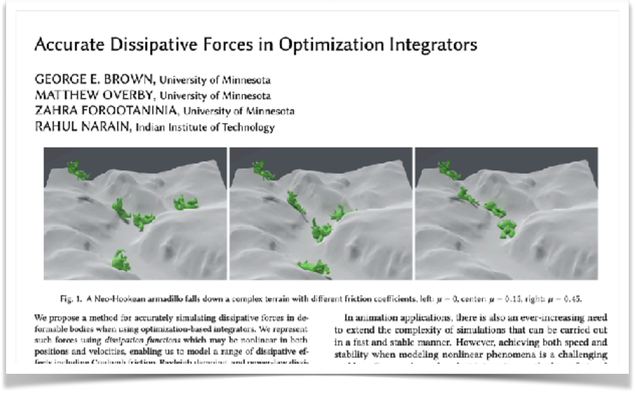
Damping

- Common damping & friction forces can be written as **dissipation potential**

  $R(\boldsymbol{x}, \boldsymbol{v})$ s.t. $\boldsymbol{f}_d = -\nabla_{\boldsymbol{v}} R(\boldsymbol{x}, \boldsymbol{v})$

- Not generally possible to define potential $\phi_d(\boldsymbol{x})$ s.t. $-\nabla_{\boldsymbol{x}} \phi_d = \boldsymbol{f}_d$

- Example: **Rayleigh damping**
  - Forces: $\boldsymbol{f}_d = -(\alpha \mathbf{M} + \beta \mathbf{K}(\boldsymbol{x}))\boldsymbol{v}$
  - Dissipation potential: $R_{\text{ray}}(\boldsymbol{x}, \boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\top(\alpha \mathbf{M} + \beta \mathbf{K}(\boldsymbol{x}))\boldsymbol{v}$
  - Approximation by lagging: $\mathbf{K}(\boldsymbol{x}) \approx \mathbf{K}(\boldsymbol{x}^{\text{prev}})$
  - Lagged potential: $\phi_{\text{ray}}(\boldsymbol{x}) = \frac{1}{2\Delta t}(\boldsymbol{x} - \boldsymbol{x}^{\text{prev}})^\top(\alpha \mathbf{M} + \beta \mathbf{K}(\boldsymbol{x}^{\text{prev}}))(\boldsymbol{x} - \boldsymbol{x}^{\text{prev}})$



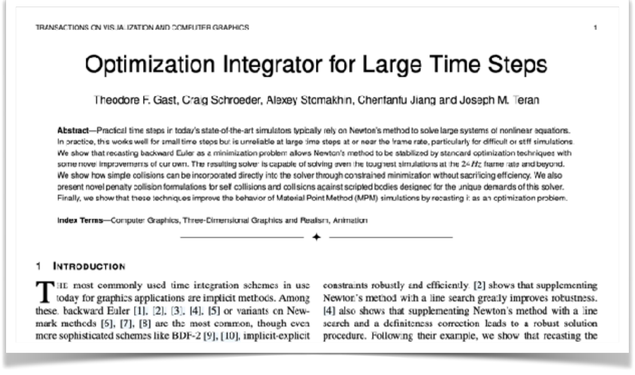
Accurate Dissipative Forces in Optimization Integrators

GEORGE E. BROWN, University of Minnesota
MATTHEW OVERBY, University of Minnesota
ZAHRA FOROOTANINIA, University of Minnesota
RAHUL NARAIN, Indian Institute of Technology

Fig. 1. A Neo-Hookean armadillo falls down a complex terrain with different friction coefficients. left: $\mu = 0$, center: $\mu = 0.15$, right: $\mu = 0.45$.

We propose a method for accurately simulating dissipative forces in deformable bodies when using optimization-based integrators. We represent such forces using *dissipation functions* which may be nonlinear in both positions and velocities, enabling us to model a range of dissipative effects including Coulomb friction, Rayleigh damping, and numerical dissi-

In animation applications, there is also an ever-increasing need to extend the complexity of simulations that can be carried out in a fast and stable manner. However, achieving both speed and stability when modeling nonlinear phenomena is a challenging



TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS 1

Optimization Integrator for Large Time Steps

Theodore F. Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang and Joseph M. Teran

Abstract—Practical time steps in today's state-of-the-art simulators typically rely on Newton's method to solve large systems of nonlinear equations. In practice, this works well for small time steps but is unreliable at large time steps at or near the frame rate, particularly for difficult or stiff simulations. We show that recasting backward Euler as a minimization problem allows Newton's method to be stabilized by standard optimization techniques with some novel improvements of our own. The resulting solver is capable of solving even the toughest simulations at the 24 Hz frame rate and beyond. We show how simple collisions can be incorporated directly into the solver through constrained minimization without sacrificing efficiency. We also present novel penalty collision formulations for self collisions and collisions against scripted bodies designed for the unique demands of this solver. Finally, we show that these techniques improve the behavior of Material Point Method (MPM) simulations by recasting it as an optimization problem.

