

Embodied Vision

Physical Modeling: Lagrangian Mechanics, Incremental
Potential Contact Modeling, and Material Point Methods

Tsung-Wei Ke

Spring 2026

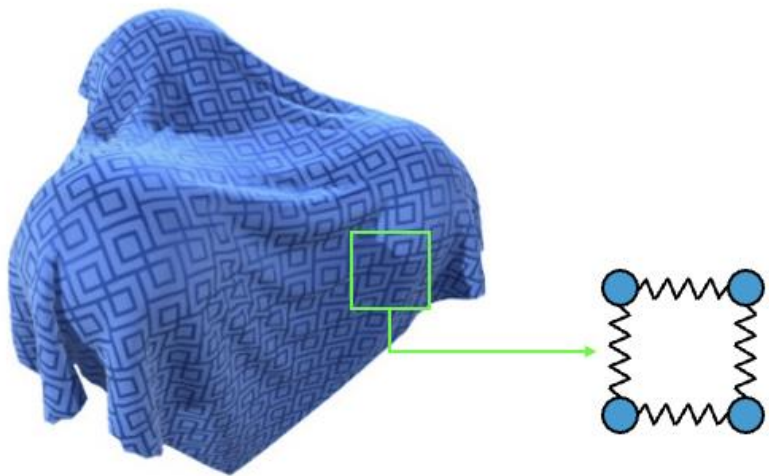


Disclaimer

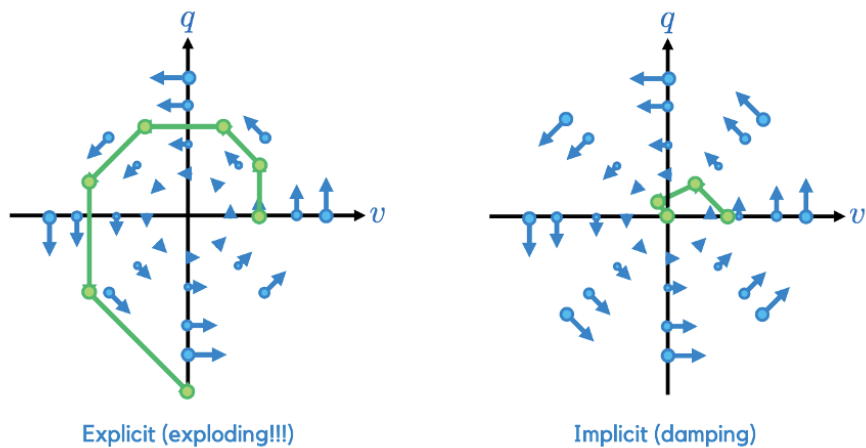
- This lecture borrows contents heavily from
 - [Physics-based Animation](#) by David I.W Levin at University of Toronto
 - [Profounds Physic](#) by Ville Hirvonen
 - [Physics-based Animation of Solids and Fluids](#) by Minchen Li at CMU
- This is an introductory lecture to physical modeling. Check the advanced courses in physic-based simulation for more details.

Recap

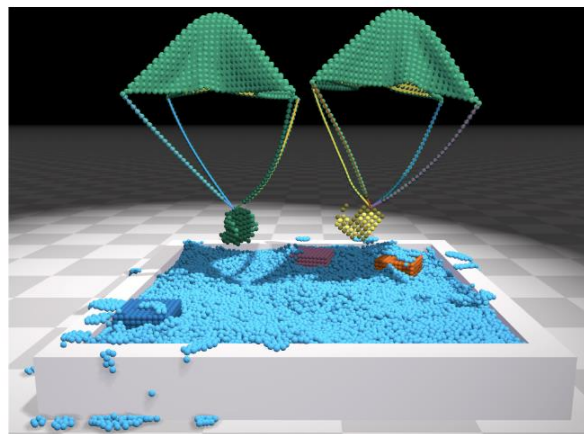
Mass-Spring System



Time Integration



Position-based Dynamic

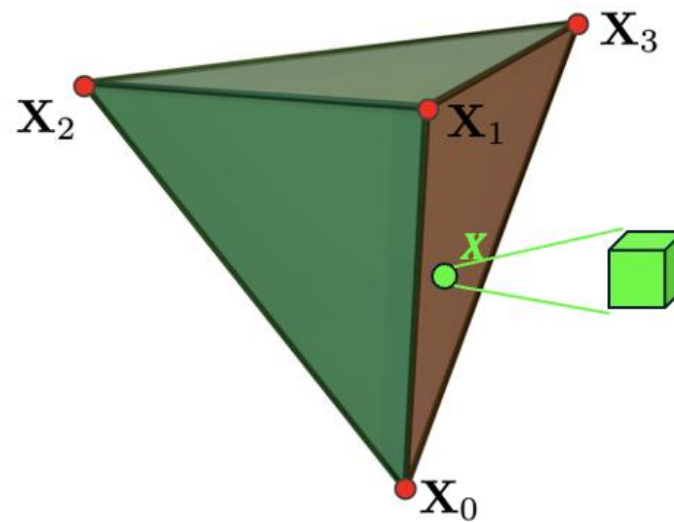


Algorithm 1 Position-based dynamics

```

1: for all vertices  $i$  do
2:   initialize  $\mathbf{x}_i = \mathbf{x}_i^0$ ,  $\mathbf{v}_i = \mathbf{v}_i^0$ ,  $w_i = 1/m_i$ 
3: end for
4: loop   Derive the initial kinetic update with Newton's law
5:   for all vertices  $i$  do  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{\text{ext}}(\mathbf{x}_i)$ 
6:   for all vertices  $i$  do  $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$ 
7:   for all vertices  $i$  do genCollConstraints( $\mathbf{x}_i \rightarrow \mathbf{p}_i$ )
8:   loop solverIteration times
9:     projectConstraints( $C_1, \dots, C_{M+M_{\text{Coll}}}$ ;  $\mathbf{p}_1, \dots, \mathbf{p}_N$ )
10:  end loop   Modify the kinetic update based on constraints
11:  for all vertices  $i$  do
12:     $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$ 
13:     $\mathbf{x}_i \leftarrow \mathbf{p}_i$ 
14:  end for
15:  velocityUpdate( $\mathbf{v}_1, \dots, \mathbf{v}_N$ )
16: end loop
  
```

Finite Elements



Content

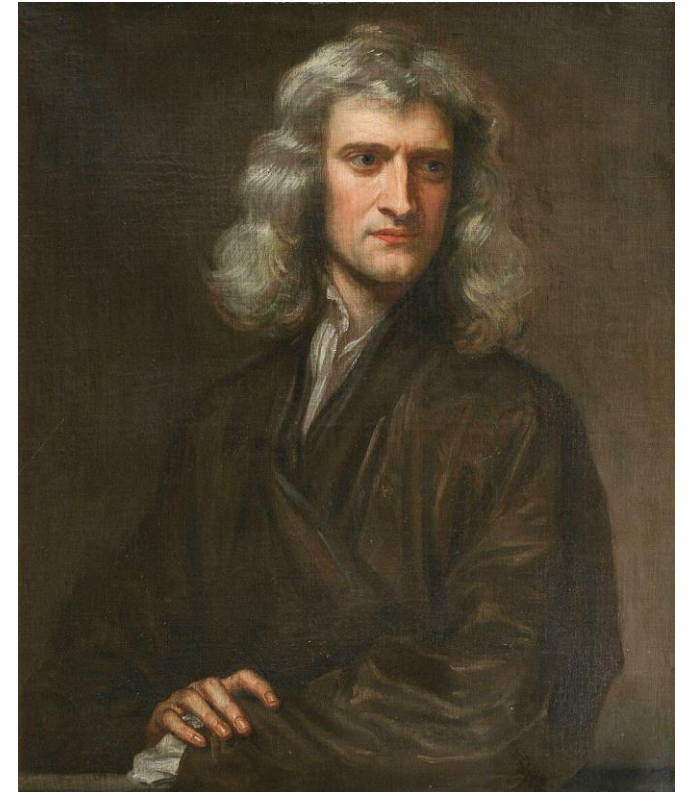
- Lagrangian Mechanics
- Incremental Potential Contact Modeling
- Material Point Methods

Content

- Lagrangian Mechanics
- Incremental Potential Contact Modeling
- Material Point Methods

Classical Mechanics (Vectorial Mechanics)

- The 1st Law of Motion: Every object will remain at rest or in uniform motion in a straight line unless compelled to change its state by the action of an external force
- The 2nd Law of Motion: $\vec{f} = m\vec{a}$, where force is a property with a magnitude and a direction
- The 3rd Law of Motion: For every action there is an equal and opposite reaction



Isaac Newton

Classical Mechanics (Vectorial Mechanics)

- In Newtonian mechanics, we think of motion as being a result of different forces
- However, what we care is to describe motion and dynamics, and **force** is just one way to describe motion
- Describing motion with force may be extremely complex, e.g. collision of deformable objects



What is the Alternative?

