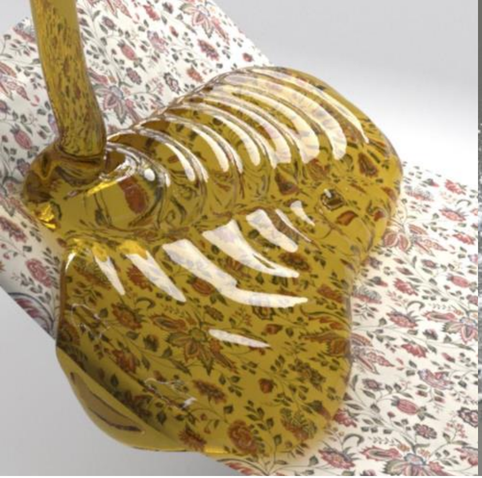
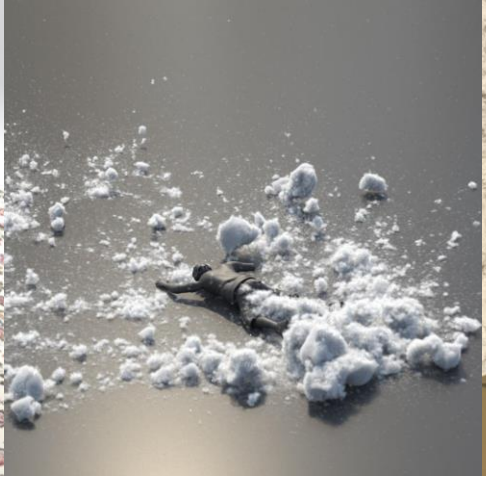



Multiphysics Simulation Methods in Computer Graphics

Daniel Holz[1,2], Stefan Rhys Jeske[3], Fabian Löschner[3], Jan Bender[3], Yin Yang[4], Sheldon Andrews[2]

Unity
ÉTS Engineering for Industry ÉCOLE DE TECHNOLOGIE SUPÉRIEURE Université du Québec
RWTH AACHEN UNIVERSITY
THE UNIVERSITY OF UTAH

Outline

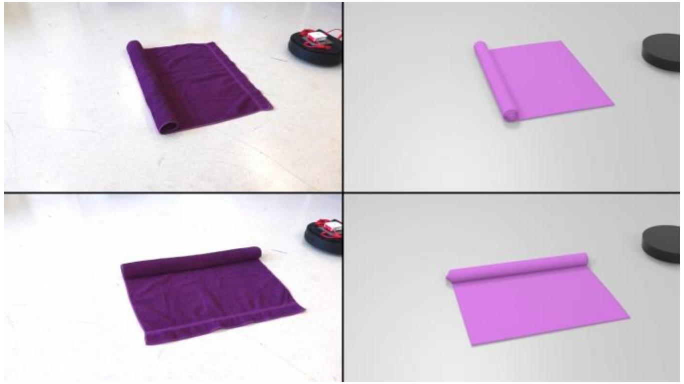
1. Introduction
2. Structure of the Talk Series
3. Part I: Constraint-Based Multiphysics Modeling
    1. A Unified Modeling Framework
    2. Position-Level Formulation
    3. Velocity-Level Formulation
4. Questions

Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling [Gissler et al., 2019]

Introduction

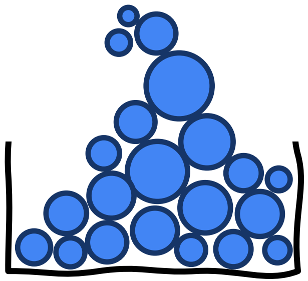
- Abundance of physics simulation methods
  - Physical materials
  - Physical phenomena
- Multiphysics simulation
  - **Interacting** materials and phenomena
- Applications in many industries
  - Entertainment
  - Engineering
  - Training

Snow simulation for the movie Frozen [Stomakhin et al., 2013]

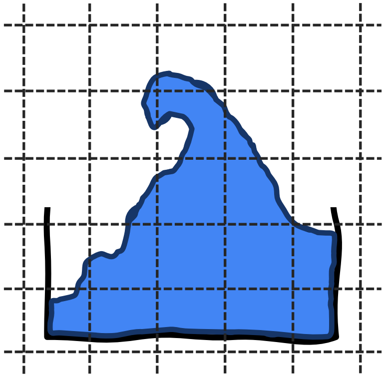

Robotic vacuum cleaner colliding with carpet [Fernandez-Fernandez et al., 2024]

Introduction

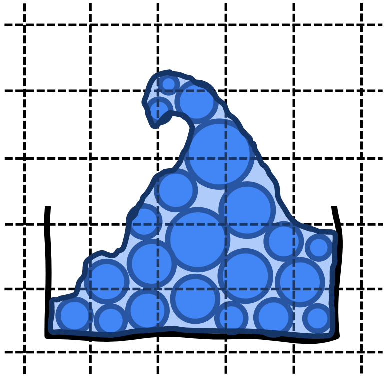
- How to classify multiphysics simulation methods?
  - **Domain discretization**



Lagrangian
(ex: particles)



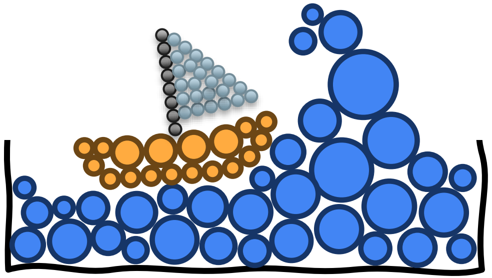
Eulerian
(ex: grid)



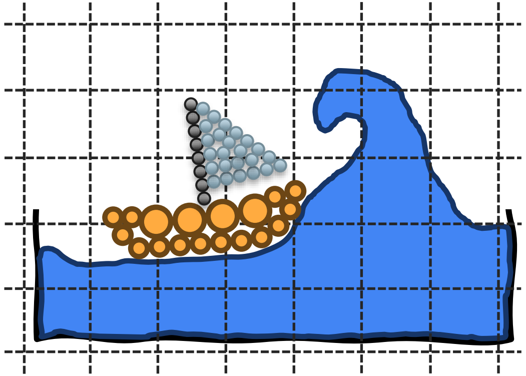
Hybrid

Introduction

- How to classify multiphysics simulation methods?
  - Domain discretization
  - **Behavior and interaction modeling**



Unified Models
Monolithic modeling frameworks, yielding strong two-way coupling



Coupling Techniques
Techniques for two-way coupling of multiple models or discretizations

Introduction

	Lagrangian Point-Based Methods (Sec. 2)	Eulerian & Hybrid Methods (Sec. 3)	Energy-Based Modeling (Sec. 4)	Constraint-Based Modeling (Sec. 5)
Deformables (elastic & plastic)	[MKN*04] [PKA*05] [SSP07] [BIT09] [MKB*10] [YJL*16] [YCL*17] [PGBT18] [CLC*20] [KBF*21] [KUKH23]	[SZS95] [CGFO06] [LLJ*11] [SSJ*14] [JSS*15] [YSB*15] [TLK16] [FGG*17] [GTJS17] [JGT17] [ZB17] [GHF*18] [HFG*18] [FLGJ19] [HGG*19] [SXH*21] [LLJ22] [TB22] [QLY*23] [LLH*24] [TLZ*24]	[BAV*10] [BUAG12] [SB12b] [SHST12] [BML*14] [GSS*15] [LBK17] [BOFN18] [SGK18] [LFS*20] [MEM*20] [LMY*22] [LCK22] [LLJ22] [KE22] [LFFJ*23]	[Jak01] [MHTG05] [MHHR06] [SLM06] [MMCK14] [BKCW14] [Cho14] [MCKM15] [CMM16] [DCB16] [MMC16] [BGAO17] [FM17] [ARM*19] [MEM*19] [WWB*19] [MMC*20] [MM21] [TTKA23] [CHC*24a] [Cet24] [MAK24] [SZDJ24] [YLL*24]
Granular Materials	[LD09] [AO11] [IWT13] [YJL*16] [YCL*17] [GHB*20]	[ZB05] [SSC*13] [DBD16] [KGP*16] [TGK*17] [GPH*18]		[Hol14] [MMCK14] [SWLB14] [FM17] [HG18] [NS18] [KKHS20] [YLL*24]
Rigid Bodies & Multibody Systems	[SSP07] [YCL*17] [GPB*19] [PT23]	[TB20] [TB22] [LLH*24] [TLZ*24]	[CDGB19] [MEM*20] [FLS*21] [CLL*22] [LKL*22]	[Bar94] [MC95] [ST96] [Bar96] [AP97] [Ste00] [Jak01] [Erl05] [MHTG05] [Lac07b, Lac07a] [GZO10] [MMCK14] [DCB16] [FM17] [MEM*19] [PAK*19] [WWB*19] [MMC*20] [MAK24]
Co-dimensional Structures	[MKB*10] [ZQC*14] [ZLQF15]	[JGT17] [GHF*18] [HGG*19] [LLH*24]	[GHDS03] [ST07] [BWR*08] [CSvRV18] [Kim20] [LKJ21] [CXY*23] [HB23] [SWP*23] [WB23] [LFFJB24]	[Jak01] [MHHR06] [GHF*07] [SL08] [SLNB10] [MKC12] [BKCW14] [MMCK14] [USS15] [MMC16] [KS16] [DKWB18] [ARM*19]

⋮

Unified Models
Monolithic modeling frameworks, yielding strong two-way coupling

⋮

	Lagrangian Point-Based Methods (Sec. 2)	Eulerian & Hybrid Methods (Sec. 3)	Energy-Based Modeling (Sec. 4)	Constraint-Based Modeling (Sec. 5)
Fluids & Fluid Phenomena	[PW02] [MCG03] [SSP07] [BT07] [BIT09] [SP09] [Pri12] [SB12a] [AAT13] [ICS*14] [HWZ*15] [TDF*15] [BK17] [PT17] [YCL*17] [YML*17] [PGBT18] [WKBB18] [BKKW19] [CBG*19] [GPB*19] [WJL*20] [ZRS*20] [KBF*21] [LWB*21] [WDK*21] [LHWW22] [XRW*22] [JWL*23] [PT23] [XLYJ23] [ZLX*24] [YWX*24]	[Har62] [HW*65] [BR86] [FM96] [Sta99] [Pes02] [TUKF02] [CMT04] [ZB05] [CGFO06] [KFCO06] [CFL*07] [MCP*09] [SABS14] [SSJ*14] [ATW15] [JSS*15] [RGJ*15] [FGG*17] [GPH*18] [HFG*18] [JGT17] [ZB17] [FLGJ19] [GAB20] [HGMRT20] [TB20] [CKMR*21] [SXH*21] [QLDGJ22] [TB22] [STBA24] [QLY*23] [LLH*24] [TLZ*24]	[TB20] [TB21] [TB22] [XLYJ23]	[BLS12] [MM13] [MMCK14] [TNF14] [BGAO17] [XRW*22] [YLL*24]
Multi-Phase, Phase Transitions & Porous Flow	[MKN*04] [SSP07] [LAD08] [SP08] [BIT09] [LD09] [PC13] [RLY*14] [YCR*15] [YJL*16] [PGBT18] [CLC*20] [GHB*20] [WFM21] [RXL21] [RHLC22] [XWW*23] [YR23] [ZLX*24]	[SSJ*14] [ATW15] [GPH*18] [GAB20] [CKMR*21] [SXH*21] [LMLD22] [TLZ*24]		[MMCK14]
Other Phenomena	[LL10] [Pri12]	[WFL*19] [WDG*19] [SNZ*21] [FCK22] [CCL*22]	[CSvRV18] [CNZ*22] [WFFJB24]	[GZO10] [Cho14] [BCK*22]

Introduction

Unified Models

Monolithic modeling frameworks, yielding strong two-way coupling

	Lagrangian Point-Based Methods (Sec. 2)	Eulerian & Hybrid Methods (Sec. 3)	Energy-Based Modeling (Sec. 4)	Constraint-Based Modeling (Sec. 5)
Deformables (elastic & plastic)	[MKN*04] [PKA*05] [SSP07] [BIT09] [MKB*10] [YJL*16] [YCL*17] [PGBT18] [CLC*20] [KBF*21] [KUKH23]	[SZS95] [CGFO06] [LLJ*11] [SSJ*14] [JSS*15] [YSB*15] [TLK16] [FGG*17] [GTJS17] [JGT17] [ZB17] [GHF*18] [HFG*18] [FLGJ19] [HGG*19] [SXH*21] [LLJ22] [TB22] [QLY*23] [LLH*24] [TLZ*24]	[BAV*10] [BUAG12] [SB12b] [SHST12] [BML*14] [GSS*15] [LBK17] [BOFN18] [SGK18] [LFS*20] [MEM*20] [LMY*22] [LCK22] [LLJ22] [KE22] [LFFJ*23]	[Jak01] [MHTG05] [MHHR06] [SLM06] [MMCK14] [BKCW14] [Cho14] [MCKM15] [CMM16] [DCB16] [MMC16] [BGAO17] [FM17] [ARM*19] [MEM*19] [WWB*19] [MMC*20] [MM21] [TTKA23] [CHC*24a] [Cet24] [MAK24] [SZDJ24] [YLL*24]
Granular Materials	[LD09] [AO11] [IWT13] [YJL*16] [YCL*17] [GHB*20]	[ZB05] [SSC*13] [DBD16] [KGP*16] [TGK*17] [GPH*18]		[Hol14] [MMCK14] [SWLB14] [FM17] [HG18] [NS18] [KKHS20] [YLL*24]
Rigid Bodies & Multibody Systems	[SSP07] [YCL*17] [GPB*19] [PT23]	[TB20] [TB22] [LLH*24] [TLZ*24]	[CDGB19] [MEM*20] [FLS*21] [CLL*22] [LKL*22]	[Bar94] [MC95] [ST96] [Bar96] [AP97] [Ste00] [Jak01] [Erl05] [MHTG05] [Lac07b, Lac07a] [GZO10] [MMCK14] [DCB16] [FM17] [MEM*19] [PAK*19] [WWB*19] [MMC*20] [MAK24]
Co-dimensional Structures	[MKB*10] [ZQC*14] [ZLQF15]	[JGT17] [GHF*18] [HGG*19] [LLH*24]	[GHDS03] [ST07] [BWR*08] [CSvRV18] [Kim20] [LKJ21] [CXY*23] [HB23] [SWP*23] [WB23] [LFFJB24]	[Jak01] [MHHR06] [GHF*07] [SL08] [SLNB10] [MKC12] [BKCW14] [MMCK14] [USS15] [MMC16] [KS16] [DKWB18] [ARM*19]
Fluids & Fluid Phenomena	[PW02] [MCG03] [SSP07] [BT07] [BIT09] [SP09] [Pri12] [SB12a] [AAT13] [ICS*14] [HWZ*15] [TDF*15] [BK17] [PT17] [YCL*17] [YML*17] [PGBT18] [WKBB18] [BKKW19] [CBG*19] [GPB*19] [WJL*20] [ZRS*20] [KBF*21] [LWB*21] [WDK*21] [LHWW22] [XRW*22] [JWL*23] [PT23] [XLYJ23] [ZLX*24] [YWX*24]	[Har62] [HW*65] [BR86] [FM96] [Sta99] [Pes02] [TUKF02] [CMT04] [ZB05] [CGFO06] [KFCO06] [CFL*07] [MCP*09] [SABS14] [SSJ*14] [ATW15] [JSS*15] [RGJ*15] [FGG*17] [GPH*18] [HFG*18] [JGT17] [ZB17] [FLGJ19] [GAB20] [HGMRT20] [TB20] [CKMR*21] [SXH*21] [QLDGJ22] [TB22] [STBA24] [QLY*23] [LLH*24] [TLZ*24]	[TB20] [TB21] [TB22] [XLYJ23]	[BLS12] [MM13] [MMCK14] [TNF14] [BGAO17] [XRW*22] [YLL*24]
Multi-Phase, Phase Transitions & Porous Flow	[MKN*04] [SSP07] [LAD08] [SP08] [BIT09] [LD09] [PC13] [RLY*14] [YCR*15] [YJL*16] [PGBT18] [CLC*20] [GHB*20] [WFM21] [RXL21] [RHLC22] [XWW*23] [YR23] [ZLX*24]	[SSJ*14] [ATW15] [GPH*18] [GAB20] [CKMR*21] [SXH*21] [LMLD22] [TLZ*24]		[MMCK14]
Other Phenomena	[LL10] [Pri12]	[WFL*19] [WDG*19] [SNZ*21] [FCK22] [CCL*22]	[CSvRV18] [CNZ*22] [WFFJB24]	[GZO10] [Cho14] [BCK*22]