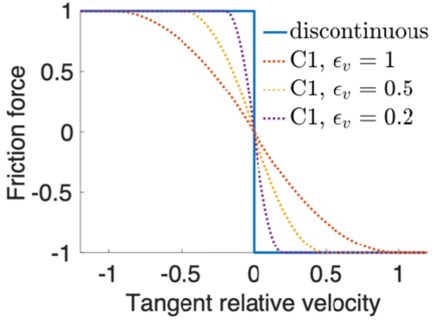
Index Terms—Computer Graphics, Three-Dimensional Graphics and Realism, Animation

1 Introduction

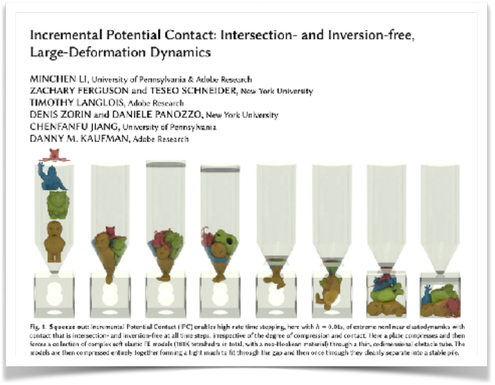
The most commonly used time integration schemes in use today for graphics applications are implicit methods. Among these, backward Euler [1], [2], [3], [4], [5] or variants on Newmark methods [5], [7], [8] are the most common, though even more sophisticated schemes like BDF-2 [9], [10], implicit-explicit constraints robustly and efficiently [2] shows that supplementing Newton's method with a line search greatly improves robustness. [4] also shows that supplementing Newton's method with a line search and a definiteness correction leads to a robust solution procedure. Following their example, we show that recasting the

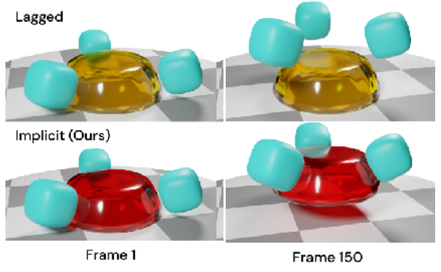
Friction

- Example: **Coulomb friction**
  - Again, can be formulated as dissipation potential $R(\boldsymbol{x}, \boldsymbol{v})$ s.t. $\boldsymbol{f}_f = -\nabla_{\boldsymbol{v}} R(\boldsymbol{x}, \boldsymbol{v})$
  - But not as energy potential without approximation
  - Also: Non-smoothness at **stick-slip transition**
- Solution of [Li et al. 2020]:
  - C^1-smooth transition between static and dynamic friction with velocity threshold ϵ_v
  - Lagging of tangential sliding basis and contact force magnitude
  - Only relative sliding displacement $\boldsymbol{x}$ is fully implicit
  - Can lead to inaccurate behavior in friction sensitive problems

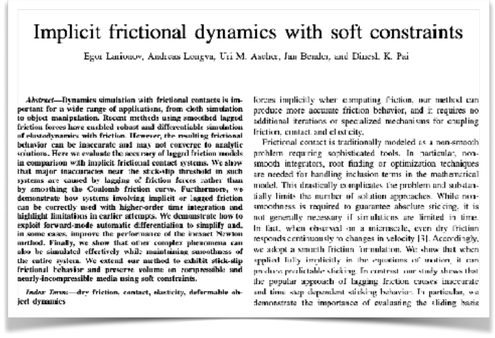


Incremental Potential Contact [Li et al. 2020]





Implicit frictional dynamics with soft constraints [Larionov et al. 2024]



Coupling Magnetic Effects

Magnetic Rigid Bodies

Strongly Coupled Simulation of Magnetic Rigid Bodies [Westhofen et al. 2024]

$$\phi_{\text{magn}} = -\sum_i V_i \mathbf{M}_i \cdot \mathbf{B}_{\text{ext},i}$$