Kinetic and Potential Energy

- Potential energy from gravity:

$$V = mgh$$

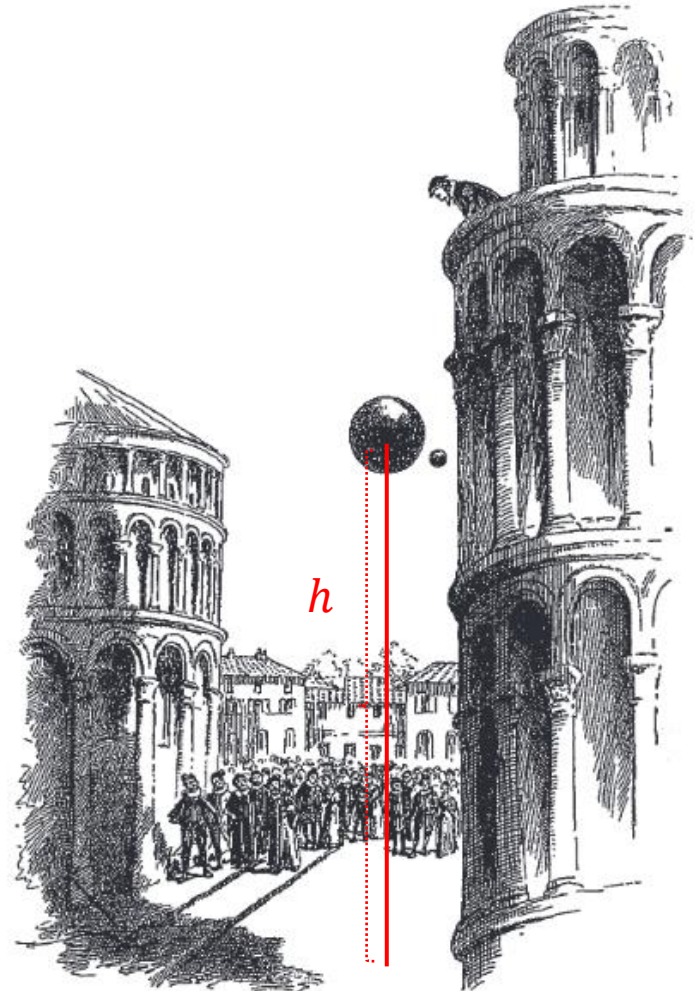
- Kinetic energy of the ball:

$$T = \frac{1}{2}mv^2$$

- Conservation of energy:

$$V_1 + T_1 = V_2 + T_2$$

- This is high-school physics



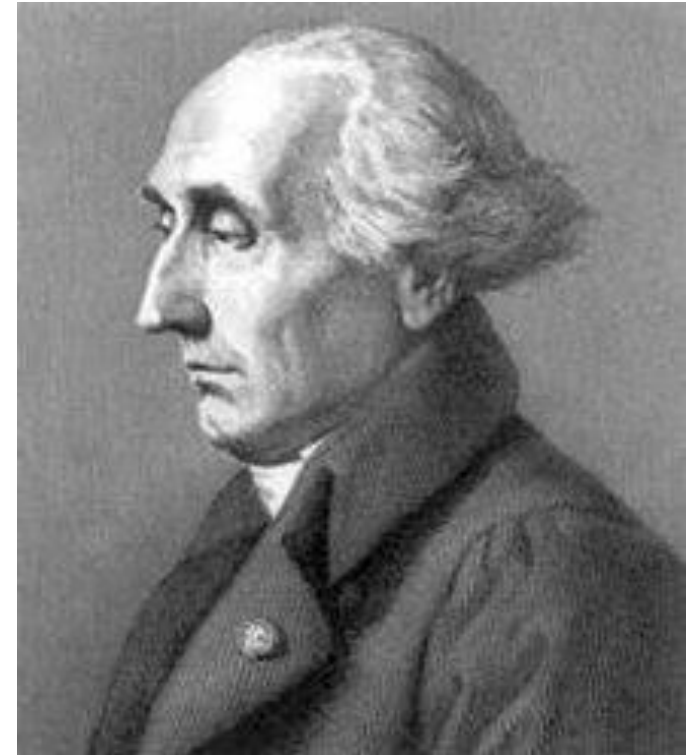
'They were seen to fall evenly.'

Lagrangian Mechanics

- Idea: Describe motion via energies
- Energy is a property without a direction
- Changes in potential energy induce changes of motion
- The Lagrangian: a state of motion at any particular point in time, described by the kinetic and potential energies

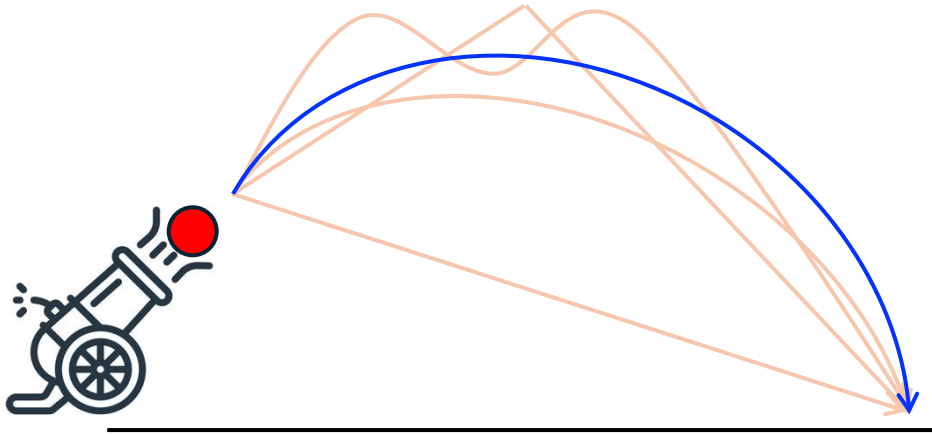
$$L = T - V$$

Kinetic energy Potential energy



Joseph-Louis Lagrange

The Principle of Least Action

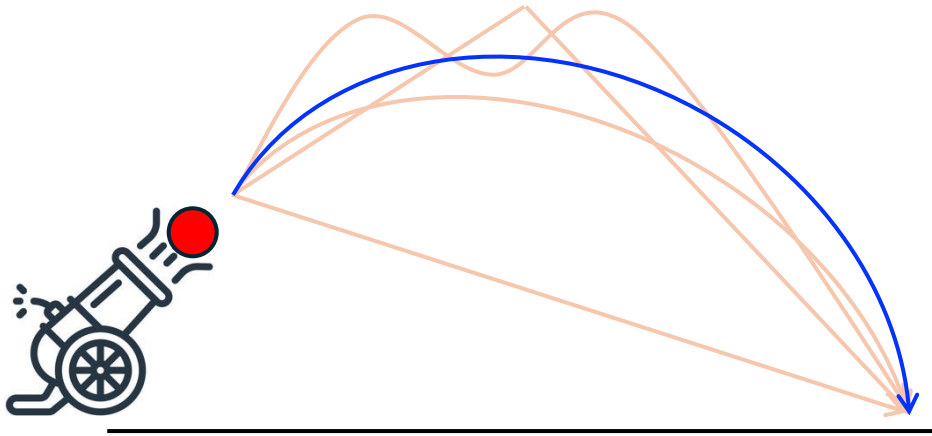


- Each trajectory through space and time has an “action”
- **Action:** an integral over time of the Lagrangian at each point of the trajectory

$$A = \int_{t_1}^{t_2} L(x, \dot{x}) dt$$

- Why does the canon ball move in the blue trajectory?
 - Classical mechanics: $F = mg \dots$
 - Lagrangian mechanics: it moves in such a way that the “action” is minimized...