Introduction

	Rigid Bodies	Granular Materials	Fluids	Rods & Shells
Deformable Solids	[MHHR06] [HFG*18] [DS19] [HGG*19] [HGG*19] [BKWK20] [TB22] [LLH*24]	[HGG*19] [LLH*24]	[GSLF05] [CGFO06] [BBB07] [RMSG*08] [RMSG*08] [ACAT13] [JSS*15] [ZB17] [SLZ17] [ANZS18] [TBFL19] [ARNM*20] [CCL*22] [TB22] [XLYJ23] [LLH*24]	[LLH*24]
Rigid Bodies	-	[AO11] [IWT13] [HFG*18] [DS19] [HGG*19] [LLH*24]	[FM96] [Pes02] [TUKF02] [GSLF05] [RZF05] [KFCO06] [BBB07] [RMSG*08] [RMEF09] [BTT09] [AIA*12] [ATW15] [FM15] [KB17] [ANZS18] [BGI*18] [BGPT18] [HFG*18] [TBFL19] [BKWK20] [WAK20] [LCD*20] [LLDL21] [TB20] [TB22] [BWRJ23] [SZYA24]	[MHHR06] [HGG*19]
Granular Materials	+	-	[GPH*18] [WFM21]	[GHF*18] [HGG*19] [LLH*24]
Fluids	+	+	-	[GSLF05] [ANZS18] [LLH*24]

Coupling Techniques

Techniques for two-way coupling of multiple models or discretizations

Introduction

	Rigid Bodies	Granular Materials	Fluids	Rods & Shells
Deformable Solids	[MHHR06] [HFG*18] [DS19] [HGG*19] [HGG*19] [BKWK20] [TB22] [LLH*24]	[HGG*19] [LLH*24]	[GSLF05] [CGFO06] [BBB07] [RMSG*08] [RMSG*08] [ACAT13] [JSS*15] [ZB17] [SLZ17] [ANZS18] [TBFL19] [ARNM*20] [CCL*22] [TB22] [XLYJ23] [LLH*24]	[LLH*24]
Rigid Bodies	-	[AO11] [IWT13] [HFG*18] [DS19] [HGG*19] [LLH*24]	[FM96] [Pes02] [TUKF02] [GSLF05] [RZF05] [KFCO06] [BBB07] [RMSG*08] [RMEF09] [BTT09] [AIA*12] [ATW15] [FM15] [KB17] [ANZS18] [BGI*18] [BGPT18] [HFG*18] [TBFL19] [BKWK20] [WAK20] [LCD*20] [LLDL21] [TB20] [TB22] [BWRJ23] [SZYA24]	[MHHR06] [HGG*19]
Granular Materials	+	-	[GPH*18] [WFM21]	[GHF*18] [HGG*19] [LLH*24]
Fluids	+	+	-	[GSLF05] [ANZS18] [LLH*24]

Coupling Techniques
Techniques for two-way coupling of multiple models or discretizations

Introduction

	Rigid Bodies	Granular Materials	Fluids	Rods & Shells
Deformable Solids	[MHHR06] [HFG*18] [DS19] [HGG*19] [HGG*19] [BKWK20] [TB22] [LLH*24]	[HGG*19] [LLH*24]	[GSLF05] [CGFO06] [BBB07] [RMSG*08] [RMSG*08] [ACAT13] [JSS*15] [ZB17] [SLZ17] [ANZS18] [TBFL19] [ARNM*20] [CCL*22] [TB22] [XLYJ23] [LLH*24]	[LLH*24]
Rigid Bodies	-	[AO11] [IWT13] [HFG*18] [DS19] [HGG*19] [LLH*24]	[FM96] [Pes02] [TUKF02] [GSLF05] [RZF05] [KFCO06] [BBB07] [RMSG*08] [RMEF09] [BTT09] [AIA*12] [ATW15] [FM15] [KB17] [ANZS18] [BGI*18] [BGPT18] [HFG*18] [TBFL19] [BKWK20] [WAK20] [LCD*20] [LLDL21] [TB20] [TB22] [BWRJ23] [SZYA24]	[MHHR06] [HGG*19]
Granular Materials	+	-	[GPH*18] [WFM21]	[GHF*18] [HGG*19] [LLH*24]
Fluids	+	+	-	[GSLF05] [ANZS18] [LLH*24]

Coupling Techniques

Techniques for two-way coupling of multiple models or discretizations

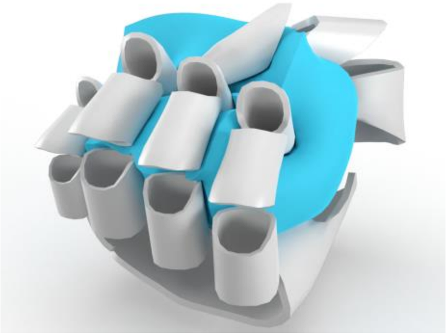
Structure of the Talk Series

- **Part I: Constraint-Based Multiphysics Modeling**
    - Flexible, Lagrangian domain discretization
    - Behavior and interactions modeled using constraint functions

Robotic hand grasping an elastic cube
[Fernandez-Fernandez et al., 2024]

Structure of the Talk Series

- Part I: Constraint-Based Multiphysics Modeling
- **Part II: Energy-Based Multiphysics Modeling**
  - Equally flexible domain discretization
  - Behavior and interactions modeled using energy functions

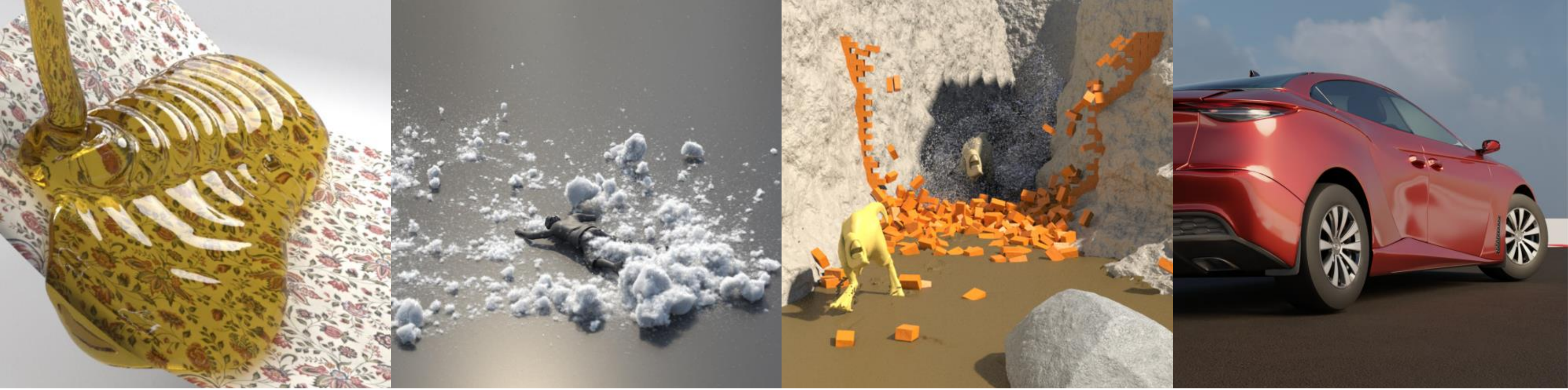


Robotic hand grasping an elastic cube
[Fernandez-Fernandez et al., 2024]

Structure of the Talk Series

- Part I: Constraint-Based Multiphysics Modeling
- Part II: Energy-Based Multiphysics Modeling
- **Part III: Lagrangian Point-Based, Eulerian and Hybrid Methods**
  - Lagrangian, Eulerian and Hybrid discretizations
  - Simulated domain treated as a continuum

Robotic hand grasping an elastic cube
[Fernandez-Fernandez et al., 2024]



Part I
Constraint-Based Multiphysics Modeling

Daniel Holz[1,2], Stefan Rhys Jeske[3], Fabian Löschner[3], Jan Bender[3], Yin Yang[4], Sheldon Andrews[2]

Unity

ÉTS Engineering for Industry ÉCOLE DE TECHNOLOGIE SUPÉRIEURE Université du Québec

RWTH AACHEN UNIVERSITY

THE UNIVERSITY OF UTAH

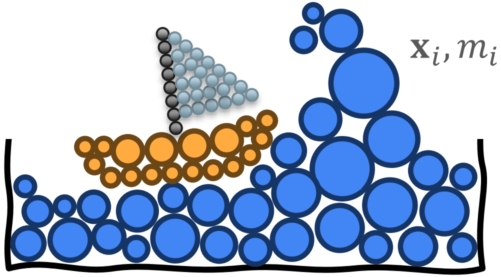
Outline

A Unified Modeling Framework

- Newton's 2nd law of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$\mathbf{x} = [x_1, \dots, x_n]^T$ Generalized coordinates

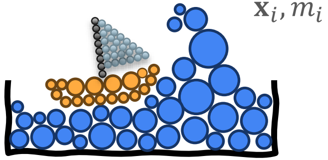


A Unified Modeling Framework

- Newton's 2nd law of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$$\mathbf{x} = [x_1, \ldots, x_n]^T$$ Generalized coordinates



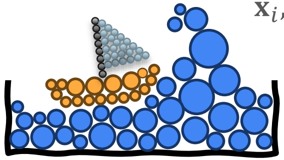
- **Constraints** as **unified modeling tool**
  - Idea: directly model the desired behavior

A Unified Modeling Framework

- Newton's 2nd law of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

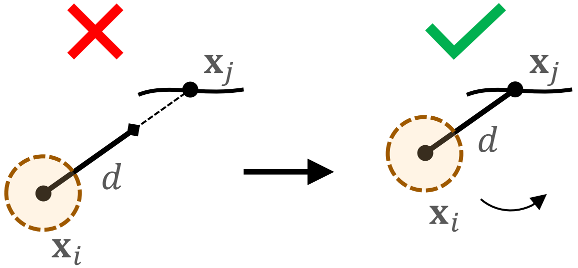
$\mathbf{x} = [x_1, \dots, x_n]^T$ Generalized coordinates



- **Constraints** as **unified modeling tool**
  - Idea: directly model the desired behavior

$\mathbf{C}_b(\mathbf{x}) \boxed{= \mathbf{0}}$ **Bilateral** constraints

$\mathbf{C} = [C_1(\mathbf{x}), \dots, C_k(\mathbf{x})]^T$



$C_b(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\| - d \boxed{= 0}$

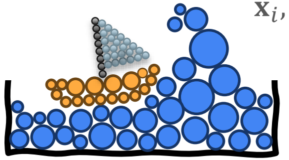
Distance constraint

A Unified Modeling Framework

- Newton's 2nd law of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$\mathbf{x} = [x_1, \ldots, x_n]^T$ Generalized coordinates

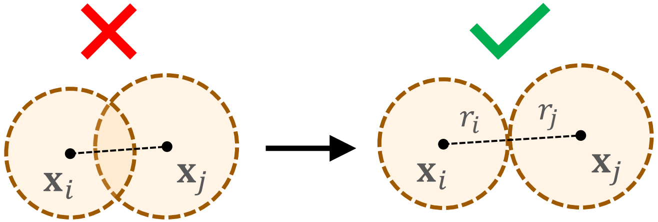


- **Constraints** as **unified modeling tool**
  - Idea: directly model the desired behavior

$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$ **Bilateral** constraints

$\mathbf{C}_u(\mathbf{x}) \boxed{\geq \mathbf{0}}$ **Unilateral** constraints

$$\mathbf{C} = [C_1(\mathbf{x}), \ldots, C_k(\mathbf{x})]^T$$



$$C_u(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\| - (r_i + r_j) \boxed{\geq 0}$$

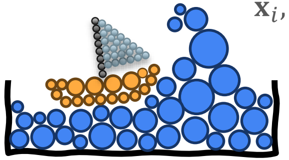
Contact constraint

A Unified Modeling Framework

- Newton's 2nd law of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$\mathbf{x} = [x_1, \dots, x_n]^T$ Generalized coordinates

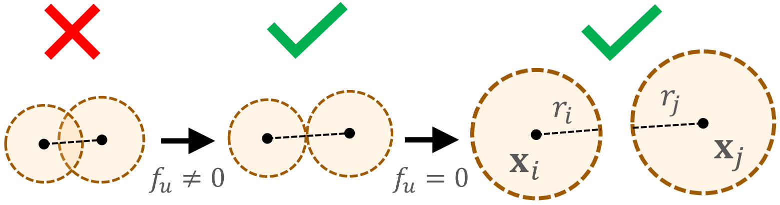


- **Constraints** as **unified modeling tool**
  - Idea: directly model the desired behavior

$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$ **Bilateral** constraints

$\mathbf{C}_u(\mathbf{x}) \boxed{\geq \mathbf{0}}$ **Unilateral** constraints

$$\mathbf{C} = [C_1(\mathbf{x}), \dots, C_k(\mathbf{x})]^T$$



$$C_u(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\| - (r_i + r_j) \boxed{\geq 0}$$

Contact constraint

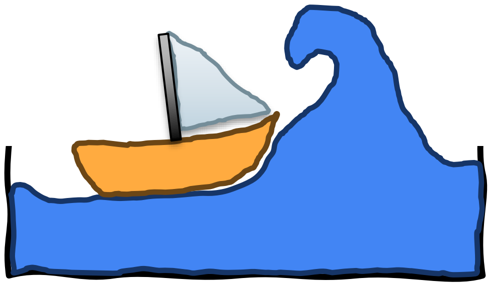
A Unified Modeling Framework: Discretization

1) **Discretization** of equations of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

Equations of motion

$$\xrightarrow[\textbf{?}]{\hspace{3cm}}$$



Simulated material domains

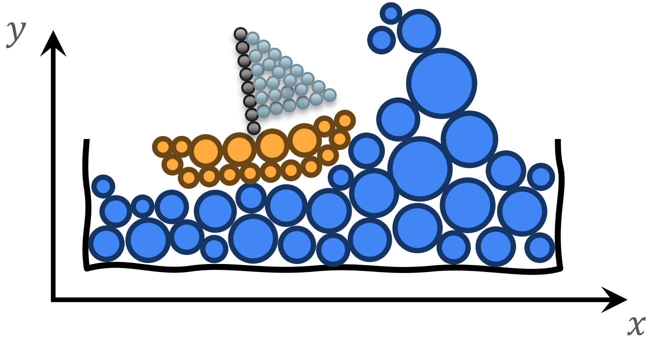
A Unified Modeling Framework: Discretization

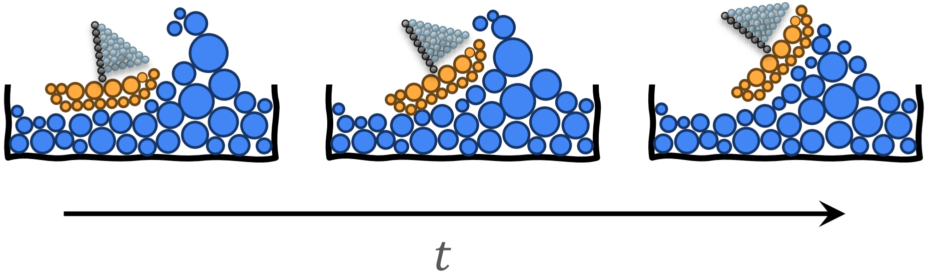
1) **Discretization** of equations of motion