Magnetoelastic Thin Shells

Simulation and Optimization of Magnetoelastic Thin Shells [Chen et al. 2022]

$$\phi_{\text{magn}} = -\sum_i h\mathbf{F}_i \mathbf{M}_i \cdot \mathbf{B}_{\text{ext},i}$$

FEM & SPH Coupling: Contact Proxy Splitting Method

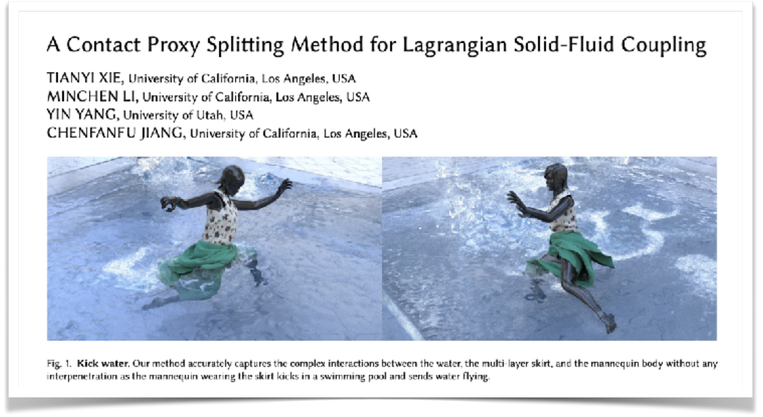
- Fully Lagrangian formulation
- Models **Weakly Compressible SPH** using quadratic potentials
  - e.g. incompressibility potential: $\phi_{\text{Inc}} = \sum_i \frac{\kappa_I}{2} V_0 \left(\frac{\rho_0}{\rho_i} - 1 \right)^2$
- Contact model: barriers on particle ↔ mesh distances

Problems of fully implicit formulation:

- "Connectivity" of SPH particles (neighbors of neighbors)
- Nonlinearity of barriers and deformables

Splitting approach:

1. Solve **fluid** phase with **linearized contact** forces
2. Solve **deformables** with **full contact** potential & **fluid proxy**

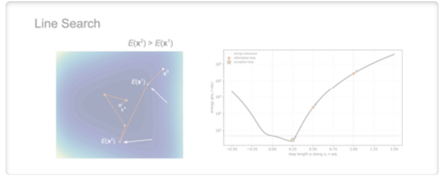


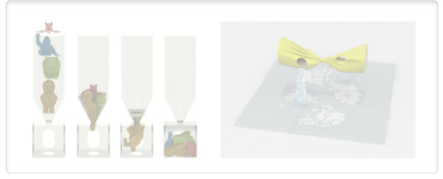
A Contact Proxy Splitting Method for Lagrangian Solid-Fluid Coupling

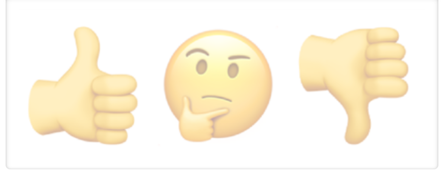
TIANYI XIE, University of California, Los Angeles, USA
MINCHEN LI, University of California, Los Angeles, USA
YIN YANG, University of Utah, USA
CHENFANFU JIANG, University of California, Los Angeles, USA

Fig. 1. **Kick water.** Our method accurately captures the complex interactions between the water, the multi-layer skirt, and the mannequin body without any interpenetration as the mannequin wearing the skirt kicks in a swimming pool and sends water flying.

Outline







1. Mathematical foundations

2. Physical models and coupling

3. Related methods (VBD, PD)

4. Summary: Models & Properties

Projective Dynamics

- Introduces local/global split
- **Local:** Non-linear **constraint projections**

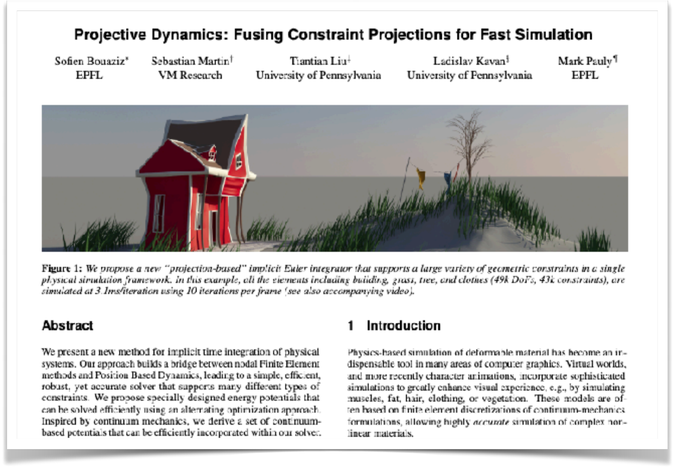
  → **Per constraint,** find closest "target positions" $\boldsymbol{p}_i$ that satisfy $\boldsymbol{C}_i(\boldsymbol{p}_i) = 0$