The Principle of Least Action



- Each trajectory through space and time has an “action”
- **Action:** an integral over time of the Lagrangian at each point of the trajectory

$$A = \int_{t_1}^{t_2} L(x, \dot{x}) dt$$

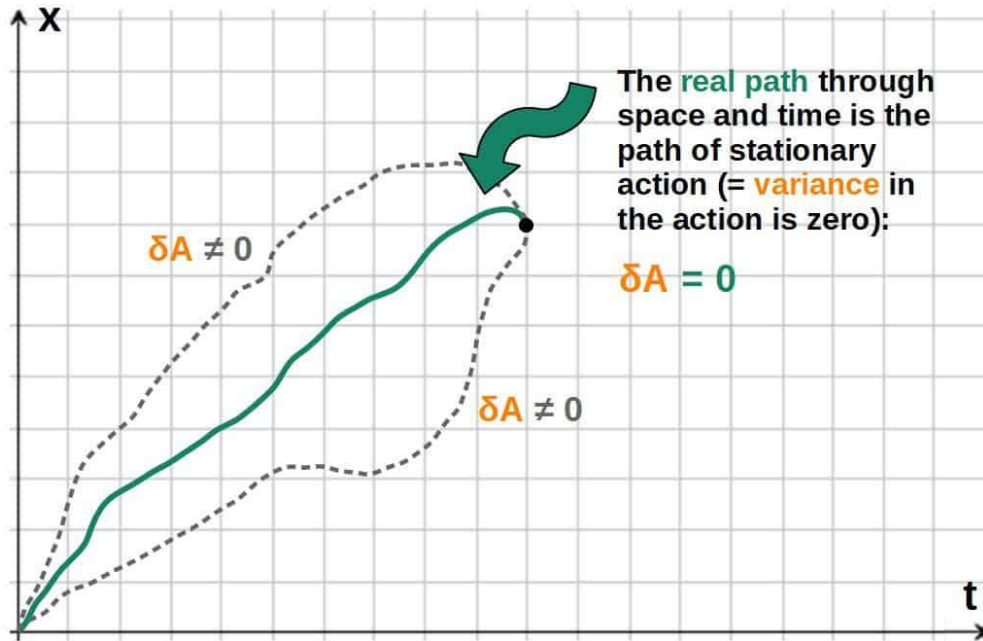
- Why does the canon ball move in the blue trajectory?

➤ Classical mechanics: $F = mg \dots$

➤ Lagrangian mechanics: it moves in such a way that the “action” is minimized...

The principle of stationary action that uniquely define the trajectory a system will take

The Principle of Stationary Action



<https://profoundphysics.com/lagrangian-mechanics-for-beginners/>

- **Action:** an integral over time of the Lagrangian at each point of the trajectory

$$A = \int_{t_1}^{t_2} L(q, \dot{q}) dt$$

- The principle of stationary action:

$$\delta A = \delta \int_{t_1}^{t_2} L(q, \dot{q}) dt = 0$$

- In fact, A can be considered as a functional (a function of functions), characterized by the path through time and space
- Minimize A to find the optimal path

Lagrangian Mechanics

- The Lagrangian:

Potential energy

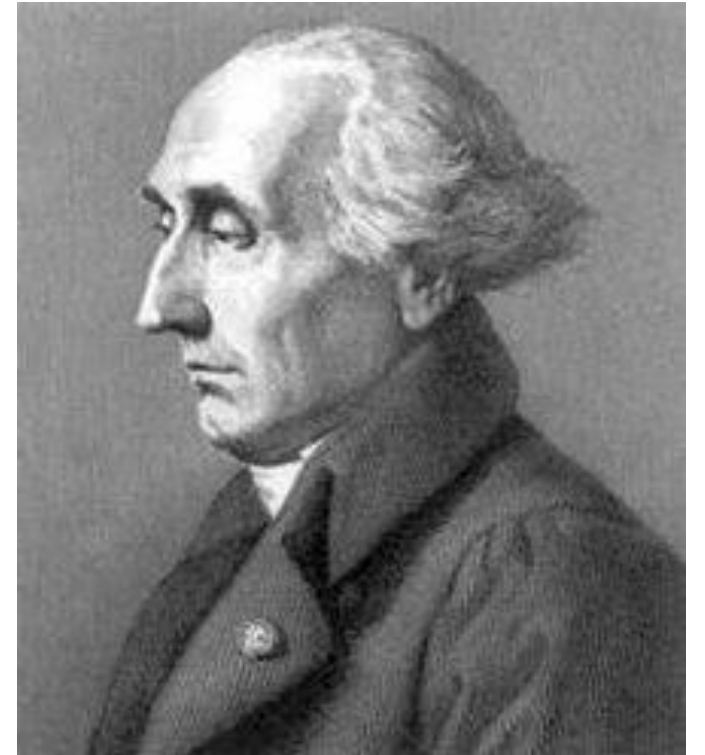
$$L = T - V$$

Kinetic energy

- Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}}$$

There was a typo...



Joseph-Louis Lagrange

Derivation of Euler-Lagrange Equation

- The Principle of Least Action:

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}) dt = 0 \Rightarrow \int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = 0$$

- Where:

$$\int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \frac{d}{dt} \delta q \right) dt$$

Derivation of Euler-Lagrange Equation

- The Principle of Least Action:

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}) dt = 0 \Rightarrow \int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = 0$$

- Where:

$$\int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \frac{d}{dt} \delta q \right) dt$$

$$\frac{d}{dt} (f(t)g(t)) = g \frac{df}{dt} + f \frac{dg}{dt} \Rightarrow f \frac{dg}{dt} = \frac{d}{dt} (fg) - g \frac{df}{dt}$$

Derivation of Euler-Lagrange Equation

- The Principle of Least Action:

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}) dt = 0 \Rightarrow \int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = 0$$

- Where:

$$\int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \frac{d}{dt} \delta q \right) dt$$

$$\frac{d}{dt} (f(t)g(t)) = g \frac{df}{dt} + f \frac{dg}{dt} \Rightarrow f \frac{dg}{dt} = \frac{d}{dt} (fg) - g \frac{df}{dt}$$

$$\frac{\partial L}{\partial \dot{x}} \frac{d}{dt} \delta x = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \delta x \right) - \delta x \frac{d}{dt} \frac{\partial L}{\partial \dot{x}}$$

- We have:

$$\begin{aligned}
\int_{t_1}^{t_2} \delta L(q, \dot{q}) dt &= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \frac{d}{dt} \delta q \right) dt \\
&= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \delta q \right) - \delta q \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt \\
&= \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \delta q \right) dt + \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q - \delta q \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt \\
&= \left| \frac{\partial L}{\partial \dot{q}} \delta q \right|_{t_1}^{t_2} + \int_{t_1}^{t_2} \delta q \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt
\end{aligned}$$

- We have:

$$\begin{aligned}
 \int_{t_1}^{t_2} \delta L(q, \dot{q}) dt &= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \delta \dot{q} \right) dt = \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{\partial L}{\partial \dot{q}} \frac{d}{dt} \delta q \right) dt \\
 &= \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q + \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \delta q \right) - \delta q \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt \\
 &= \int_{t_1}^{t_2} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \delta q \right) dt + \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial q} \delta q - \delta q \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt \\
 &= \left| \frac{\partial L}{\partial \dot{q}} \delta q \right|_{t_1}^{t_2} + \int_{t_1}^{t_2} \delta q \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt
 \end{aligned}$$

- The start and end points of the path should be kept the same when we vary the path
- In other words, the variance in the start and end points should equal zero

$$\delta x(t_2) = 0$$

$$\delta x(t_1) = 0$$

Derivation of Euler-Lagrange Equation

- The Principle of Least Action:

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}) dt = 0 \Rightarrow \int_{t_1}^{t_2} \delta L(q, \dot{q}) dt = 0 \Rightarrow \int_{t_1}^{t_2} \delta q \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) dt = 0$$

- For any arbitrary $\delta \mathbf{x}$, the integral is always zero, if the integrand is ways zero:

$$\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = 0 \Rightarrow \frac{\partial L}{\partial q} = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}}$$

Lagrangian Mechanics

- The Lagrangian:

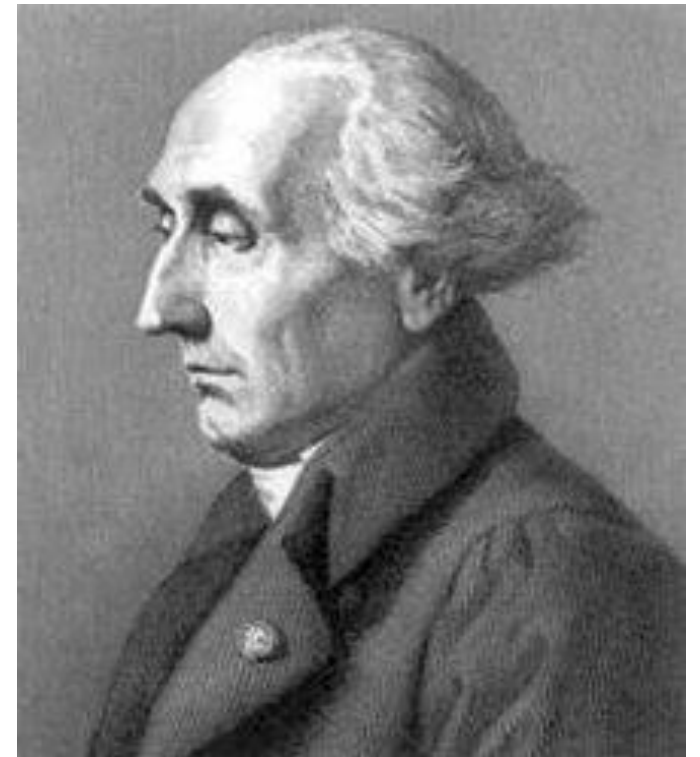
Potential energy

$$L = T - V$$

Kinetic energy

- Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial L}{\partial \mathbf{q}} \Rightarrow \frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}} = - \frac{\partial V}{\partial \mathbf{q}}$$