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

Equations of motion

Discretize in **space** and **time**



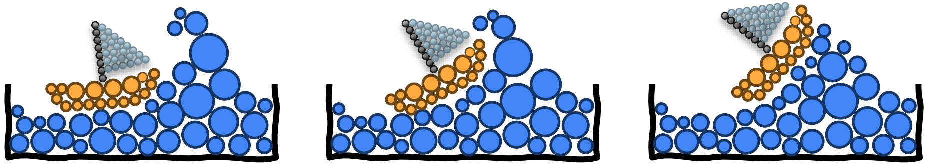


A Unified Modeling Framework: Discretization

1) Discretization of equations of motion

- Two popular approaches
  - **Position-level formulation:** Position Based Dynamics
  - **Velocity-level formulation:** Nonsmooth Multidomain Dynamics
- Very stable, even under coarse discretizations and large time steps
  - Good choice for **real-time** and **interactive** applications

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$



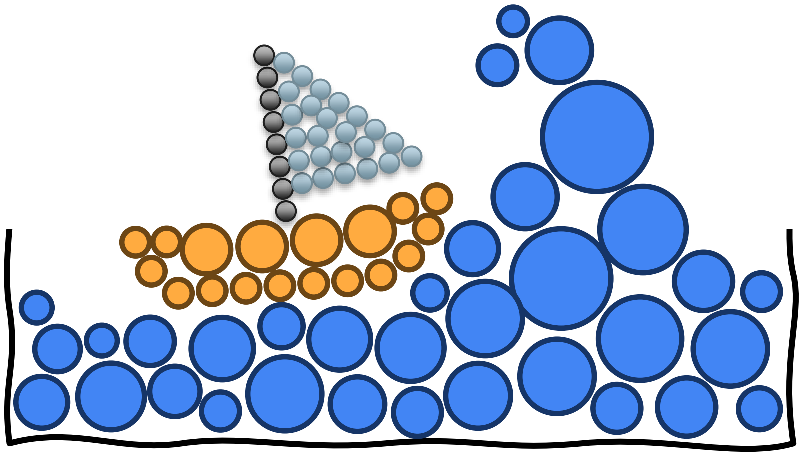
A Unified Modeling Framework: Multiphysics Modeling

2) Multiphysics Modeling

- What do we need?
- Constraint function definition

$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

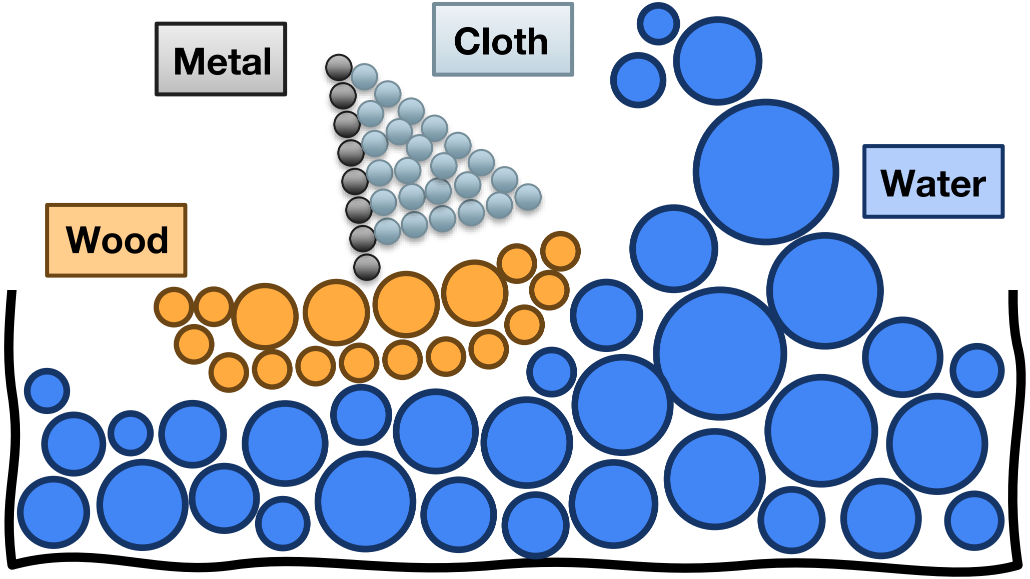
$\xrightarrow{?}$



Constraint functions

Multiple material domains in interaction

A Unified Modeling Framework



Multiple material domains

$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

$\longrightarrow$

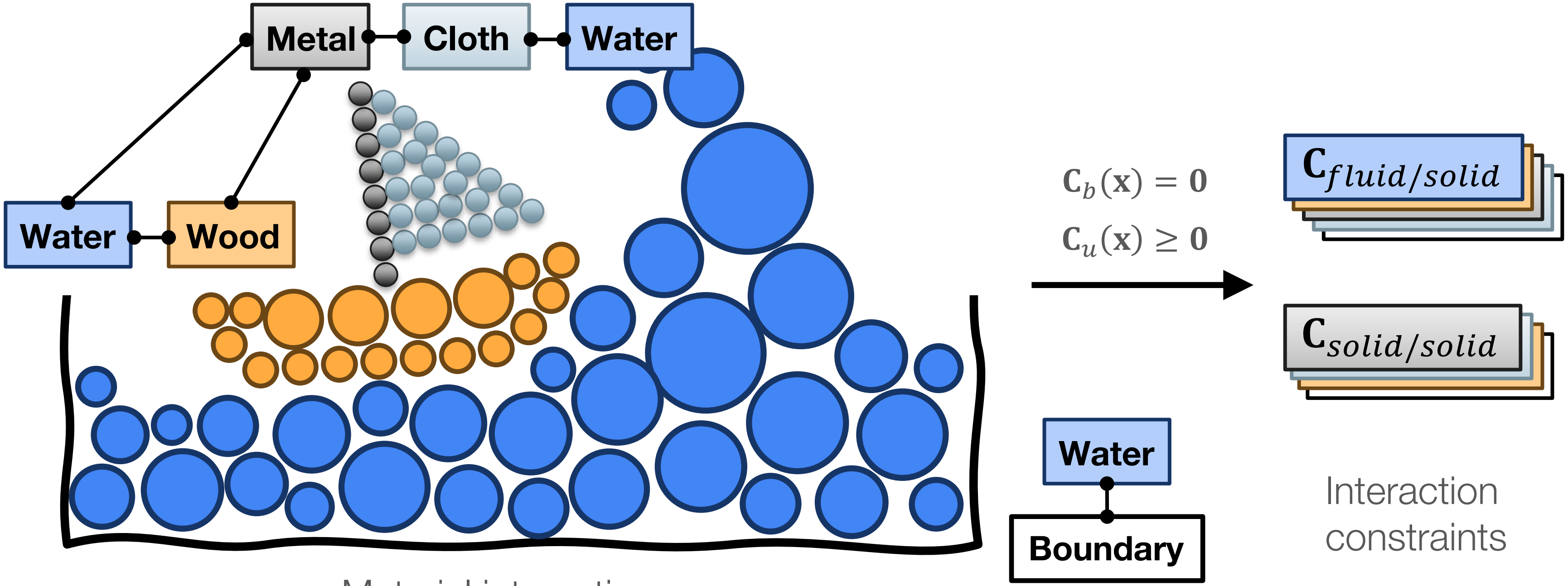
$\mathbf{C}_{fluid}$

$\mathbf{C}_{elastoplastic}$

$\mathbf{C}_{viscoelastic}$

Material constraints

A Unified Modeling Framework



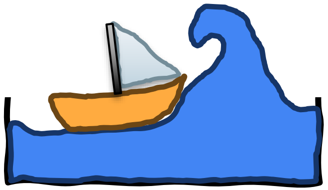
Material interactions

Interaction constraints

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

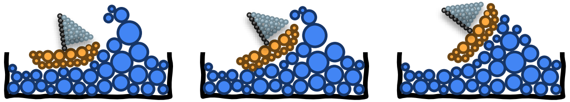
Equations of motion



Simulated material domains

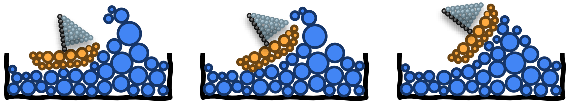
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

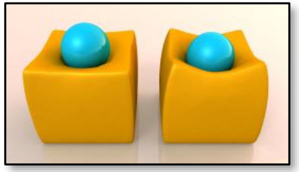


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



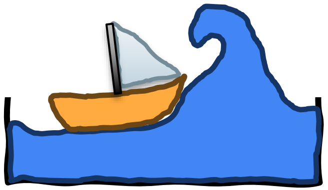
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

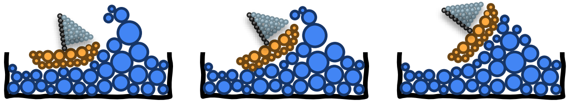
Equations of motion



Simulated material domains

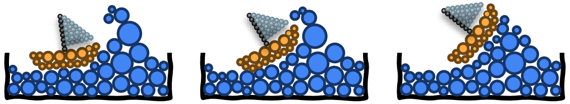
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

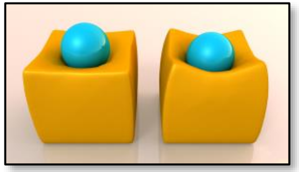


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



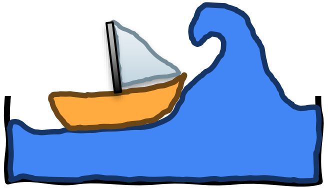
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

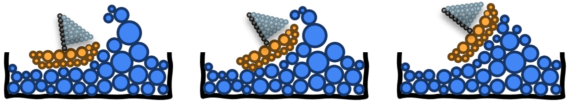
Equations of motion



Simulated material domains

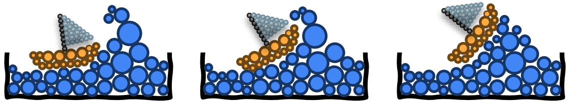
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

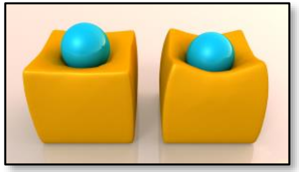


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies

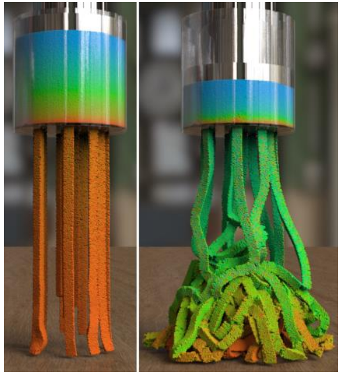


Deformables

Granular Materials

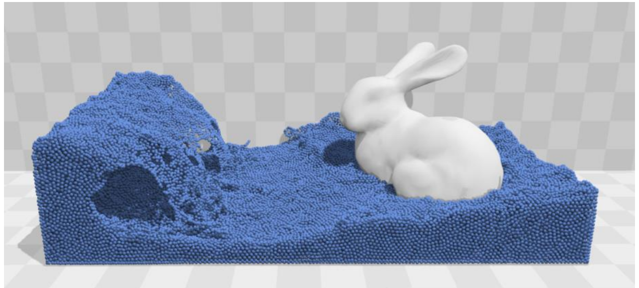
Position Based Dynamics

Rigid car with flexible tires [Müller et al., 2020]

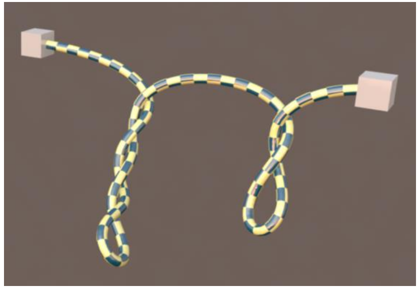


Plastic modeling clay [Yu et al., 2024]

Deformed cloth [Bender et al., 2014]



Position Based Fluids [Macklin and Müller, 2013]



Rod forming plectonemes [Deul et al., 2018]

Cream [Barreiro et al., 2017]

Position Based Dynamics: Discretization

- Predictor-Corrector scheme [Müller et al., 2006]
  1. **Predict**: Predict **unconstrained** positions
  2. **Correct**: Correct **constrained** positions in solver
- Different **solver** types:
  - Iterative, e.g., Gauss-Seidel, or Jacobi
  - Direct [Goldenthal et al., 2007; Deul et al., 2018]

1. Prediction with forward Euler:

$$\mathbf{p}_i = \mathbf{x}_i + \Delta t \mathbf{v}_i + \Delta t^2 \frac{\mathbf{f}(\mathbf{x}_i)}{m_i}$$

2. Solver:

$$\Delta \mathbf{p} = \lambda \mathbf{M}^{-1} \nabla C(\mathbf{p})$$

$$\lambda = \frac{-C(\mathbf{p})}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})}$$

Correction along constraint gradient

Position Based Dynamics: Discretization

- Predictor-Corrector scheme [Müller et al., 2006]
  1. **Predict**: Predict **unconstrained** positions
  2. **Correct**: Correct **constrained** positions in solver
- Different **solver** types:
  - Iterative, e.g., Gauss-Seidel, or Jacobi
  - Direct [Goldenthal et al., 2007; Deul et al., 2018]

1. Prediction with forward Euler:

$$\mathbf{p}_i = \mathbf{x}_i + \Delta t \mathbf{v}_i + \Delta t^2 \frac{\mathbf{f}(\mathbf{x}_i)}{m_i}$$

2. Solver:

$$\Delta \mathbf{p} = \boxed{\lambda} \mathbf{M}^{-1} \boxed{\nabla C(\mathbf{p})}$$

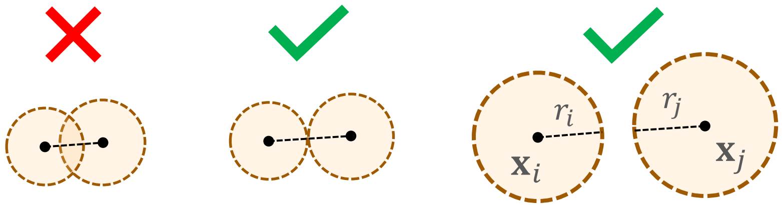
$$\lambda = \frac{\boxed{-C(\mathbf{p})}}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})}$$

Correction along constraint gradient

Need: Constraint error and gradient

Position Based Dynamics: Discretization

- Predictor-Corrector scheme [Müller et al., 2006]
  1. **Predict**: Predict **unconstrained** positions
  2. **Correct**: Correct **constrained** positions in solver



$\lambda_u > 0$ $\quad$ $\lambda_u = 0$ $\quad$ $\lambda_u = 0$

$$C_u(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\| - (r_i + r_j) \boxed{\geq 0} \rightarrow \boxed{\lambda_u = \max(0, \lambda)}$$

Cap correction for **unilateral constraints**

1. Prediction with forward Euler:

$$\mathbf{p}_i = \mathbf{x}_i + \Delta t \mathbf{v}_i + \Delta t^2 \frac{\mathbf{f}(\mathbf{x}_i)}{m_i}$$

2. Solver:

$$\Delta \mathbf{p} = \lambda \mathbf{M}^{-1} \nabla C(\mathbf{p})$$

$$\boxed{\lambda} = \frac{-C(\mathbf{p})}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})}$$

Correction along constraint gradient

Position Based Dynamics: Discretization

- Predictor-Corrector scheme [Müller et al., 2006]
  1. **Predict**: Predict **unconstrained** positions
  2. **Correct**: Correct **constrained** positions in solver
- Different **solver** types:
  - Iterative, e.g., Gauss-Seidel, or Jacobi
  - Direct [Goldenthal et al., 2007; Deul et al., 2018]
- **3. Update velocity** from position change
  - No extrapolation, yields high stability
  - But some limitations (more later)

1. Prediction with forward Euler:

$$\mathbf{p}_i = \mathbf{x}_i + \Delta t \mathbf{v}_i + \Delta t^2 \frac{\mathbf{f}(\mathbf{x}_i)}{m_i}$$