- **Global:** for fixed $\boldsymbol{p}_i$, minimize BE **incremental potential**

$$E_{\mathrm{PD}}(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{2\Delta t^2}(\boldsymbol{x} - \tilde{\boldsymbol{x}})^{\mathsf{T}}\mathbf{M}(\boldsymbol{x} - \tilde{\boldsymbol{x}}) + \sum_i d_i(\boldsymbol{x}, \boldsymbol{p}_i)$$

→ Where $d_i(\boldsymbol{x}, \boldsymbol{p}_i)$ are **quadratic** functions, pulling the DOF $\boldsymbol{x}$ towards $\boldsymbol{p}_i$

E_{PD} has Constant Hessian → Pre-Factorize



Projective Dynamics: Fusing Constraint Projections for Fast Simulation

Sofien Bouaziz* EPFL · Sebastian Martin† VM Research · Tiantian Liu‡ University of Pennsylvania · Ladislav Kavan§ University of Pennsylvania · Mark Pauly¶ EPFL

Figure 1: *We propose a new "projection-based" implicit Euler integrator that supports a large variety of geometric constraints in a single physical simulation framework. In this example, all the elements including building, grass, tree, and clothes (498 DoF's, 43k constraints), are simulated at 3 frame/second using 10 iterations per frame (see also accompanying video).*

Abstract

We present a new method for implicit time integration of physical systems. Our approach builds a bridge between nodal Finite Element methods and Position Based Dynamics, leading to a simple, efficient, robust, yet accurate solver that supports many different types of constraints. We propose specially designed energy potentials that can be solved efficiently using an alternating optimization approach. Inspired by continuum mechanics, we derive a set of continuum-based potentials that can be efficiently incorporated within our solver.

1 Introduction

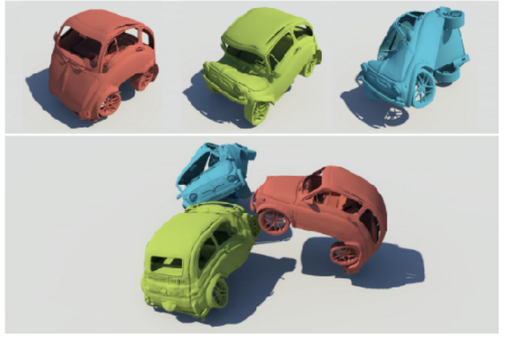
Physics-based simulation of deformable material has become an indispensable tool in many areas of computer graphics. Virtual worlds, and more recently character animations, incorporate sophisticated simulations to greatly enhance visual experience, e.g., by simulating muscles, fat, hair, clothing, or vegetation. These models are often based on finite element discretizations of continuum-mechanics formulations, allowing highly *accurate* simulation of complex nonlinear materials.

Projective Dynamics

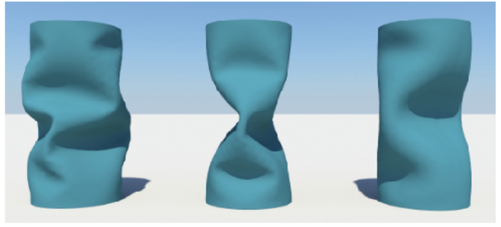
- Geometrically motivated models for deformables, shells, rods
- Constraints for strain limiting, joints and contacts
- Generalizations with ADMM and as Quasi-Newton method
- Compared to XPBD: better convergence properties but less general

Penetration-free Projective Dynamics on the GPU [Lan et al. 2022]

Example-based materials



Thin shells



[Bouaziz et al. 2014]

Vertex Block Descent

- Uses a block coordinate descent approach
- Solves **local** (per-vertex) energies:

$$E_i(\boldsymbol{x}) = \frac{m_i}{2\Delta t^2}\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|^2 + \sum_{j \in \mathcal{V}_i} \phi_j(\boldsymbol{x})$$

Energies affecting vertex i

- Analytic inversion of local **3x3 Hessians**
- Designed for efficient implementation on GPU
- Alternative to XPBD, but is a **primal** method
- Reduces global energy but does not necessarily converge to its minimum

Vertex Block Descent

ANKA HE CHEN, University of Utah, USA
ZIHENG LIU, University of Utah, USA
YIN YANG, University of Utah, USA
CEM YUKSEL, University of Utah, USA and Roblox, USA

Fig. 1. *Example simulation results using our solver, both of these methods involve more than 100 million DoFs and 1 million active collisions.*

We introduce vertex block descent, a block coordinate descent solution for the variational form of implicit Euler through vertex ... Gauss-Seidel iterations. It operates with local vertex ... tions in global variational energy with ...