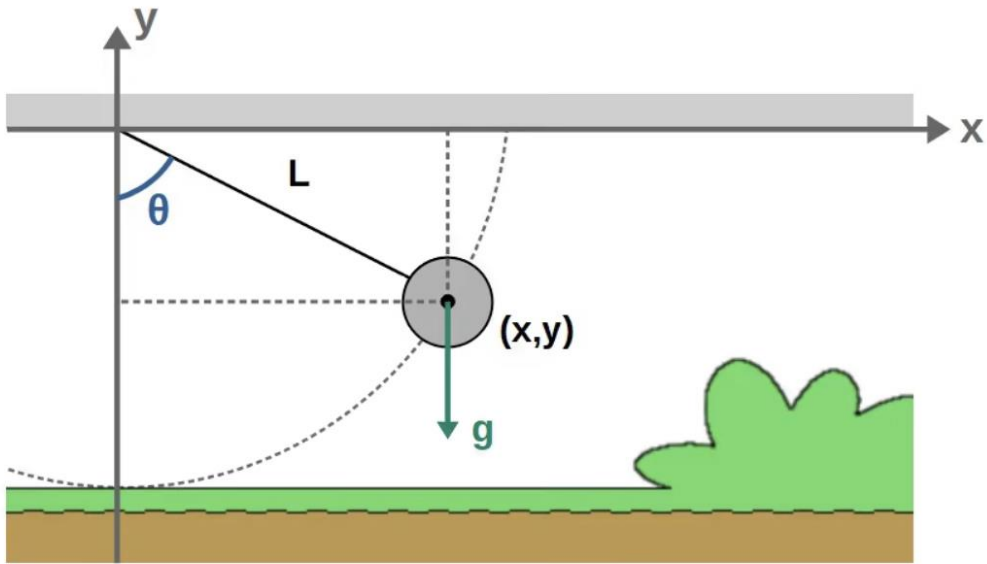


Joseph-Louis Lagrange

Why Lagrangian Mechanics?

- A unified principle for deriving the equations of motions
 - Rigid bodies
 - Deformable objects
 - Fluids
 - ...
- We can re-derive the same equations of motions as the one derived by classical mechanics

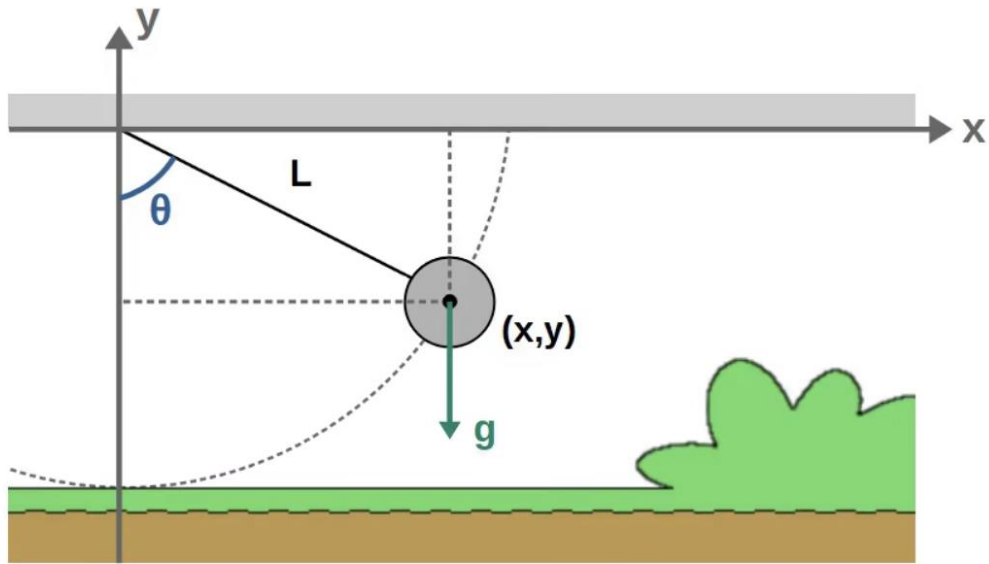
Re-derivation of Pendulum Motion



<https://profoundphysics.com/lagrangian-mechanics-for-beginners/>

- We have: $x = L \sin \theta$ and $y = L \cos \theta$
- Kinetic energy: $T = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2) = \frac{1}{2} m \left((L\dot{\theta} \cos \theta)^2 + (L\dot{\theta} \sin \theta)^2 \right) = \frac{1}{2} mL^2 \dot{\theta}^2$
- Potential energy: $V = -mgL \cos \theta$

Re-derivation of Pendulum Motion



<https://profoundphysics.com/lagrangian-mechanics-for-beginners/>

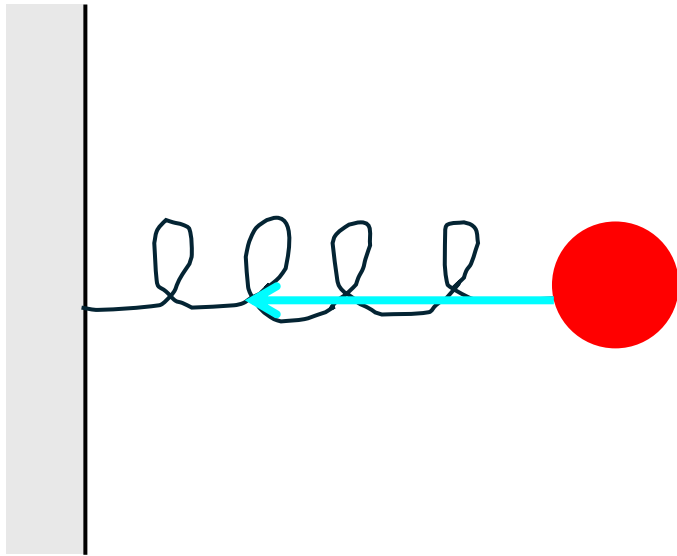
- We have: $x = L \sin \theta$ and $y = L \cos \theta$
- Kinetic energy: $T = \frac{1}{2} mL^2 \dot{\theta}^2$
- Potential energy: $V = -mgL \cos \theta$
- Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\theta}} = - \frac{\partial V}{\partial \theta}$$

$$\Rightarrow \frac{d}{dt} \frac{\partial}{\partial \dot{\theta}} \left(\frac{1}{2} mL^2 \dot{\theta}^2 \right) = - \frac{\partial}{\partial \theta} (-mgL \cos \theta)$$

$$\Rightarrow \frac{d}{dt} (mL^2 \dot{\theta}) = -(mgL \sin \theta) \Rightarrow \ddot{\theta} = -\frac{g}{L} \sin \theta$$

Remember Our Simple Spring-Mass System



- The total force on node i :

$$F_i = \sum_{i,j \in \mathcal{E}} F_{i,j}^{spring} + F_{i,j}^{dashpot} + F_i^{ext}$$

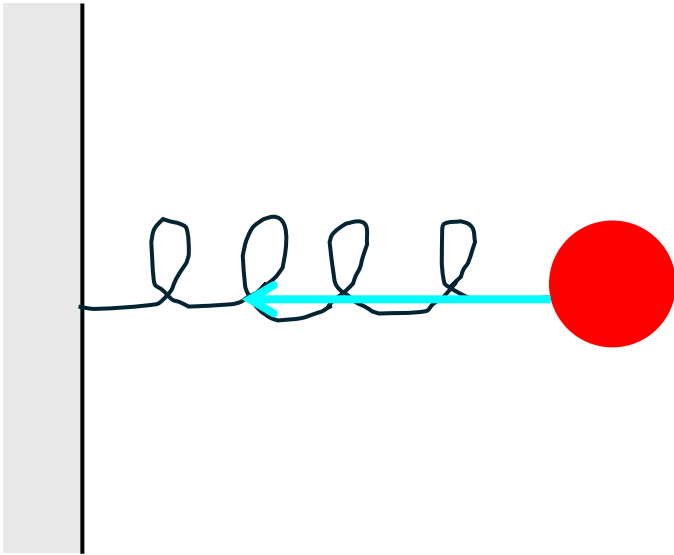
- Another end is attached to the wall: $x_j = 0$
- The rest length of the spring is zero: $l_{i,j} = 0$
- Spring force:

$$F_i^{spring} = -kx_i(t)$$

- Newton's second law:

$$F_i = m\ddot{x}_i(t) \Rightarrow \underbrace{-kx_i(t)}_{\text{2nd order ODE}} = m\ddot{x}_i(t)$$

Re-derivation of Spring Motion



- Kinetic energy: $T = \frac{1}{2} m \dot{x}^2$
- Potential energy: $V = \frac{1}{2} k x^2$
- Euler-Lagrange Equation:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{x}} = - \frac{\partial V}{\partial x}$$

$$\Rightarrow \frac{d}{dt} \frac{\partial}{\partial \dot{x}} \left(\frac{1}{2} m \dot{x}^2 \right) = - \frac{\partial}{\partial x} \left(\frac{1}{2} k x^2 \right)$$

$$\Rightarrow \frac{d}{dt} (m \dot{x}) = -(kx) \Rightarrow m \ddot{x} = -kx$$

Why Lagrangian Mechanics?