2. Solver:

$$\Delta \mathbf{p} = \lambda \mathbf{M}^{-1} \nabla C(\mathbf{p})$$

$$\lambda = \frac{-C(\mathbf{p})}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})}$$

Correction along constraint gradient

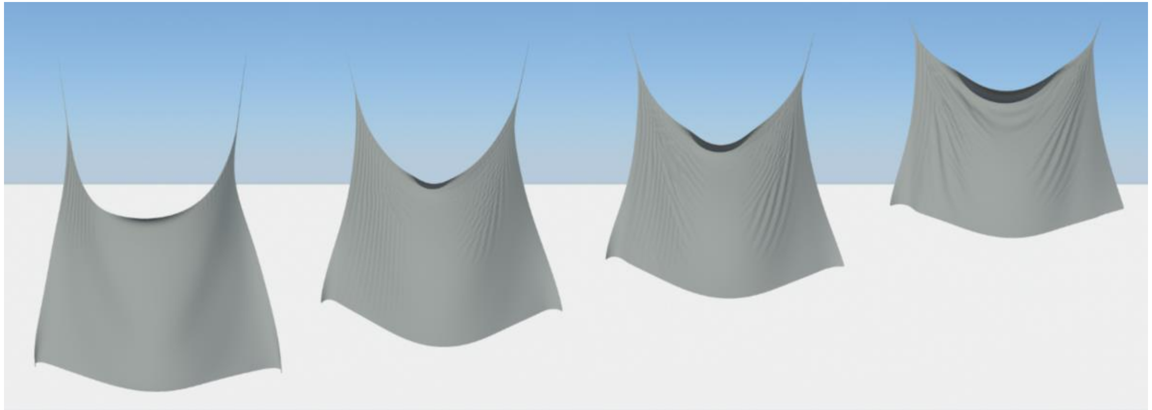
3. Update velocity and position:

$$\mathbf{v}_i = \frac{\mathbf{p}_i - \mathbf{x}_i}{\Delta t}, \qquad \mathbf{x}_i = \mathbf{p}_i$$

Position Based Dynamics: Discretization

Limitations of original PBD

1. **Time-step and iteration dependence**
2. **Resolution dependence**

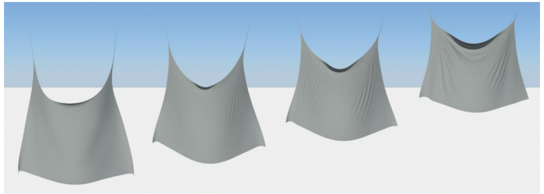


Increased stiffness with larger iteration count in original PBD [Bouaziz et al., 2014]

Position Based Dynamics: Discretization

Limitations of original PBD

1. **Time-step and iteration dependence**
2. Resolution dependence



Bouaziz et al., 2014

1. **overcome by eXtended PBD (XPBD)** [Macklin et al., 2016]
   - Derived from elastic energy potential U [Servin et al., 2006]

Elastic energy:

$$U(\mathbf{p}) = \frac{1}{2} C(\mathbf{p})^T \boxed{\alpha^{-1}} C(\mathbf{p})$$

α: compliance

Original PBD:

$$\lambda = \frac{-C(\mathbf{p})}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p})} \longrightarrow$$

XPBD:

$$\Delta\lambda = \frac{-C(\mathbf{p}) \boxed{- \alpha/\Delta t^2\, \lambda}}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p}) \boxed{+ \alpha/\Delta t^2}}$$

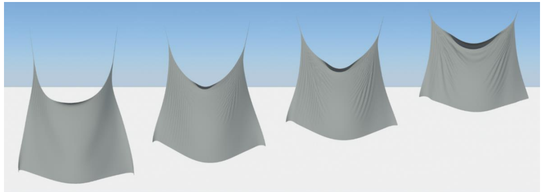
$$\lambda := \lambda + \Delta\lambda$$

Position Based Dynamics: Discretization

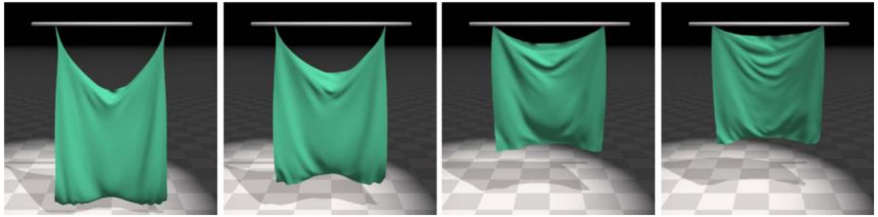
Limitations of original PBD

1. **Time-step and iteration dependence**
2. Resolution dependence

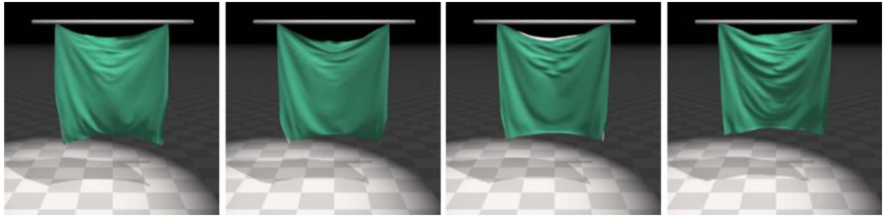
1. **overcome by eXtended PBD (XPBD)** [Macklin et al., 2016]
   - Derived from elastic energy potential U [Servin et al., 2006]



Bouaziz et al., 2014



Original PBD



XPBD

Macklin et al., 2016

Position Based Dynamics: Discretization

Limitations of original PBD

1. Time-step and iteration dependence
2. **Resolution dependence**

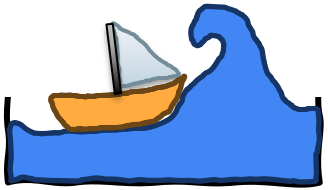
1. overcome by eXtended PBD (XPBD) [Macklin et al., 2016]
2. **overcome by Bender et al. (2014)**
   - Continuous, deformable materials
   - Tetrahedral volume elements (FEM-style)
   - Position-level constraint of the strain energy

Bender et al., 2014

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

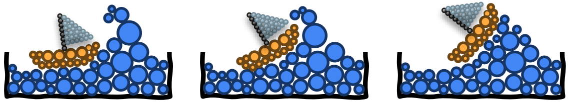
Equations of motion



Simulated material domains

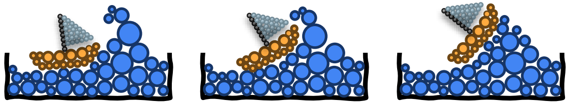
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

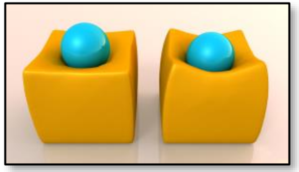


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



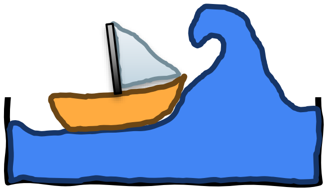
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

Equations of motion

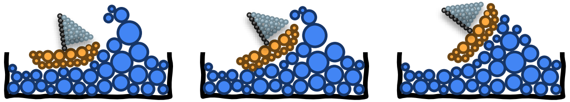


Simulated material domains

---

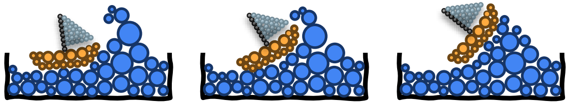
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



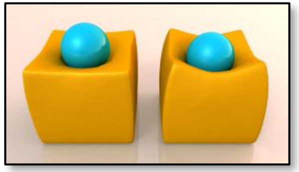
Velocity-level formulation

---

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



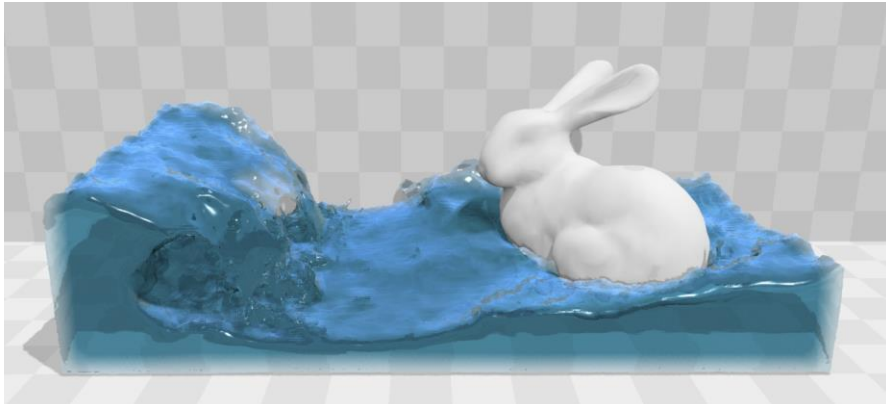
Deformables

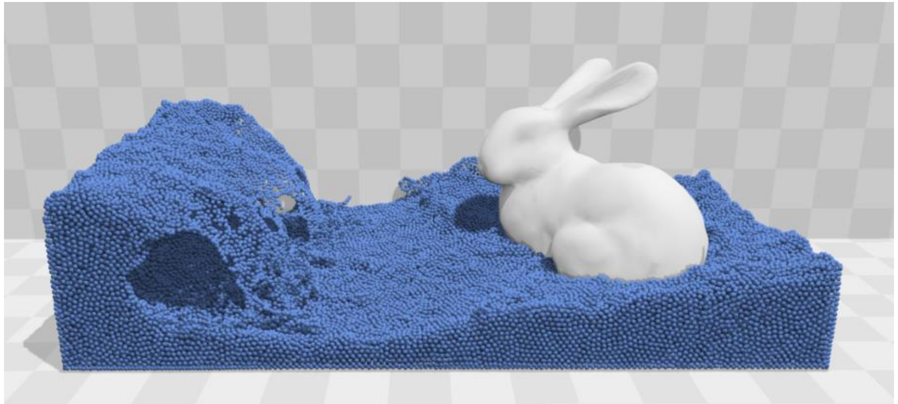
Granular Materials

Position Based Fluids [Macklin and Müller, 2013]

- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$

$$C_i = \frac{\rho_i}{\rho_0} - 1 \stackrel{!}{=} 0$$





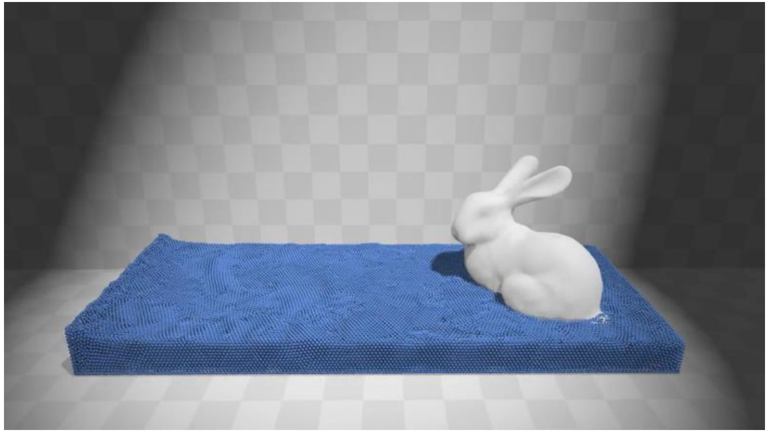
Real-time Position Based Fluid simulation [Macklin and Müller, 2013]

Position Based Fluids [Macklin and Müller, 2013]

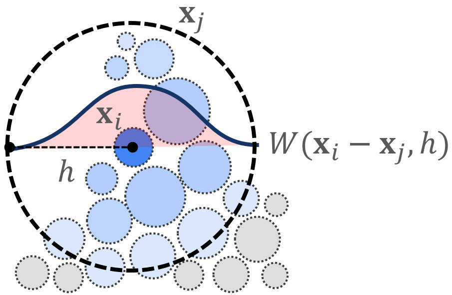
- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- SPH-based material interpolation
  - Smoothing kernel W with radius h

$$C_i = \frac{\rho_i}{\rho_0} - 1 \overset{!}{=} 0$$

$$\rho_i = \sum_{j=1}^{n} m_j \, W(\mathbf{x}_i - \mathbf{x}_j, h)$$



Macklin and Müller, 2013



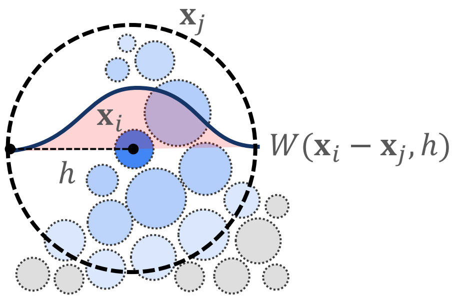
Position Based Fluids [Macklin and Müller, 2013]

- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- SPH-based material interpolation
  - Smoothing kernel W with radius h
- Constraint gradient for inclusion in PBD

$$C_i = \frac{\rho_i}{\rho_0} - 1 \stackrel{!}{=} 0$$

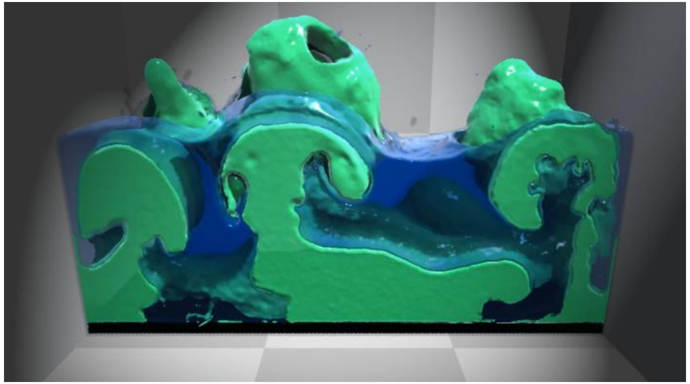
$$\rho_i = \sum_{j=1}^{n} m_j W(\mathbf{x}_i - \mathbf{x}_j, h)$$

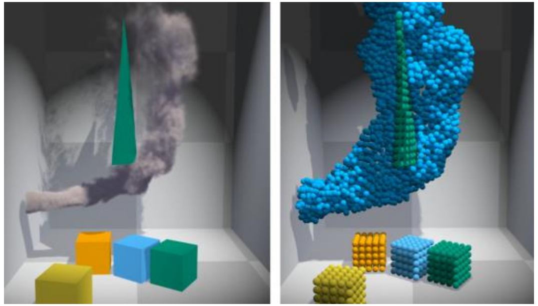
$$\boxed{\nabla C_i} = \frac{1}{\rho_0} \sum_{j=1}^{n} m_j \boxed{\nabla W}(\mathbf{x}_i - \mathbf{x}_j, h)$$



Position Based Fluids [Macklin and Müller, 2013]

- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- Extensions
  - **Multi-phase fluids & smoke**
    [Macklin et al., 2014]

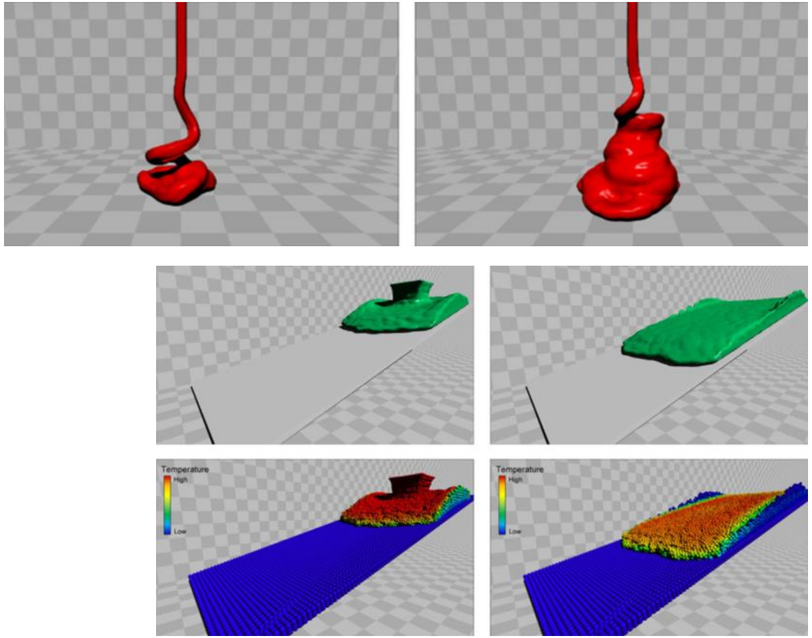




Macklin et al., 2014

Position Based Fluids [Macklin and Müller, 2013]