1 INTRODUCTION

Physics-based simulation is the cornerstone of most graphics ap-

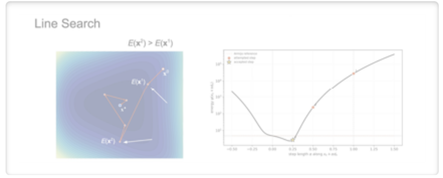
Augmented Vertex Block Descent

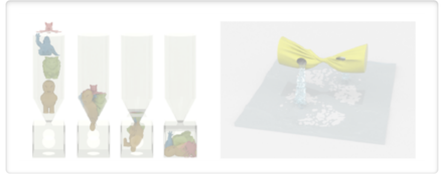
CHRIS GILES, Roblox, USA
ELIE DIAZ, University of Utah, USA
CEM YUKSEL, University of Utah, USA

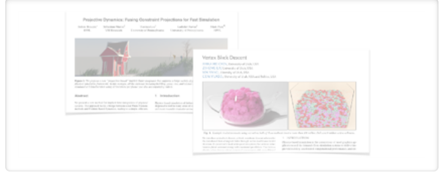
Fig. 1. *Our augmented vertex block descent method allows simulating hard constraints, which are critical for handling contacts and stacking, shown here simulating a pile of 110,000 blocks smashed by a sphere using 4 iterations, taking 9.5 ms (9.8 ms including collision detection) per frame on an NVIDIA RTX 4090 GPU.*

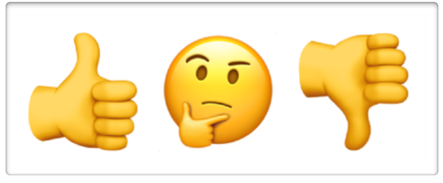
Vertex Block Descent is a fast physics-based simulation method that is unconditionally stable, highly parallelizable, and capable of converging to the implicit Euler solution. We extend it using an augmented Lagrangian formulation to address some of its fundamental limitations. First, we introduce a mechanism to handle hard constraints with infinite stiffness without convergence behavior over prior techniques for soft body dynamics, delivering higher performance, improved parallelism, unconditional stability, and the ability to converge to the implicit Euler solution with the desired accuracy. In addition, VBD was shown to handle other simulation problems, such as particle systems and rigid body

Outline









1. Mathematical foundations

2. Physical models and coupling

3. Related methods (VBD, PD)

4. Summary: Models & Properties

Summary: Energy-based Models

- 🌟 Deformable solids, cloth, shells and rods
  - 👍 Geometrically motivated & continuum-based models
  - 👍 Material models: isotropic, anisotropic, example-based, data-driven…
- 👍 Dynamic frictional contact & intersection-free simulations
- 👍 Rigid bodies & multi-body systems
- 👍 Other physical phenomena (e.g. magnetics, heating, wetting…)

- 🤔 “Accurate” friction and certain plasticity models
  - → Typically needs approximations (e.g. lagging)
- 🤔 Fluids & granular media
  - → Typically needs operator splitting, semi-implicit approaches or constrained optimization

Summary: Properties of Energy-Based Simulation

👍 **Simple:** in terms of formulation and numerical approach

👍 **Robust convergence:** due to line search & second-order derivatives

🤔 **Primal methods:** Good for high mass ratios, very large stiffness ratios numerically challenging

👎 **Limited to potentials:** no arbitrary dissipative and non-conservative forces

$$f(\boldsymbol{x}) = -\nabla_{\boldsymbol{x}}\phi(\boldsymbol{x})$$
or
$$f_D(\boldsymbol{v}) = -\nabla_{\boldsymbol{v}}\phi_D(\boldsymbol{v})$$

- 👉 Solution: Adapted and approximate models (for friction, damping and plasticity)

👎 **High computational cost:** problematic for interactive applications

- 👉 Alternative: Related methods for interactive applications, e.g.:
  - VBD: Vertex Block Descent [Chen et al. 2024]
  - PD: Projective Dynamics [Bouaziz et al. 2014]

Trade-offs in terms of convergence and/or generality