- Lagrangian mechanics offers a perspective that dynamics and motion are optimization process.

$$\min \int_{t_1}^{t_2} L(q, \dot{q}) dt$$

- In other words, we can use off-the-shelf optimization tool boxes to solve dynamics

Content

- Lagrangian Mechanics
- Incremental Potential Contact Modeling
- Material Point Methods

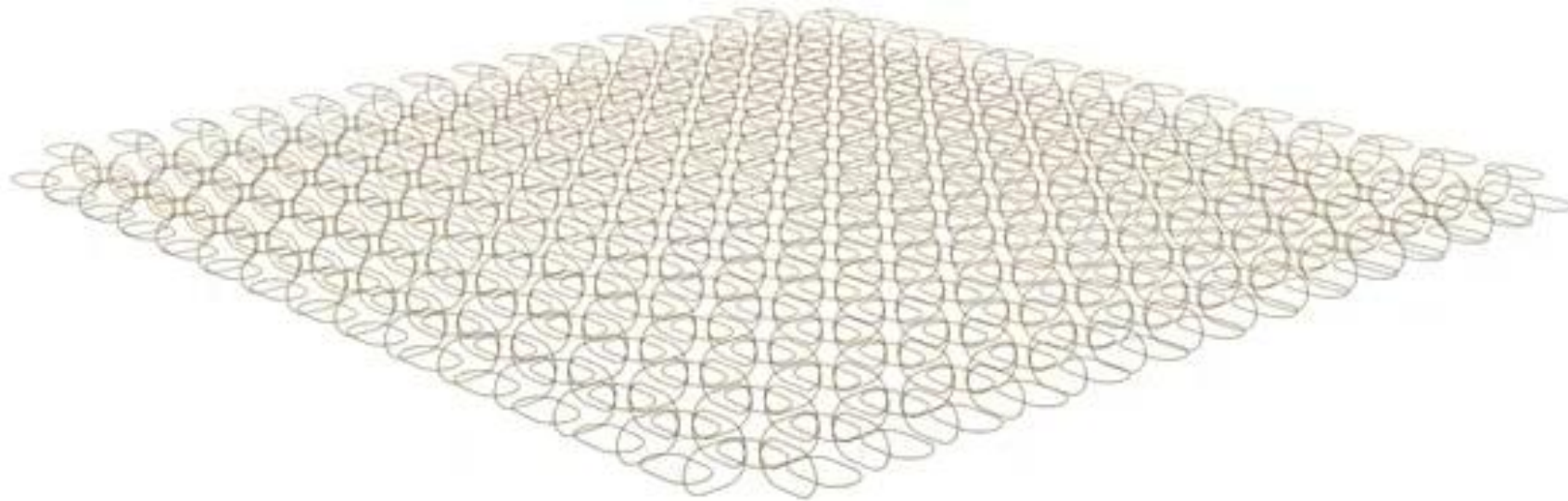
Formulate Complex Dynamics and Motions as Optimization Problems

Table Cloth Trick - 0.5m/s Pull

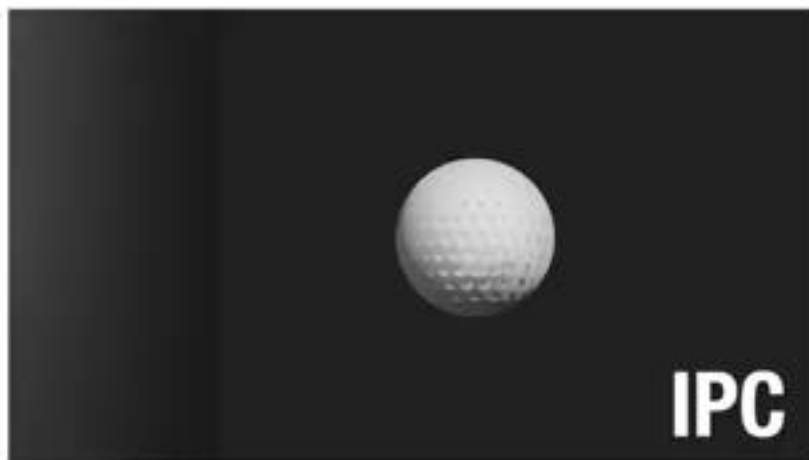


57.1K nodes
Contacts/step (max): 89.1K
h: 0.01s
1x playback speed

Formulate Complex Dynamics and Motions as Optimization Problems



Formulate Complex Dynamics and Motions as Optimization Problems



5e-4X



**Recorded
Footage**

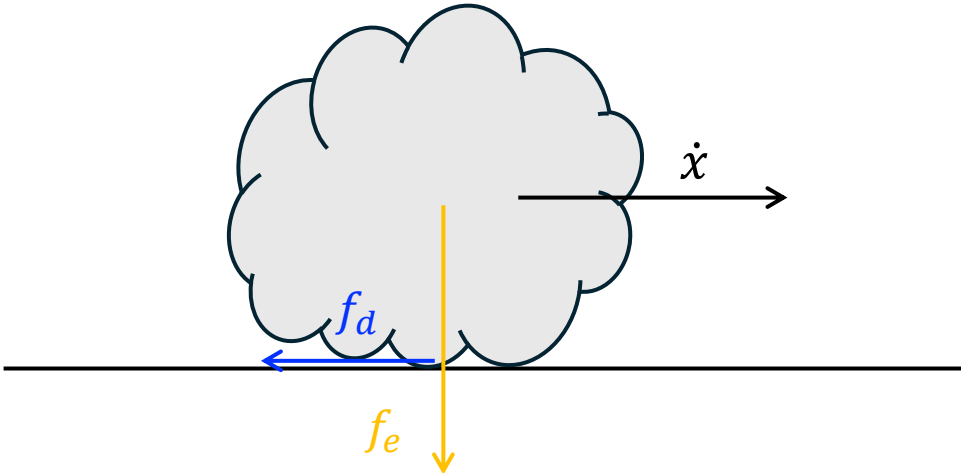
Contact Modeling

- Consider elastic objects with contact and friction. The Lagrangian is:

$$L(x, \dot{x}) = \frac{1}{2} \dot{x}^T M \dot{x} - \Psi(x) + x^T (f_e + f_d)$$

dissipative frictional forces

Potential energy of elastic objects external forces



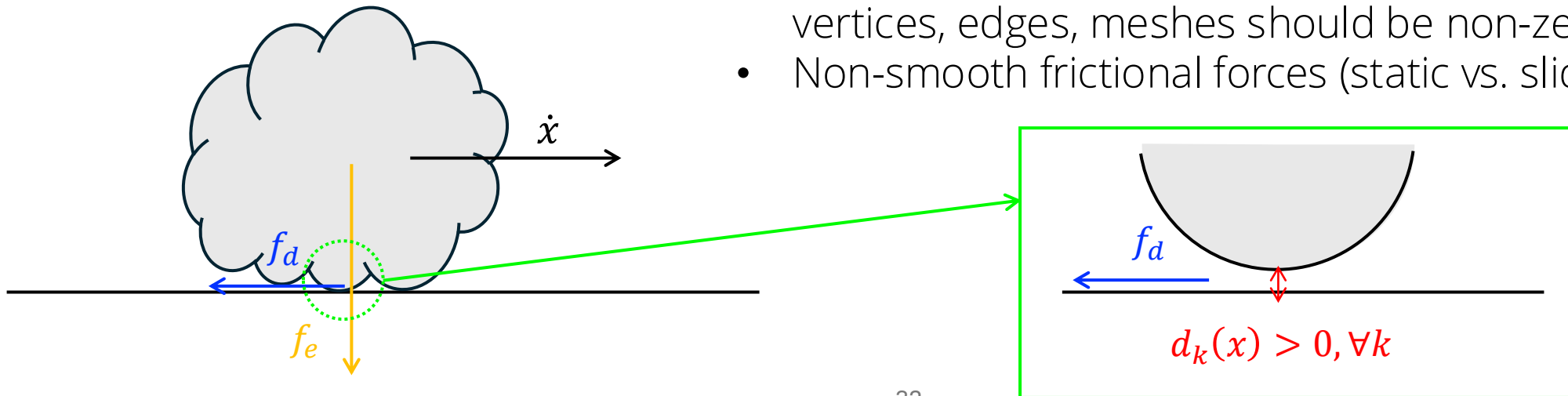
Contact Modeling

- Consider elastic objects with contact and friction. The Lagrangian is:

$$L(x, \dot{x}) = \frac{1}{2} \dot{x}^T M \dot{x} - \Psi(x) + x^T (f_e + f_d)$$

Challenges in contact modeling

- Intersection-free trajectories \mathcal{A} : distances between vertices, edges, meshes should be non-zero
- Non-smooth frictional forces (static vs. sliding surface)