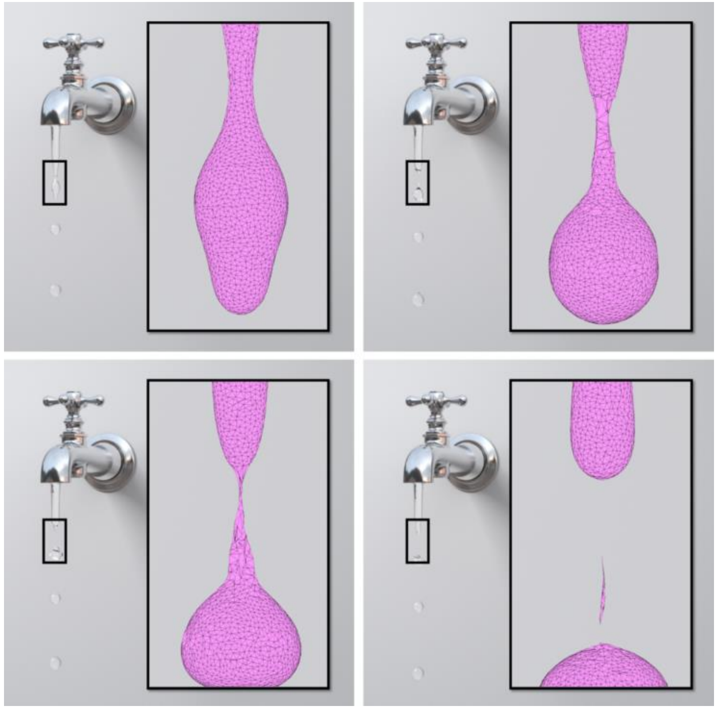
- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- Extensions
  - Multi-phase fluids & smoke [Macklin et al., 2014]
  - **Viscous fluids & phase transitions** [Takahashi et al., 2014]



Takahashi et al., 2014

Position Based Fluids [Macklin and Müller, 2013]

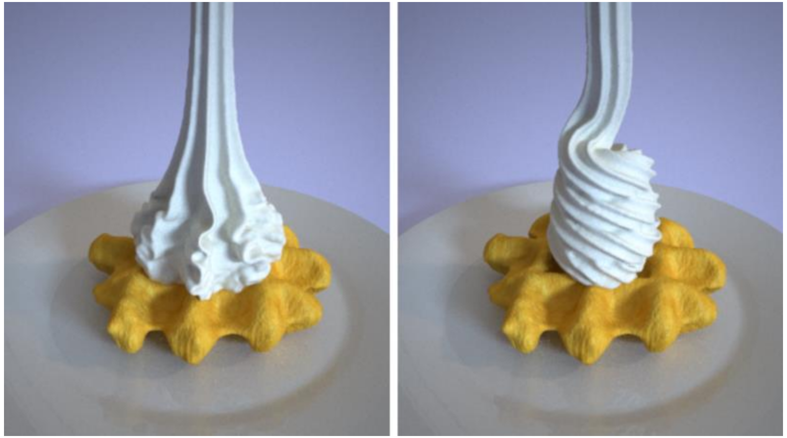
- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- Extensions
  - Multi-phase fluids & smoke [Macklin et al., 2014]
  - Viscous fluids & phase transitions [Takahashi et al., 2014]
  - **Strong surface tension** [Xing et al., 2022]



Xing et al., 2022

Position Based Fluids [Macklin and Müller, 2013]

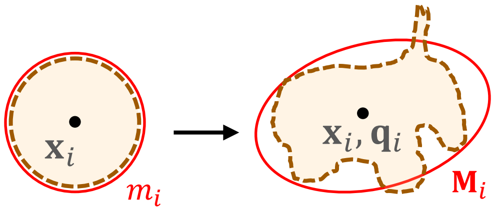
- Density constraint C_i [Bodin et al., 2012]
  - Models incompressible fluid
  - Enforces uniform reference density ρ_0 at $\mathbf{x}_i$
- Extensions
  - Multi-phase fluids & smoke [Macklin et al., 2014]
  - Viscous fluids & phase transitions [Takahashi et al., 2014]
  - Strong surface tension [Xing et al., 2022]
  - **Viscoelastic and elastoplastic fluids** [Barreiro et al., 2017]
    - Hybrid velocity-based / position-based solver
    - Enforces both **position-level** and **velocity-level** constraints



Barreiro et al., 2017

Position Based Rigid Bodies

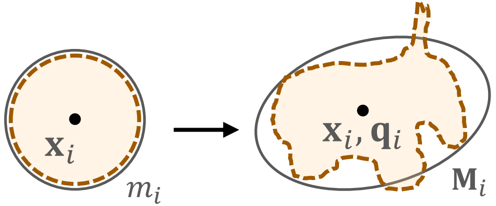
- PBD extended to rigid bodies [Deul et al., 2014]
  - Introduce rotations $\mathbf{q}_i$
  - Scalar particle mass m_i becomes rigid body mass matrix $\mathbf{M}_i$
  - Efficient representation

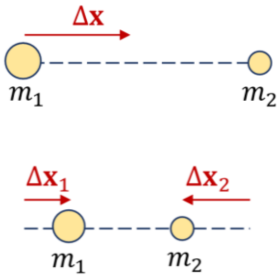


Rigid elk collides with rubber ducky [Deul et al., 2014]

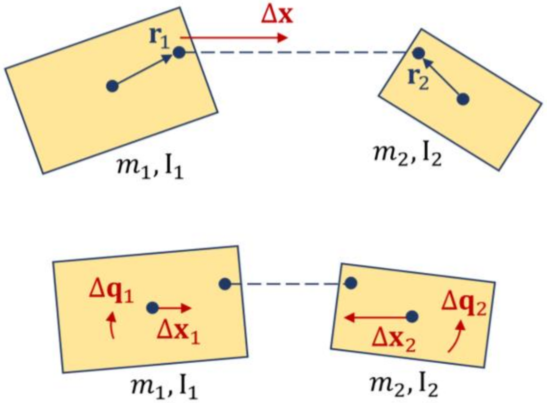
Position Based Rigid Bodies

- PBD extended to rigid bodies [Deul et al., 2014]
  - Introduce rotations $\mathbf{q}_i$
  - Scalar particle mass m_i becomes rigid body mass matrix $\mathbf{M}_i$
  - Efficient representation



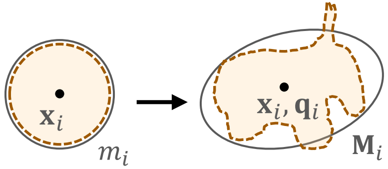


Correction of **particles**

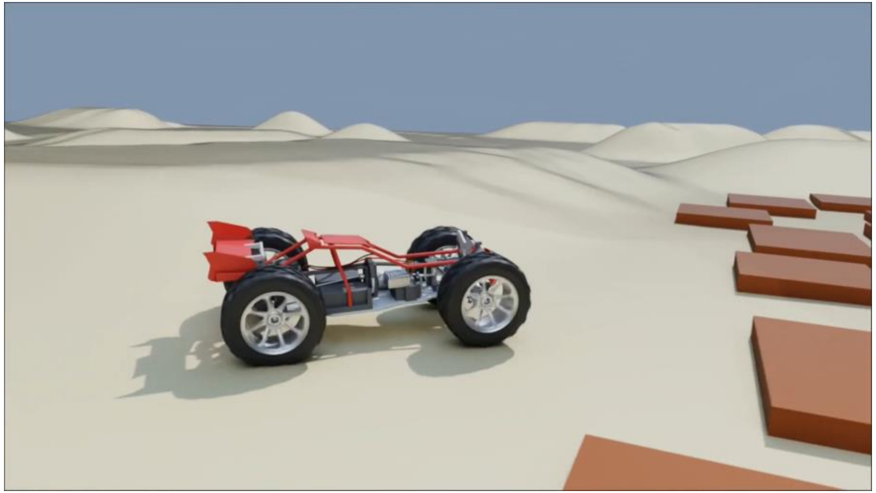


Correction of **rigid bodies** [Müller et al., 2020]

Position Based Rigid Bodies



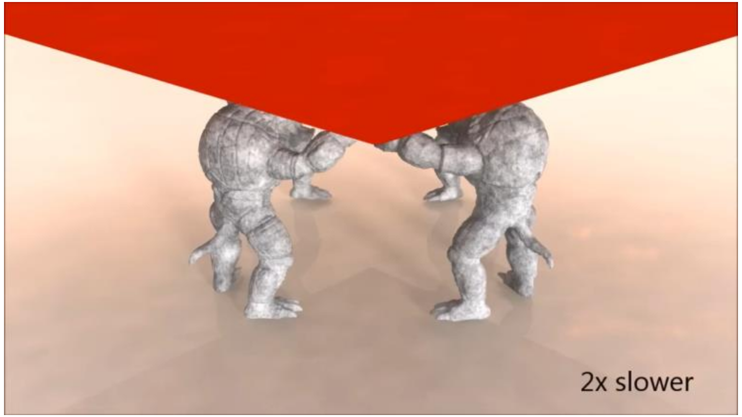
- PBD extended to rigid bodies [Deul et al., 2014]
- Compatible with PBD particles
  - Strong two-way coupling
  - Shows flexibility of PBD discretization
- Joints and friction [Deul et al., 2014]
- Restitution [Müller et al., 2020]
  - Via additional **velocity-level** solve



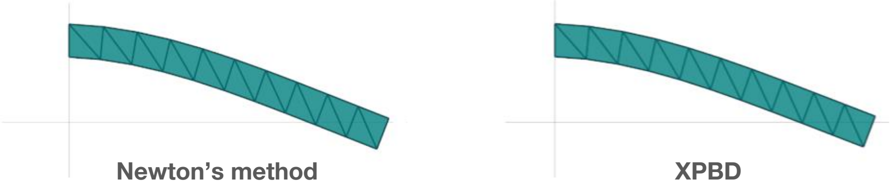
Rigid car with deformable tires drives over obstacles [Müller et al., 2020]

Position Based Deformables

- Mostly tetrahedral/triangular discretization
- Deformable material constraints
  - Constrain **strain energy** [Bender et al., 2014]
  - Constrain **strain tensor** [Müller et al., 2015]
  - Constraint derived from **elastic energy potential** [Macklin et al., 2016; Francu et al., 2017]
  - **Shape matching** [Chentanez et al., 2016]



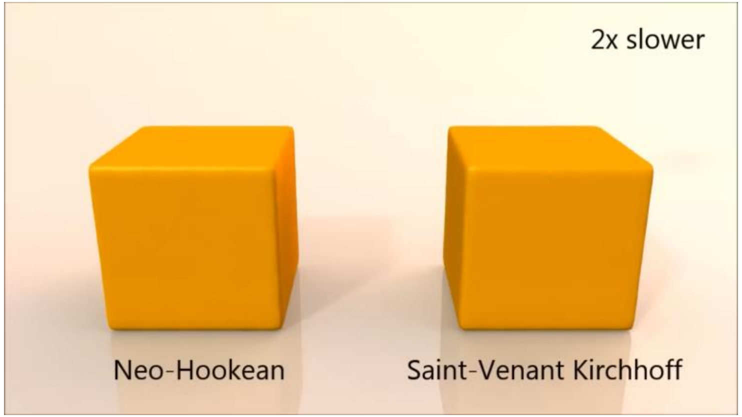
Cloth simulation using strain energy constraint [Bender et al., 2014]



Hookean, isotropic cantilever beam [Macklin et al., 2016]

Position Based Deformables

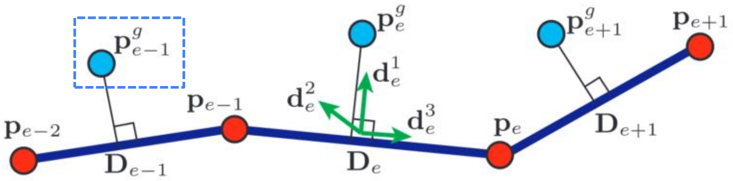
- Mostly tetrahedral/triangular discretization
- Deformable material constraints
- **Support for…**
  - Hookean materials
    [Bender et al., 2014; Macklin et al., 2016; Francu et al., 2017]
  - Hyperelastic materials [Bender et al., 2014;
    Macklin and Müller, 2021; Ton-That et al., 2022; Chen et al., 2024]
  - Higher-order finite elements
    [Saillant et al., 2024]



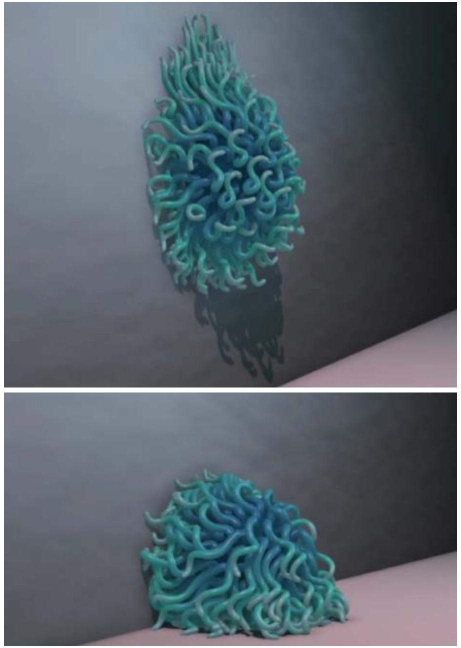
Hyperelastic materials under compression [Bender et al., 2014]

Position Based Rods

- Co-dimensional structure
  - Ideal for hair, fur, vegetation, cables, ropes, …
- Cosserat rod theory in PBD [Umetani et al., 2015]
  - Elastic rods with stretching, bending and torsion
  - Consistent material frames through added **ghost points**



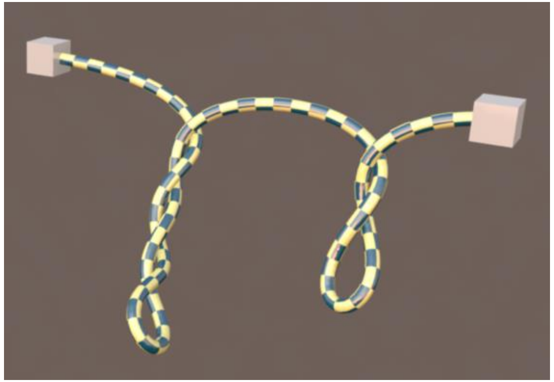
Ghost points (cyan) for defining rod material frames [Umetani et al., 2015]



Squishy ball hits a wall [Umetani et al., 2015]

Position Based Rods

- Co-dimensional structure
  - Ideal for hair, fur, vegetation, cables, ropes, …
- Cosserat rod theory in PBD [Umetani et al., 2015]
- **Extensions**
  - Material frames via quaternions [Kugelstadt and Schömer, 2016]
  - High material stiffness via direct solver [Deul et al., 2018]
  - Volumetric deformations with volume conservation [Angles et al., 2019]



Twisted rod forming plectonemes [Deul et al., 2018]

Muscles modeled using volumetric rods [Angles et al., 2019]

Position Based Granular Materials

- Two common approaches
  - DEM-based & Continuum-based
- DEM-based
  - PBD soil particles
  - Particle interaction constraints
    - Viscoelastic collisions [Holz, 2014; Macklin et al., 2016; Francu et al., 2017]
    - Coulomb friction [Macklin et al., 2014; Holz, 2014]
    - Cohesion [Holz and Galarneau, 2018]

Excavator digging in clay type soil [Holz and Galarneau, 2018]

Position Based Granular Materials

- Continuum-based
  - Plasticity theory in PBD [Yu et al., 2024]
    - **eXtended Position-Based Inelasticity (XPBI)**
  - Viscoplastic and elastoplastic materials
    - Strongly coupled sand, water, snow, metal, and foam
  - Employs a hybrid grid-particle discretization (cf. MPM)
    - Well-suited for large deformations and fracture
  - **Challenge:**
    - Needs accurate estimates of deformation gradient $\boxed{\mathbf{F}^+}$
    - Depends on future velocity gradient $\boxed{\nabla\mathbf{v}^+}$
    - **Approach:** Reformulate XPBD on the **velocity-level**
      - Next slide…

Sandcastle washed away by water in XPBI [Yu et al., 2024]

$$\boxed{\mathbf{F}^+} = \mathbf{F} + \Delta t \boxed{\nabla\mathbf{v}^+} \mathbf{F}$$

Calculation of future deformation gradient $\mathbf{F}^+$

XPBD vs. XPBI (velocity-level)

XPBD:

1. Predict position:

$$\mathbf{p} = \mathbf{x} + \Delta t \mathbf{v} + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}(\mathbf{x})$$

2. Correct positions iteratively in solver:

$$\Delta\lambda = \frac{-C(\mathbf{p}) - \alpha/\Delta t^2 \, \lambda}{\nabla C(\mathbf{p})^T \mathbf{M}^{-1} \nabla C(\mathbf{p}) + \alpha/\Delta t^2}$$

$$\lambda \leftarrow \lambda + \Delta\lambda$$

$$\Delta\mathbf{p} = \mathbf{M}^{-1} \nabla C(\mathbf{p}) \Delta\lambda$$

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$$

3. Update velocity:

$$\mathbf{v} \leftarrow (\mathbf{p} - \mathbf{x})/\Delta t$$

$$\mathbf{x} \leftarrow \mathbf{p}$$

XPBD vs. XPBI (velocity-level)

XPBD:

1. Predict position:

$$\boxed{\mathbf{p}} = \mathbf{x} + \Delta t\mathbf{v} + \Delta t^2\mathbf{M}^{-1}\mathbf{f}(\mathbf{x})$$

2. Correct positions iteratively in solver:

$$\Delta\lambda = \frac{-C(\boxed{\mathbf{p}}) - \alpha/\Delta t^2\,\lambda}{\nabla C(\mathbf{p})^T\mathbf{M}^{-1}\nabla C(\mathbf{p}) + \alpha/\Delta t^2}$$

$$\lambda \leftarrow \lambda + \Delta\lambda$$

$$\Delta\mathbf{p} = \mathbf{M}^{-1}\nabla C(\boxed{\mathbf{p}})\Delta\lambda$$

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$$

3. Update velocity from new position:

$$\mathbf{v} \leftarrow (\mathbf{p} - \mathbf{x})/\Delta t$$

$$\mathbf{x} \leftarrow \mathbf{p}$$

XPBI (velocity-level XPBD):

1. Predict velocity:

$$\boxed{\mathbf{v}} \leftarrow \mathbf{v} + \Delta t\mathbf{M}^{-1}\mathbf{f}(\mathbf{x})$$

2. Correct velocities iteratively in solver:

$$\Delta\lambda = \frac{-C(\boxed{\mathbf{x} + \Delta t\mathbf{v}}) - \alpha/\Delta t^2\lambda}{\nabla C(\mathbf{x} + \Delta t\mathbf{v})^T\mathbf{M}^{-1}\nabla C(\mathbf{x} + \Delta t\mathbf{v}) + \alpha/\Delta t^2}$$

$$\lambda \leftarrow \lambda + \Delta\lambda$$

$$\Delta\mathbf{v} = \boxed{\frac{1}{\Delta t}}\mathbf{M}^{-1}\nabla C(\boxed{\mathbf{x} + \Delta t\mathbf{v}})\Delta\lambda$$

$$\mathbf{v} \leftarrow \mathbf{v} + \Delta\mathbf{v}$$

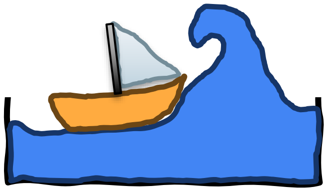
3. Update position from new velocity:

$$\mathbf{x} \leftarrow \mathbf{x} + \Delta t\mathbf{v}$$

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

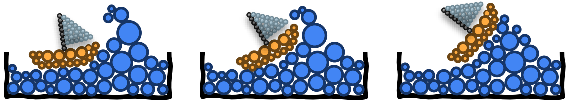
Equations of motion



Simulated material domains

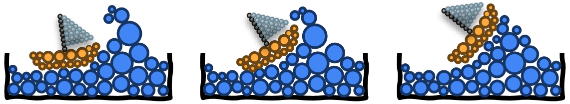
1) Discretization of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

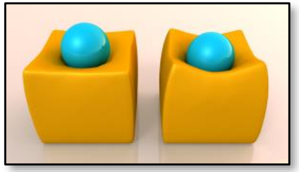


Velocity-level formulation

2) Multiphysics modeling via **constraints**

Fluids

Rigid Bodies



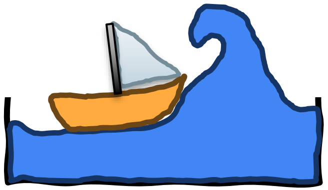
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

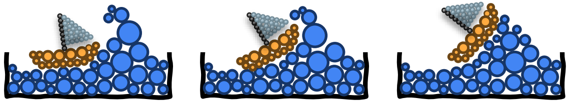
Equations of motion



Simulated material domains

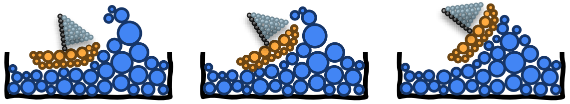
1) Discretization of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

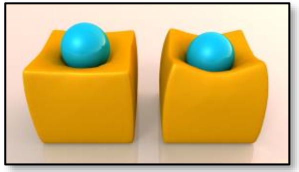


Velocity-level formulation

2) Multiphysics modeling via **constraints**

Fluids

Rigid Bodies



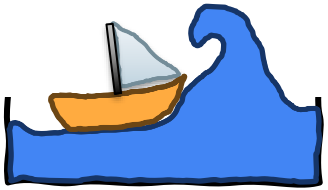
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

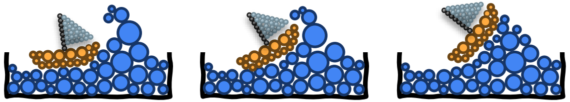
Equations of motion



Simulated material domains

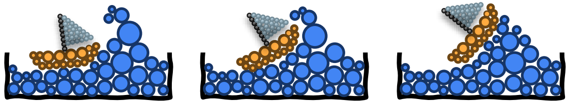
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

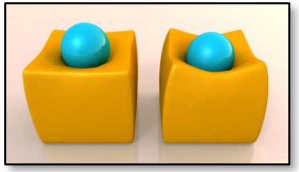


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



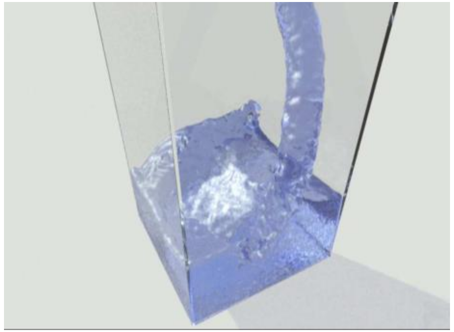
Deformables

Granular Materials

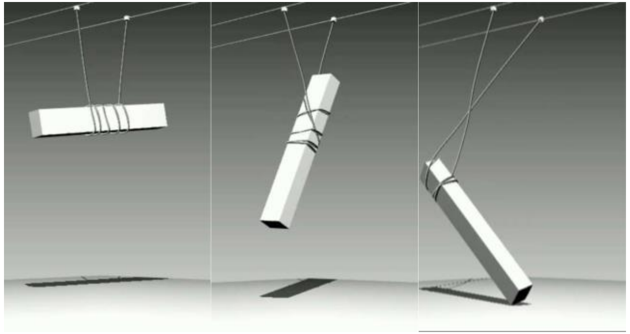
Velocity-Level Formulation

Adhesive jello [Gascon et al., 2010]

Chain with heavy weight [Andrews et al., 2022]



Constraint Fluids [Bodin et al., 2012]



Rods wrapped around rigid beam [Servin et al., 2010]

Constraint-based suction [Bernardin et al., 2022]

Velocity-Level Formulation: Nonsmooth Dynamics

- Nonsmooth, rigid body dynamics modeling [Moreau, 1963; Moreau, 1985]
  - Well-suited for hard, unilateral interactions, e.g., contacts
- Applied in numerous popular rigid body dynamics methods [Bender et al., 2014]

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

Equations of motion

$\longrightarrow$

$$\mathbf{M}\ddot{\mathbf{x}} \approx \mathbf{M}\frac{\boxed{\mathbf{v}^+} - \mathbf{v}}{\Delta t} = \mathbf{f}_{\text{ext}} + \mathbf{f}_c$$

$$\nabla\mathbf{C}_b^T\mathbf{v}^+ = \mathbf{0}$$
$$\nabla\mathbf{C}_u^T\mathbf{v}^+ \geq \mathbf{0}$$

Equations of motion on **velocity level**

Velocity-Level Formulation: Nonsmooth Dynamics

- Nonsmooth, rigid body dynamics modeling [Moreau, 1963; Moreau, 1985]
  - Well-suited for hard, unilateral interactions, e.g., contacts
- Applied in numerous popular rigid body dynamics methods [Bender et al., 2014]
  - Lagrange multiplier method
  - Linear Complementarity Problem (LCP) formulation

$$\left.\begin{aligned} \mathbf{M}\frac{\mathbf{v}^+ - \mathbf{v}}{\Delta t} &= \mathbf{f}_{\text{ext}} + \boxed{\mathbf{f}_c} \\ \nabla\mathbf{C}_b^T\mathbf{v}^+ &= \mathbf{0} \\ \nabla\mathbf{C}_u^T\mathbf{v}^+ &\boxed{\geq} \mathbf{0} \end{aligned}\right\} \xrightarrow{} \left\{\begin{aligned} &\begin{bmatrix} \mathbf{M} & -\nabla\mathbf{C}_b & -\nabla\mathbf{C}_u \\ \nabla\mathbf{C}_b^T & \mathbf{0} & \mathbf{0} \\ \nabla\mathbf{C}_u^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ \\ \boldsymbol{\lambda}_b \\ \boxed{\boldsymbol{\lambda}_u} \end{bmatrix} - \begin{bmatrix} \mathbf{M}\mathbf{v} + \Delta t\mathbf{f}_{\text{ext}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boxed{\mathbf{w}} \end{bmatrix} \\ &\boxed{\mathbf{0} \leq \mathbf{w} \perp \boldsymbol{\lambda}_u \geq \mathbf{0}} \end{aligned}\right.$$

$$\boxed{\Delta t\mathbf{f}_c = -\nabla\mathbf{C}_b\boldsymbol{\lambda}_b - \nabla\mathbf{C}_u\boldsymbol{\lambda}_u}$$

Constraint impulses as **Lagrange multipliers**

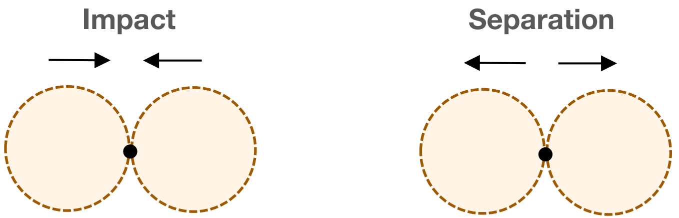
Equations of motion on **velocity level**

Mixed Linear **Complementarity** Problem (MLCP)

Velocity-Level Formulation: Nonsmooth Dynamics

- Nonsmooth, rigid body dynamics modeling [Moreau, 1963; Moreau, 1985]
  - Well-suited for hard, unilateral interactions, e.g., contacts
- Applied in numerous popular rigid body dynamics methods [Bender et al., 2014]

Contact example:



$w = 0 \wedge \boxed{\lambda_u > 0}$ $\qquad$ $\boxed{w > 0} \wedge \lambda_u = 0$

Unilateral constraints through **complementarity**

Slack variable: **constraint velocity**

$$\begin{cases} \begin{bmatrix} \mathbf{M} & -\nabla\mathbf{C}_b & -\nabla\mathbf{C}_u \\ \nabla\mathbf{C}_b^T & \mathbf{0} & \mathbf{0} \\ \nabla\mathbf{C}_u^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ \\ \boldsymbol{\lambda}_b \\ \boxed{\boldsymbol{\lambda}_u} \end{bmatrix} - \begin{bmatrix} \mathbf{M}\mathbf{v} + \Delta t\mathbf{f}_{\text{ext}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boxed{\mathbf{w}} \end{bmatrix} \\ \boxed{\mathbf{0} \le \mathbf{w} \perp \boldsymbol{\lambda}_u \ge \mathbf{0}} \end{cases}$$

Mixed Linear **Complementarity** Problem (MLCP)

Velocity-Level Formulation: Nonsmooth Dynamics

- Nonsmooth, rigid body dynamics modeling [Moreau, 1963; Moreau, 1985]
- Applied in numerous popular rigid body dynamics methods [Bender et al., 2014]
- **Different solver types**
  - Iterative, e.g., Gauss-Seidel, or Jacobi
    - Favorable efficiency
  - Direct, e.g., pivoting solvers
    - Ideal for high stiffness/mass ratios
  - Hybrid (direct/iterative)
    - Combined benefits

Hybrid direct/iterative solver in Unity Physics

Nonsmooth Multidomain Dynamics

- **Limitations**
  1. Only hard constraints, e.g., rigid bodies with hard contacts
  2. Position "drift" at velocity-level

$$\mathbf{C}(\mathbf{x}) \stackrel{!}{=} \mathbf{0} \longrightarrow \nabla\mathbf{C}^T\mathbf{v}^+ \stackrel{!}{=} \mathbf{0}$$

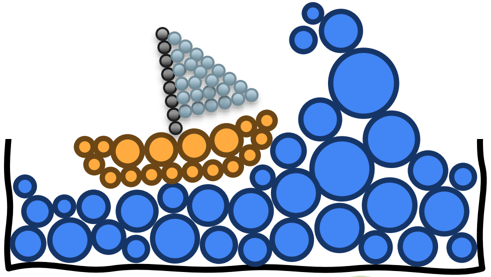
Andrews et al., 2022

Nonsmooth Multidomain Dynamics

- Limitations
  1. Only hard constraints, e.g., rigid bodies with hard contacts
  2. Position "drift" at velocity-level

$$\mathbf{C}(\mathbf{x}) \overset{!}{=} \mathbf{0} \longrightarrow \nabla\mathbf{C}^T\mathbf{v}^+ \overset{!}{=} \mathbf{0}$$

- Generalization to **Nonsmooth Multidomain Dynamics** (NMD)
  1. **Multiple material domains:** model more than rigid bodies



Nonsmooth Multidomain Dynamics

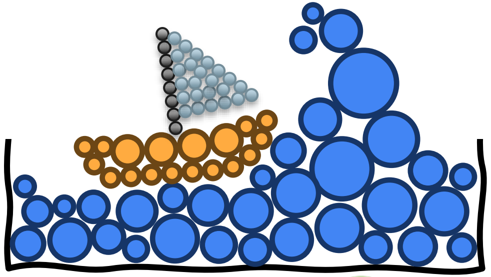
- Limitations
  1. Only hard constraints, e.g., rigid bodies with hard contacts
  2. Position "drift" at velocity-level

$$\mathbf{C}(\mathbf{x}) \stackrel{!}{=} \mathbf{0} \longrightarrow \nabla\mathbf{C}^T\mathbf{v}^+ \stackrel{!}{=} \mathbf{0}$$