With Lagrangian, We Consider Dynamics as an Optimization Process

- With some math works, we know simulating such dynamics with Implicit Euler time integration can be formulated as an optimization problem of Incremental Potential (IP).

$$L(x, \dot{x}) = \frac{1}{2} \dot{x}^T M \dot{x} - \Psi(x) + x^T (f_e + f_d)$$

With Lagrangian, We Consider Dynamics as an Optimization Process

- With some math works, we know simulating such dynamics with Implicit Euler time integration can be formulated as an optimization problem of Incremental Potential (IP).

$$\min \int_{t_1}^{t_2} L(x, \dot{x}) dt = \min \int_{t_1}^{t_2} \left(\frac{1}{2} \dot{x}^\top M \dot{x} - \Psi(x) + x^\top (f_e + f_d) \right) dt$$



$$x^{t+1} = \arg \min_x E(x, x^t, v^t), \quad x^\tau \in \mathcal{A}, \quad \tau \in [t, t+1]$$

$$E(x, x^t, v^t) = \frac{1}{2} (x - \hat{x})^\top M (x - \hat{x}) - h^2 x^\top f_d + h^2 \Psi(x)$$

simulation time step

$$\hat{x} = x^t + h v^t + h^2 M^{-1} f_e$$

Dynamics as an Optimization Process

- The optimization problem with intersection-free constraints

$$x^{t+1} = \arg \min_x E(x, x^t, v^t), \quad s.t. \ d_k(x) > 0 \quad \forall k$$

- The optimization problem is difficult to solve...
 - It involves non-linear functions (e.g. $\Psi(x)$)
 - It becomes computationally inefficient for handling contact constraints (too large k)
 - It involves non-smooth functions (e.g. f_d)

Contact Modeling is Hard

- Problem 1: How to model friction, which exerts forces only when surfaces contact, while preserving intersection-free constraints?
- Problem 2: How to simulate contact / collision accurately while preserving intersection-free constraints?
- Problem 3: How to solve the hard optimization problem while involving frictional forces and intersection-free constraints?

Incremental Potential Contact: Intersection- and Inversion-free, Large-Deformation Dynamics

MINCHEN LI, University of Pennsylvania & Adobe Research

ZACHARY FERGUSON and TESEO SCHNEIDER, New York University

TIMOTHY LANGLOIS, Adobe Research

DENIS ZORIN and DANIELE PANOZZO, New York University


CHENFANFU JIANG, University of Pennsylvania

DANNY M. KAUFMAN, Adobe Research



How to solve the hard optimization problem with intersection-free constraints?

- Idea: Convert constraint optimization problems into unconstrained optimization problems
- Construct a continuous barrier energy b that approximates the non-smooth contact constraints. We augment the potential $E(x, x^t, v^t)$ with a sum of these barriers

$$B_t(x) = E(x, x^t, v^t) + \kappa \sum_k b(d_k(x))$$


The barrier energy creates a highly localized repulsion force, affecting motion only when primitives are close to collision, and vanishing if primitives are a small user-specified distance apart

How to solve the hard optimization problem with intersection-free constraints?

- Idea: Convert constraint optimization problems into unconstrained optimization problems

Constraint optimization:

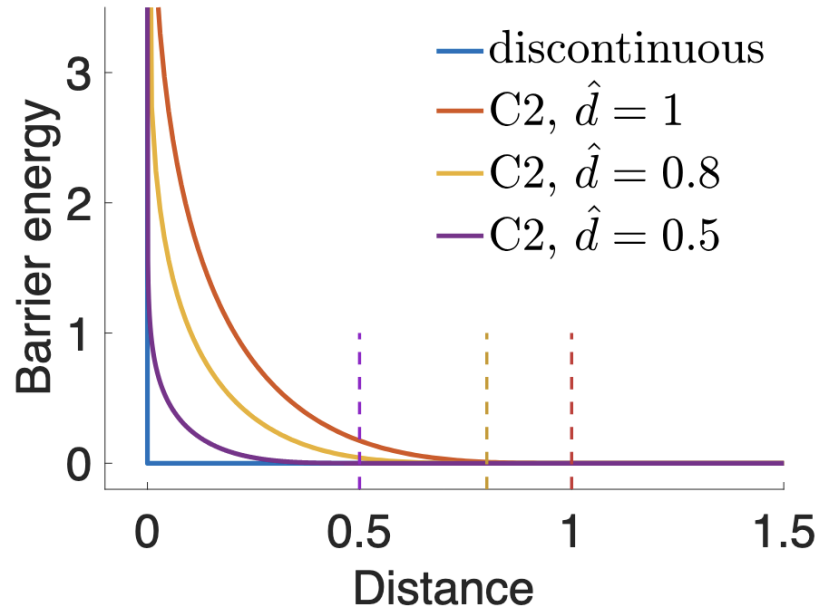
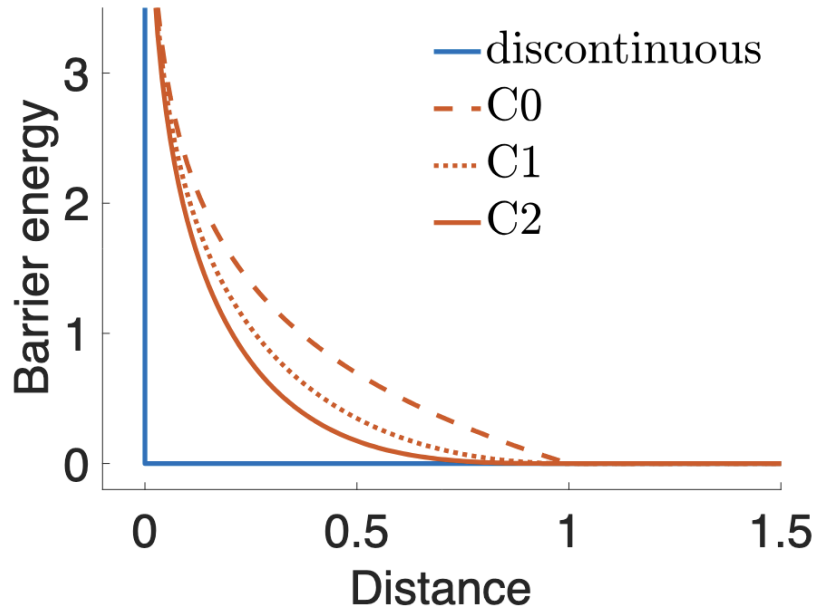
$$x^{t+1} = \arg \min_x E(x, x^t, v^t), \quad s.t. \ d_k(x) > 0 \quad \forall k$$



Unconstrained optimization:

$$x^{t+1} = \arg \min_x B_t(x) = \arg \min_x E(x, x^t, v^t) + \kappa \sum_k b(d_k(x))$$

What Could be the Barrier Energy b ?



- The function should be smooth
- The function should prevent intersection: the energy should be infinitely large when two entities contact

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

\hat{d} : the the maximum distance at which contact repulsions can act

How to Model Frictional Contact?

- Note that, frictional forces as two modes: sliding and static friction
- How to describe sliding?
 - Define an oriented sliding bases $T_k(\mathbf{x})$ at contact k . We'll explain later
 - The local relative sliding displacement can be described as $\mathbf{u}_k = T_k(\mathbf{x})(\mathbf{x} - \mathbf{x}^t)$
 - The sliding velocity is $\mathbf{v}_k = \dot{\mathbf{u}}_k/h$
- We can define frictional force as:

$$F_k(\mathbf{x}, \lambda) = T_k(\mathbf{x}) \arg \min_{\beta} \beta^\top \mathbf{v}_k$$

friction coefficient

$$s. t. \quad \|\beta\| \leq \mu \lambda_k$$

contact force magnitude

How to Model Frictional Contact?