- Generalization to **Nonsmooth Multidomain Dynamics** (NMD)
  1. **Multiple material domains:** model more than rigid bodies
     - "Soften" the constraints [Servin et al., 2006]
     - Robust, physically meaningful parametrization
       - Constraints as stiff limits of energy potentials $U(\mathbf{x})$
       - Compliance matrix α

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{C}(\mathbf{x})^T \boxed{\alpha^{-1}} \mathbf{C}(\mathbf{x})$$

$$\mathbf{C}(\mathbf{x}) + \boxed{\lambda\alpha} \stackrel{!}{=} \mathbf{0}$$



Nonsmooth Multidomain Dynamics

- Limitations
  1. Only hard constraints, e.g., rigid bodies with hard contacts
  2. Position "drift" at velocity-level

$$\mathbf{C}(\mathbf{x}) \overset{!}{=} \mathbf{0} \longrightarrow \nabla\mathbf{C}^T\mathbf{v}^+ \overset{!}{=} \mathbf{0}$$

- Generalization to **Nonsmooth Multidomain Dynamics** (NMD)
  1. Multiple material domains: model more than rigid bodies
     - "Soften" the constraints [Servin et al., 2006]
     - Robust, physically meaningful parametrization
       - Constraints as stiff limits of energy potentials $U(\mathbf{x})$
       - Compliance matrix α
  2. **Constraint stabilization**
     - Deals with position "drift" problem

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{C}(\mathbf{x})^T \boxed{\alpha^{-1}} \mathbf{C}(\mathbf{x})$$

$$\mathbf{C}(\mathbf{x}) + \boxed{\lambda\alpha} \overset{!}{=} \mathbf{0}$$

$$\mathbf{C}(\mathbf{x}^+) \approx \boxed{\mathbf{C}(\mathbf{x})} + \boxed{\Delta t}\nabla\mathbf{C}^T\mathbf{v}^+$$

$$\downarrow$$

$$\nabla\mathbf{C}^T\mathbf{v}^+ + \boxed{\alpha\lambda} \overset{!}{=} \boxed{-\mathbf{C}(\mathbf{x})/\Delta t}$$

Nonsmooth Multidomain Dynamics

- Update formulation:

$$\begin{cases} \begin{bmatrix} \mathbf{M} & -\nabla\mathbf{C}_b & -\nabla\mathbf{C}_u \\ \nabla\mathbf{C}_b^T & \mathbf{0} & \mathbf{0} \\ \nabla\mathbf{C}_u^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ \\ \boldsymbol{\lambda}_\mathrm{b} \\ \boldsymbol{\lambda}_\mathrm{u} \end{bmatrix} - \begin{bmatrix} \mathbf{Mv} + \Delta t \mathbf{f}_\mathrm{ext} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w} \end{bmatrix} \\ \mathbf{0} \leq \mathbf{w} \perp \boldsymbol{\lambda}_u \geq \mathbf{0} \end{cases}$$

$$\nabla\mathbf{C}_b^T \mathbf{v}^+ = \mathbf{0}$$

$$\nabla\mathbf{C}_u^T \mathbf{v}^+ \geq \mathbf{0}$$

Nonsmooth Multidomain Dynamics

- Update formulation:

Constraint Relaxation

$$\begin{cases} \begin{bmatrix} \mathbf{M} & -\nabla\mathbf{C}_b & -\nabla\mathbf{C}_u \\ \nabla\mathbf{C}_b^T & \alpha_b & \mathbf{0} \\ \nabla\mathbf{C}_u^T & \mathbf{0} & \alpha_u \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ \\ \boldsymbol{\lambda}_\mathrm{b} \\ \boldsymbol{\lambda}_\mathrm{u} \end{bmatrix} - \begin{bmatrix} \mathbf{Mv} + \Delta t\mathbf{f}_{\mathrm{ext}} \\ -\mathbf{C}_b(\mathbf{x})/\Delta t \\ -\mathbf{C}_u(\mathbf{x})/\Delta t \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w} \end{bmatrix} \\ \mathbf{0} \leq \mathbf{w} \perp \boldsymbol{\lambda}_u \geq \mathbf{0} \end{cases}$$

Constraint Stabilization

$$\begin{aligned} \nabla\mathbf{C}_b^T\mathbf{v}^+ &= \mathbf{0} \\ \nabla\mathbf{C}_u^T\mathbf{v}^+ &\geq \mathbf{0} \end{aligned} \longrightarrow \begin{aligned} \nabla\mathbf{C}_b^T\mathbf{v}^+ + \alpha_b\lambda_b &= -\mathbf{C}_b(\mathbf{x})/\Delta t \\ \nabla\mathbf{C}_u^T\mathbf{v}^+ + \alpha_u\lambda_u &\geq -\mathbf{C}_u(\mathbf{x})/\Delta t \end{aligned}$$

Nonsmooth Multidomain Dynamics

- Update formulation:

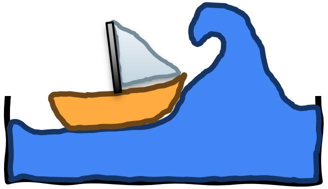
$$
\begin{cases}
\begin{bmatrix} \mathbf{M} & -\nabla \mathbf{C}_b & -\nabla \mathbf{C}_u \\ \nabla \mathbf{C}_b^T & \alpha_b & \mathbf{0} \\ \nabla \mathbf{C}_u^T & \mathbf{0} & \alpha_u \end{bmatrix} \begin{bmatrix} \mathbf{v}^+ \\ \boldsymbol{\lambda}_\mathrm{b} \\ \boldsymbol{\lambda}_\mathrm{u} \end{bmatrix} - \begin{bmatrix} \mathbf{M}\mathbf{v} + \Delta t \mathbf{f}_\mathrm{ext} \\ -\mathbf{C}_b(\mathbf{x})/\Delta t \\ -\mathbf{C}_u(\mathbf{x})/\Delta t \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w} \end{bmatrix} \\
\mathbf{0} \le \mathbf{w} \perp \boldsymbol{\lambda}_u \ge \mathbf{0}
\end{cases}
$$

Guaranteed positive definite

$$
\begin{aligned}
\nabla \mathbf{C}_b^T \mathbf{v}^+ &= \mathbf{0} \\
\nabla \mathbf{C}_u^T \mathbf{v}^+ &\ge \mathbf{0}
\end{aligned}
\longrightarrow
\begin{aligned}
\nabla \mathbf{C}_b^T \mathbf{v}^+ + \alpha_b \lambda_b &= -\mathbf{C}_b(\mathbf{x})/\Delta t \\
\nabla \mathbf{C}_u^T \mathbf{v}^+ + \alpha_u \lambda_u &\ge -\mathbf{C}_u(\mathbf{x})/\Delta t
\end{aligned}
$$

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

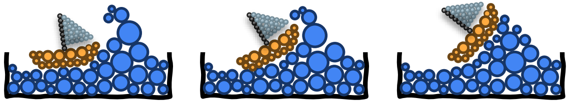


Equations of motion

Simulated material domains

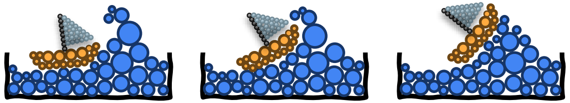
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

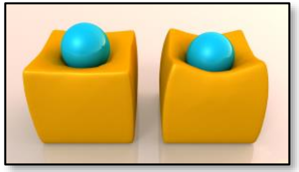


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



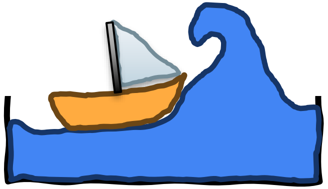
Deformables

Granular Materials

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

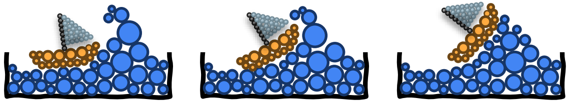
Equations of motion



Simulated material domains

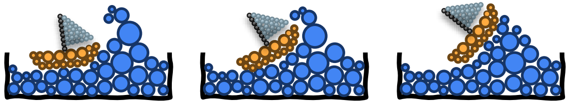
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

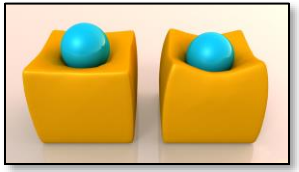


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies



Deformables

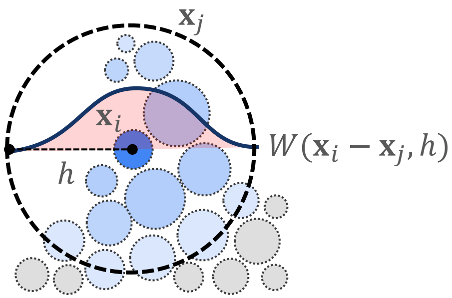
Granular Materials

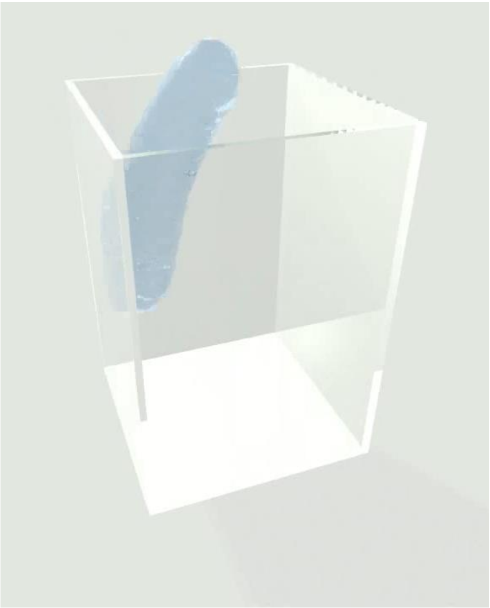
Velocity Based Fluids

- Uniform density constraint, as in PBD [Bodin et al., 2012]
  - Enforces incompressibility
  - SPH-based material interpolation

$$C_i = \frac{\boxed{\rho_i}}{\rho_0} - 1 \stackrel{!}{=} 0$$

$$\boxed{\rho_i = \sum_{j=1}^{n} m_j \, W(\mathbf{x}_i - \mathbf{x}_j, h)}$$





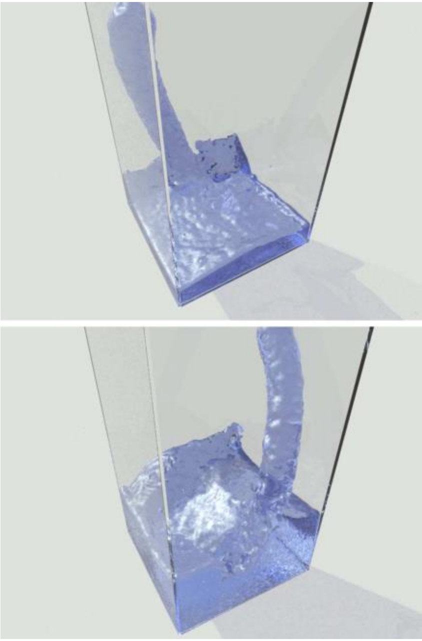
Constraint Fluids [Bodin et al., 2012]

Velocity Based Fluids

- Uniform density constraint, as in PBD [Bodin et al., 2012]
  - Enforces incompressibility
  - SPH-based material interpolation
- **Conversion to velocity-level**

$$C_i = \frac{1}{\rho_0} \underbrace{\sum_{j=1}^{n} m_j \, W(\mathbf{x}_i - \mathbf{x}_j, h)}_{\rho_i} - 1$$

Density constraint on
position-level



Constraint Fluids [Bodin et al., 2012]

Velocity Based Fluids

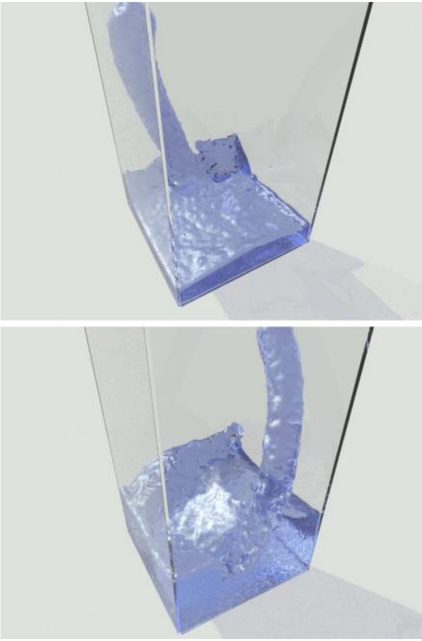
- Uniform density constraint, as in PBD [Bodin et al., 2012]
  - Enforces incompressibility
  - SPH-based material interpolation
- **Conversion to velocity-level**

$$C_i = \underbrace{\frac{1}{\rho_0}\sum_{j=1}^{n} m_j W(\mathbf{x}_i - \mathbf{x}_j, h)}_{\rho_i} - 1 \longrightarrow \boxed{\frac{d}{dt}} C_i = \frac{1}{\rho_0}\sum_{j=1}^{n} m_j \boxed{\nabla W}(\mathbf{x}_i - \mathbf{x}_j, h)\mathbf{n}_{ij}^T \boxed{(\mathbf{v}_i - \mathbf{v}_j)}$$

$$\mathbf{n}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}$$

Density constraint on **position-level**

Density constraint on **velocity-level**



Constraint Fluids [Bodin et al., 2012]

Velocity Based Fluids

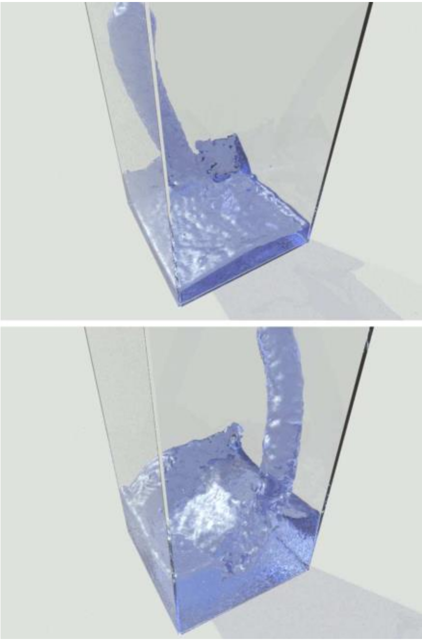
- Uniform density constraint, as in PBD [Bodin et al., 2012]
  - Enforces incompressibility
  - SPH-based material interpolation
- **Conversion to velocity-level**

$$C_i = \frac{1}{\rho_0} \underbrace{\sum_{j=1}^{n} m_j \, W(\mathbf{x}_i - \mathbf{x}_j, h)}_{\rho_i} - 1 \quad \longrightarrow \quad \frac{d}{dt} C_i = \underbrace{\boxed{\frac{1}{\rho_0} \sum_{j=1}^{n} m_j \, \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \mathbf{n}_{ij}^T}}_{\text{Constraint gradient}} \boxed{(\mathbf{v}_i - \mathbf{v}_j)} \overset{!}{=} \mathbf{0}$$

$$\boxed{\nabla C_i^T} \boxed{\mathbf{v}^+} \overset{!}{=} \mathbf{0}$$

Density constraint on **position-level**

Density constraint on **velocity-level**



Constraint Fluids [Bodin et al., 2012]

Velocity Based Rigid Bodies

- Well-suited for multibody systems with contacts
  [Anitescu & Potra, 1997; Lacoursière 2007]
- Velocity-level effects directly supported
  - Joint actuation, kinetic friction, restitution
    See SIGGRAPH course by Andrews et al. [2022]

Andrews et al., 2022

Velocity Based Rigid Bodies

- Well-suited for multibody systems with contacts
  [Anitescu & Potra, 1997; Lacoursière 2007]
- Velocity-level effects directly supported
  - Joint actuation, kinetic friction, restitution
    See SIGGRAPH course by Andrews et al. [2022]
    - **PBD**: include additional velocity-level solve
      [Barreiro et al., 2017; Müller et al., 2020]
    - **XPBI**: reformulate PBD to velocity-level

Andrews et al., 2022

Velocity Based Rigid Bodies

- Well-suited for multibody systems with contacts
  [Anitescu & Potra, 1997; Lacoursière 2007]
- Velocity-level effects directly supported
  - Joint actuation, kinetic friction, restitution
    See SIGGRAPH course by Andrews et al. [2022]
    - **PBD**: include additional velocity-level solve
      [Barreiro et al., 2017; Müller et al., 2020]
    - **XPBI**: reformulate PBD to velocity-level
    - **NMD**: add target velocity $\mathbf{v}_0$ to constraint

Andrews et al., 2022

$$\nabla \mathbf{C}_b^T \mathbf{v}^+ + \alpha_b \lambda_b = -\mathbf{C}_b(\mathbf{x})/\Delta t \longrightarrow \nabla \mathbf{C}_b^T \mathbf{v}^+ + \alpha_b \lambda_b = \boxed{\mathbf{v}_0} - \mathbf{C}_b(\mathbf{x})/\Delta t$$

Target velocity

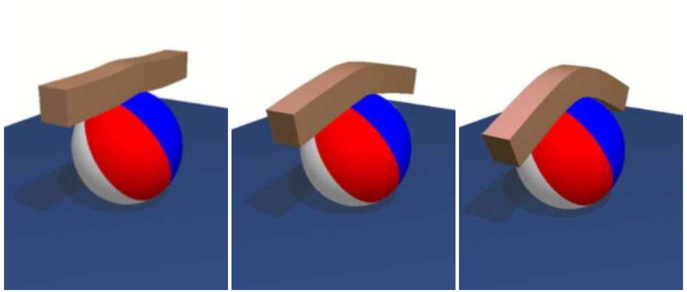
Velocity Based Rigid Bodies

- Well-suited for multibody systems with contacts
  [Anitescu & Potra, 1997; Lacoursière 2007]
- Velocity-level effects directly supported
- **Constraint-based adhesion** [Gascon et al., 2010]

Gascon et al., 2010

Velocity Based Deformables

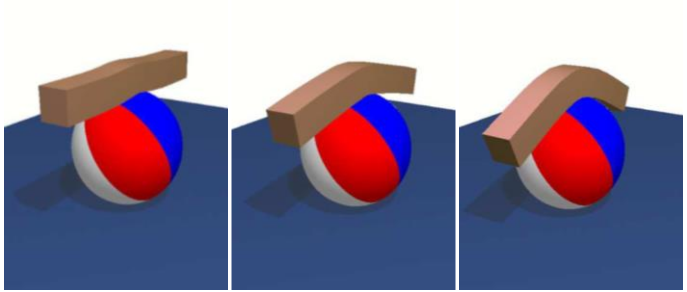
- **Elastic solids** [Servin et al., 2006]
  - FEM-based material discretization into tetrahedra T_i
  - Constraint derived from elastic strain energy U_i
    - Same as in XPBD [Macklin et al., 2016]



Servin et al., 2006

Velocity Based Deformables

- **Elastic solids** [Servin et al., 2006]
  - FEM-based material discretization into tetrahedra T_i
  - Constraint derived from elastic strain energy U_i
    - Same as in XPBD [Macklin et al., 2016]



Servin et al., 2006

Elastic energy (general form):

$$U(\mathbf{x}) = \frac{1}{2}\mathbf{C}(\mathbf{x})^T \alpha^{-1} \boxed{\mathbf{C}(\mathbf{x})}$$

constraints $\mathbf{C}(\mathbf{x})$

$\longrightarrow$

Elastic strain energy for T_i :

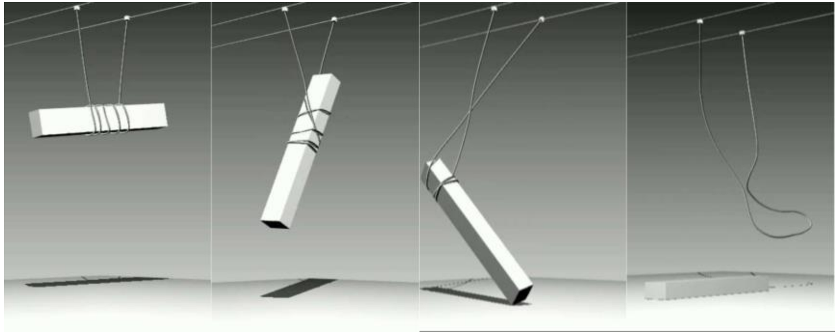
$$U_i(\mathbf{x}) = \frac{1}{2}\left(V^{1/2}\varepsilon\right)^T \alpha^{-1} \boxed{\left(V^{1/2}\varepsilon\right)}$$

V: tetrahedron volume
ε : Green strain tensor

constraints $\mathbf{C}(\mathbf{x}) = V^{1/2}\varepsilon$

Velocity Based Deformables

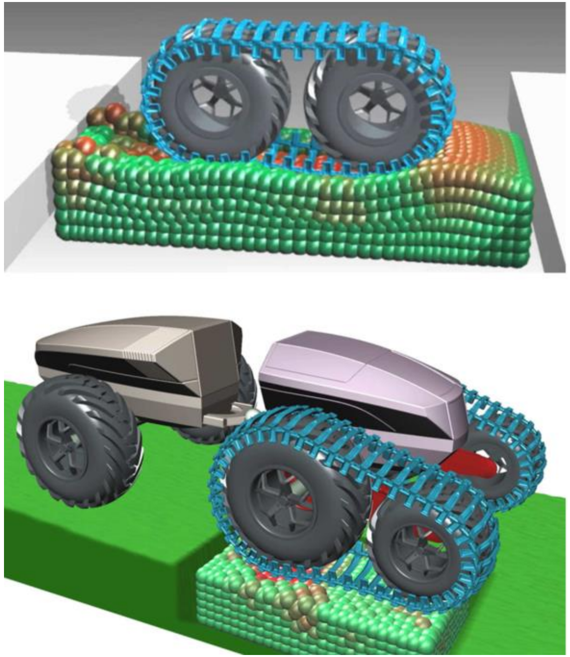
- Elastic solids [Servin et al., 2006]
- **Rods** [Servin and Lacoursière, 2008]
  - Kirchhoff rod model
  - Geometric stiffness for improved accuracy and stability [Tournier et al., 2015; Andrews et al., 2017]



Servin et al., 2010

Velocity Based Deformables

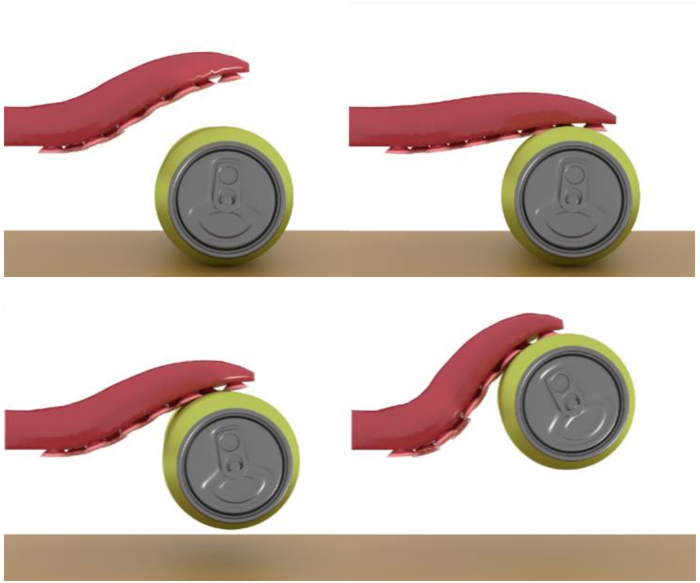
- Elastic solids [Servin et al., 2006]
- Rods [Servin and Lacoursière, 2008]
- **Granular Materials**
  - Nonsmooth DEM model [Servin et al., 2014]
  - Elastoplastic materials via plasticity theory [Nordberg and Servin, 2018]
    - Also: XPBI, PBD on velocity-level [Yu et al., 2024]



Nordberg and Servin, 2018

Velocity Based Deformables

- Elastic solids [Servin et al., 2006]
- Rods [Servin and Lacoursière, 2008]
- Granular Materials
- **Suction** [Bernardin et al., 2022]
    - Passive suction between elastic solids
    - Constraint-based friction, contact and pressure

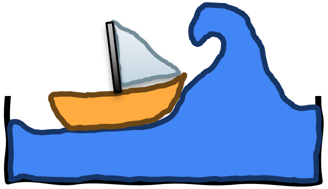


Bernardin et al., 2022

Unified modeling framework

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$

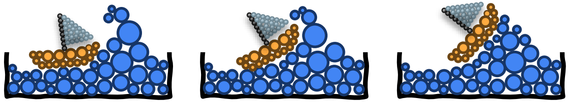
Equations of motion



Simulated material domains

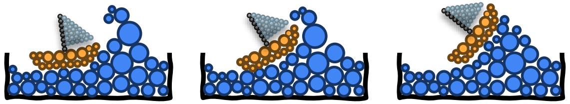
1) **Discretization** of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

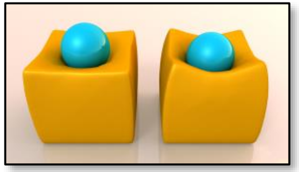


Velocity-level formulation

2) **Multiphysics** modeling via **constraints**

Fluids

Rigid Bodies

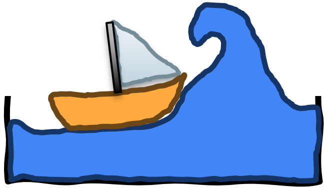


Deformables

Granular Materials

Unified modeling framework

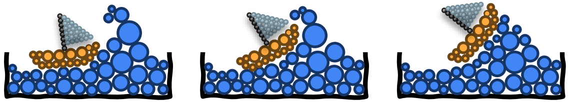
$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$
$$\mathbf{C}_b(\mathbf{x}) = \mathbf{0}$$
$$\mathbf{C}_u(\mathbf{x}) \geq \mathbf{0}$$



Equations of motion | Simulated material domains

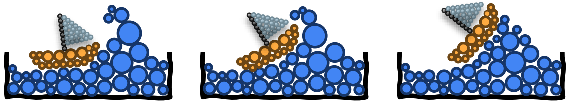
1) Discretization of equations of motion

$$\mathbf{x}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$



Position-level formulation

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$

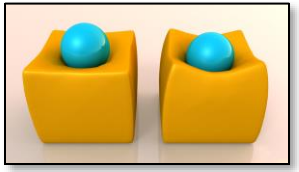


Velocity-level formulation

2) Multiphysics modeling via **constraints**

Fluids

Rigid Bodies



Deformables

Granular Materials

Relation to Position Based Dynamics

- Strong similarities of NMD and XPBD
  - Same constraint relaxation [Servin et al., 2006]

Relation to Position Based Dynamics

- Strong similarities of NMD and XPBD
  - Same constraint relaxation [Servin et al., 2006]
  - Yields similar formulation
    - But: XPBD employs Quasi-Newton method (nonlinear)

$$\begin{bmatrix} \mathbf{M} & -\nabla \mathbf{C}_b \\ \nabla \mathbf{C}_b^T & \boxed{\alpha_b} \end{bmatrix} \begin{bmatrix} \boxed{\mathbf{v}^+} \\ \lambda_b \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{v} + \Delta t \mathbf{f}_{ext} \\ -\mathbf{C}_b(\mathbf{x})/\Delta t \end{bmatrix}$$

NMD system
(only bilateral constraints)

$$\begin{bmatrix} \mathbf{M} & -\nabla \mathbf{C}_b \\ \nabla \mathbf{C}_b^T & \boxed{\alpha_b/\Delta t^2} \end{bmatrix} \begin{bmatrix} \boxed{\Delta \mathbf{x}} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ * \end{bmatrix}$$

XPBD system: one Newton iteration
(only bilateral constraints)

Relation to Position Based Dynamics

- Different time discretization
  - (X)PBD: implicit Euler
    - ✗ high numerical dissipation
    - ✓ increased stability, beneficial for interactive applications
  - NMD: semi-implicit (symplectic) Euler
    - ✓ improved energy conservation
    - ✗ lower stability

PBD velocity and position update:

$$\mathbf{p} = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$
$$\mathbf{v}^+ = \frac{\mathbf{p} - \mathbf{x}}{\Delta t}$$
$$\mathbf{x}^+ = \mathbf{p}$$

NMD velocity and position update:

$$\mathbf{v}^+ = solver(\mathbf{x}, \mathbf{v}, \Delta t)$$
$$\mathbf{x}^+ = \boldsymbol{x} + \Delta t \mathbf{v}^+$$

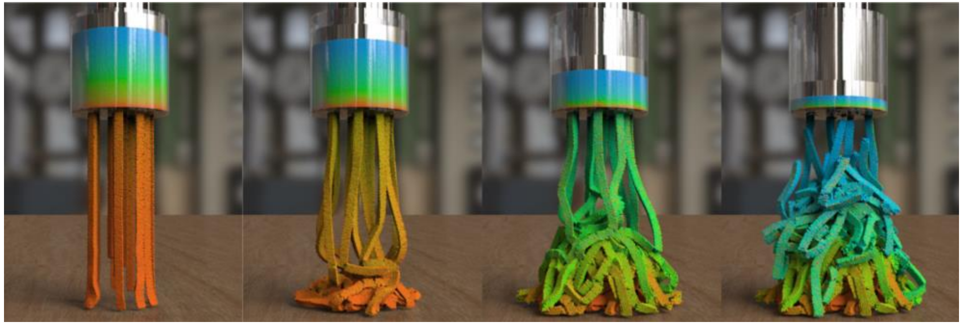
Relation to Position Based Dynamics

- Velocity-level effect support (joint actuation, kinetic friction, restitution)
  - **NMD**: Supported in formulation
  - **PBD**: Requires special treatment [Barreiro et al., 2017; Müller et al., 2020]

Relation to Position Based Dynamics

- Velocity-level effect support (joint actuation, kinetic friction, restitution)
  - **NMD**: Supported in formulation
  - **PBD**: Requires special treatment [Barreiro et al., 2017; Müller et al., 2020]
  - **XPBI**: Uses **velocity-level reformulation** of XPBD [Yu et al., 2024]
    - For accurate deformation gradient $\mathbf{F}^+$
    - Resembles NMD with Schur-Complement and Projected Gauss-Seidel solver

$$\mathbf{F}^+ = \mathbf{F} + \Delta t \boxed{\nabla \mathbf{v}^+} \mathbf{F}$$

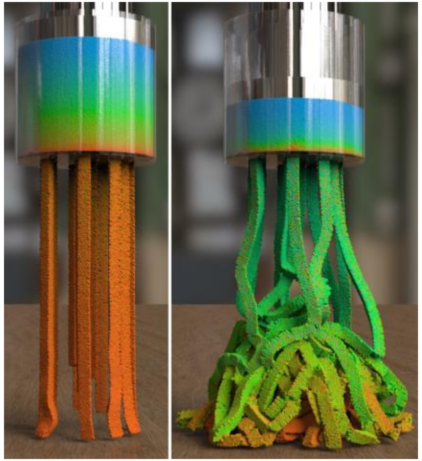


Modeling clay-like material pressed through a cylindrical mold in XPBI [Yu et al., 2024]

Conclusion

Part I: Constraint-Based Multiphysics Modeling

- Flexible, Lagrangian domain discretization
- Materials and phenomena modeled using constraints
  - Unified modeling framework
  - Strongly coupled materials and phenomena
- Position-level and velocity-level formulations
  - Can be combined based on requirements



Yu et al., 2024

Upcoming talks:

Part II: Energy-Based Multiphysics Modeling

Part III: Lagrangian Point-Based, Eulerian and Hybrid Methods

End of Part I of III

Thank you!

Do you have any questions?

Simulated (left) vs. real excavator digging in PBD soil [Holz and Galarneau, 2018]

Viscoplastic paint on cloth in XPBI [Yu et al., 2024]