- We can define frictional force as:

$$F_k(x, \lambda) = T_k(x) \arg \min_{\beta} \beta^\top v_k \quad s. t. \quad \|\beta\| \leq \mu \lambda_k$$

- When there is sliding $\|v_k\| > 0$, $F_k(x, \lambda) = -\mu \lambda_k T_k(x) \frac{u_k}{\|u_k\|}$
- When there is sticking $\|v_k\| = 0$, $F_k(x, \lambda) = -\mu \lambda_k T_k(x) f$, where f takes any value in the closed 2D unit disk

What is an Oriented Sliding Bases $T_k(\mathbf{x})$

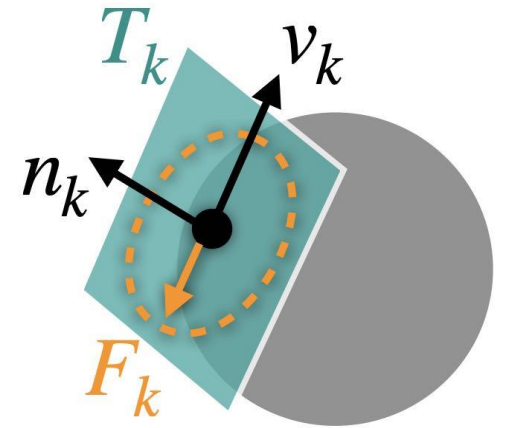
Example 9.1.1 (Particle Sliding on Sphere). For a particle with velocity $\mathbf{v}_p \in \mathbb{R}^3$ moving on the surface of a sphere with velocity $\mathbf{v}_s \in \mathbb{R}^3$ (no rotation), the relative sliding velocity \mathbf{v}_k here can be calculated as

$$\mathbf{v}_k = (\mathbf{v}_p - \mathbf{v}_s) - \mathbf{n}_k \cdot (\mathbf{v}_p - \mathbf{v}_s) \mathbf{n}_k = (\mathbf{I}_3 - \mathbf{n}_k \mathbf{n}_k^T) (\mathbf{v}_p - \mathbf{v}_s).$$

If we stack the velocity of the particle and the sphere for this system to obtain $\mathbf{v} = [\mathbf{v}_p^T, \mathbf{v}_s^T]^T \in \mathbb{R}^6$, we now know that T_k is simply

$$T_k(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_3 - \mathbf{n}_k(\mathbf{x}) \mathbf{n}_k(\mathbf{x})^T \\ \mathbf{n}_k(\mathbf{x}) \mathbf{n}_k(\mathbf{x})^T - \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{6 \times 3}.$$

For more general cases like mesh-mesh contact, the form of T_k only varies in how the relative velocity at the contact point pair k is related to the velocity at the simulated nodes.



How to Model Frictional Contact?

- We can define frictional force as:

$$F_k(x, \lambda) = T_k(x) \arg \min_{\beta} \beta^\top v_k \quad \text{s. t. } \|\beta\| \leq \mu \lambda_k$$

- When there is sliding $\|v_k\| > 0$, $F_k(x, \lambda) = -\mu \lambda_k T_k(x) \frac{u_k}{\|u_k\|}$
- When there is sticking $\|v_k\| = 0$, $F_k(x, \lambda) = -\mu \lambda_k T_k(x) f$, where f takes any value in the closed 2D unit disk

F_k is non-smooth!

Smoothed Frictional Contact

- Non-smoothed friction:
 - When there is sliding $\|v_k\| > 0$, $F_k(x, \lambda) = -\mu\lambda_k T_k(x) \frac{u_k}{\|u_k\|}$
 - When there is sticking $\|v_k\| = 0$, $F_k(x, \lambda) = -\mu\lambda_k T_k(x) f$, where f takes any value in the closed 2D unit disk
- An equivalent expression:

$$F_k(x, \lambda) = -\mu\lambda_k T_k(x) f(\|u_k\|) s(\|u_k\|)$$

where $s(\|u_k\|) = \frac{u_k}{\|u_k\|}$ when $\|u_k\| > 0$ otherwise any value

and $f(\|u_k\|) = 1$ when $\|u_k\| > 0$ otherwise $f(\|u_k\|) \in [0, 1]$

Smoothed Frictional Contact

- A smoothed expression of friction:

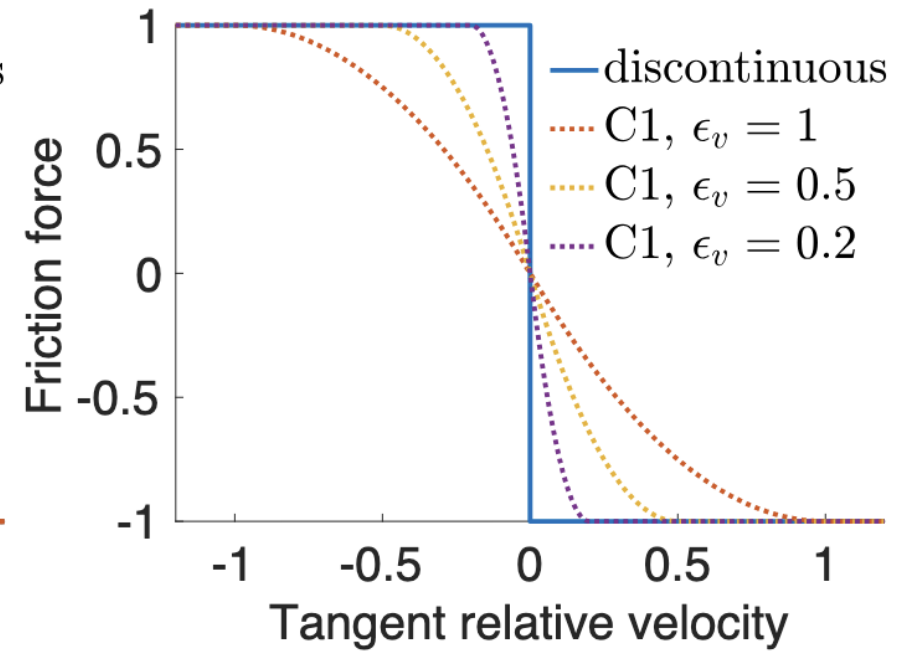
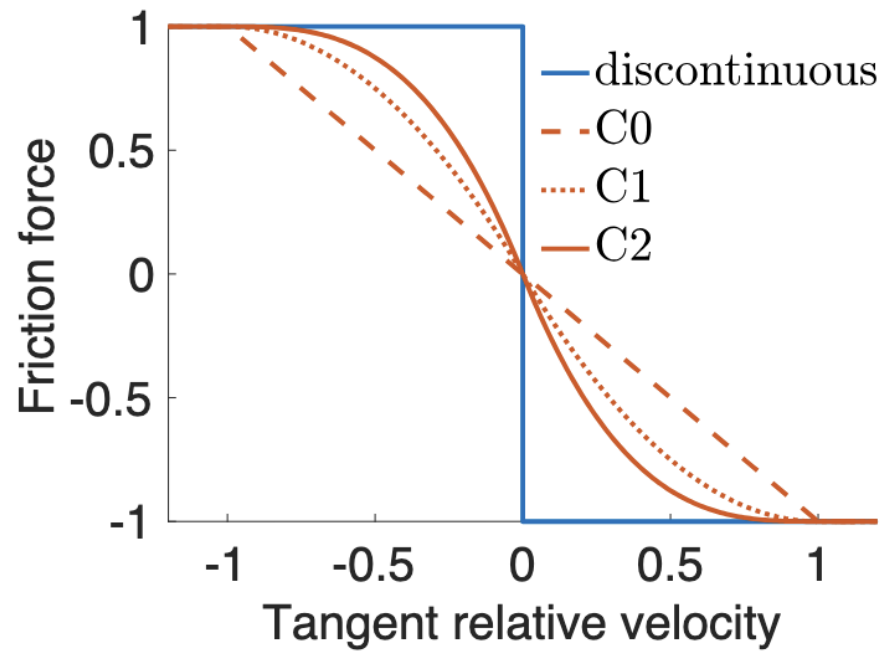
$$F_k(x, \lambda) = -\mu \lambda_k T_k(x) f_1(\|u_k\|) s(\|u_k\|)$$

where $s(\|u_k\|) = \frac{u_k}{\|u_k\|}$ when $\|u_k\| > 0$ otherwise any value

$$f_1(y) = \begin{cases} -\frac{y^2}{\epsilon_v^2 h^2} + \frac{2y}{\epsilon_v h}, & y \in (0, h\epsilon_v) \\ 1, & y \geq h\epsilon_v, \end{cases}$$

velocity $v_k = \frac{u_k}{h}$ below ϵ_v is treated static

Smoothed Frictional Contact



Smoothed Frictional Contact

- A smoothed expression of friction:

$$F_k(\mathbf{x}, \lambda) = -\mu\lambda_k T_k(\mathbf{x}) f_1(\|u_k\|) s(\|u_k\|)$$

- We want to derive a potential D_k such that $F_k(\mathbf{x}) = -\nabla_{\mathbf{x}} D_k(\mathbf{x})$
- If we decouple T_k and λ_k from \mathbf{x} , using T^n and λ^n derived from previous time step n , we obtain a simple and compact frictional potential:

$$D_k(\mathbf{x}) = \mu\lambda_k^n f_0(\|u_k\|)$$

f_0 is given by $f_0' = f_1$ and $f_0(\epsilon_v h) = \epsilon_v h$

Check the paper for detailed derivation!

Contact Modeling is Hard

- ✓ Problem 1: How to model friction, which exerts forces only when surfaces contact, while preserving intersection-free constraints?
- Problem 2: How to simulate contact / collision accurately while preserving intersection-free constraints?
- ⊗ Problem 3: How to solve the hard optimization problem while involving frictional forces and intersection-free constraints?

Newton's Method in Optimization

- We can use Newton's method to solve the optimization problem. Let's consider a scalar optimization problem:

$$\min_x f(x)$$

- Let x_k be the guess at iteration k . How can we obtain a better guess at the next iteration? By Taylor's expansion, we have:

$$\begin{aligned} f(x_k + t) &\approx f(x_k) + f'(x_k)t + \frac{1}{2}f''(x_k)t^2 \\ \Rightarrow \frac{df(x_k + t)}{dt} = 0 &\Rightarrow \frac{d\left(f(x_k) + f'(x_k)t + \frac{1}{2}f''(x_k)t^2\right)}{dt} = 0 \\ &\Rightarrow f'(x_k) + f''(x_k)t = 0 \Rightarrow t = -\frac{f'(x_k)}{f''(x_k)} \end{aligned}$$

Newton's Method in Optimization

- We can use Newton's method to solve the optimization problem. Let's consider a high-dimensional optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x})$$

- Let \mathbf{x}_k be the guess at iteration k . How can we obtain a better guess at the next iteration? By Taylor's expansion, we have:

$$\Rightarrow \mathbf{t} = -\overset{\vdots}{\boxed{H^{-1}}}\nabla f(\mathbf{x})$$

Hessian matrix, capturing the information of 2nd order derivative

Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{t}$

Contact Modeling is Hard

- ✓ Problem 1: How to model friction, which exerts forces only when surfaces contact, while preserving intersection-free constraints?
- ❓ Problem 2: How to simulate contact / collision accurately while preserving intersection-free constraints?
- ✓ Problem 3: How to solve the hard optimization problem while involving frictional forces and intersection-free constraints?

Backtracking Line Search

Method 3.2.1 (Backtracking Line Search). Given a descent direction, we can find a reasonably large α by simply halving it starting from 1 until the energy at the new location is smaller than the current (see [Algorithm 3.2.1](#)).

Algorithm 3.2.1 (The Backtracking Line Search Algorithm).

Result: α

```
1  $\alpha \leftarrow 1$ ;  
2 while  $E(x^i + \alpha p) > E(x^i)$  do  
3    $\alpha \leftarrow \alpha/2$ ;
```

Algorithm 1 Barrier Aware Projected Newton

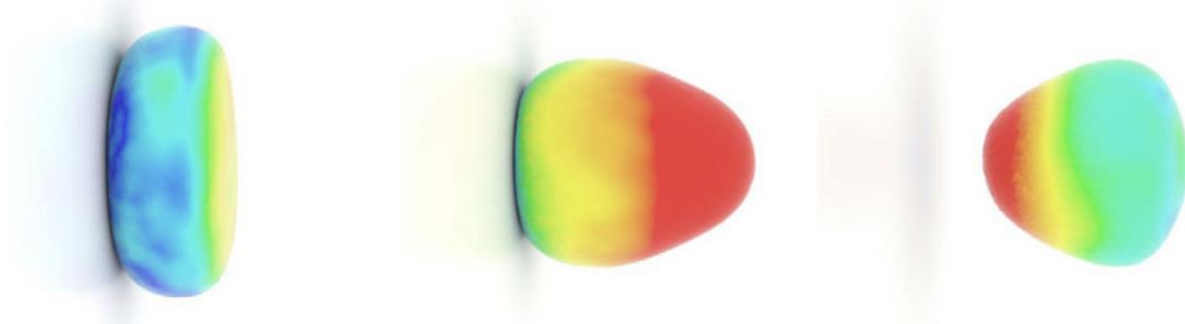
```
1: procedure BARRIERAWAREPROJECTEDNEWTON( $x^t, \epsilon$ )
2:    $x \leftarrow x^t$ 
3:    $\hat{C} \leftarrow \text{ComputeConstraintSet}(x, \hat{d})$  ▷ Section 4.6, 6.1
4:    $E_{\text{prev}} \leftarrow B_t(x, \hat{d}, \hat{C})$ 
5:    $x_{\text{prev}} \leftarrow x$ 
6:   do
7:      $H \leftarrow \text{SPDProject}(\nabla_x^2 B_t(x, \hat{d}, \hat{C}))$  ▷ Section 4.3
8:      $p \leftarrow -H^{-1} \nabla_x B_t(x, \hat{d}, \hat{C})$ 
9:     // CCD line search: ▷ Section 4.4
10:     $\alpha \leftarrow \min(1, \text{StepSizeUpperBound}(x, p, \hat{C}))$ 
11:    do
12:       $x \leftarrow x_{\text{prev}} + \alpha p$ 
13:       $\hat{C} \leftarrow \text{ComputeConstraintSet}(x, \hat{d})$ 
14:       $\alpha \leftarrow \alpha/2$ 
15:    while  $B_t(x, \hat{d}, \hat{C}) > E_{\text{prev}}$  Actually there are more engineering
16:     $E_{\text{prev}} \leftarrow B_t(x, \hat{d}, \hat{C})$  details to make things work...
17:     $x_{\text{prev}} \leftarrow x$ 
18:    Update  $\kappa$ , BCs and equality constraints ▷ Supplemental
19:  while  $\frac{1}{h} \|p\|_\infty > \epsilon_d$ 
20:  return  $x$ 
```

Results

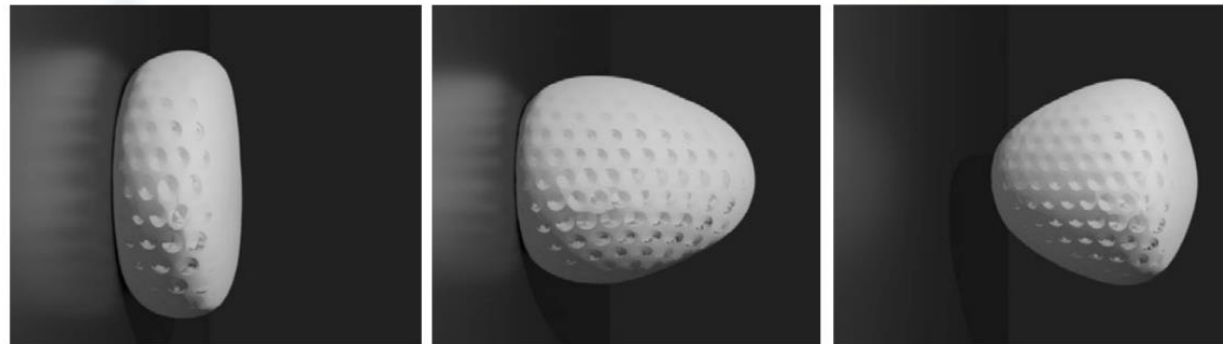
video footage



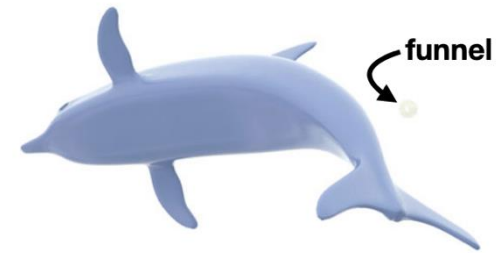
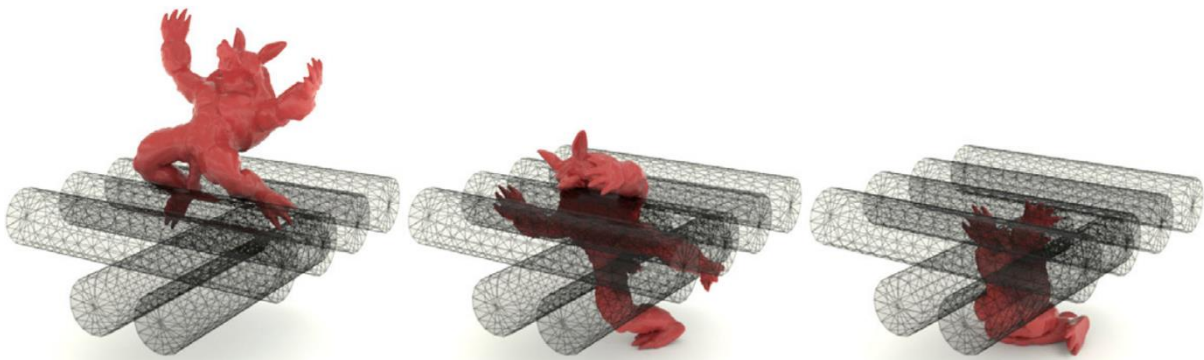
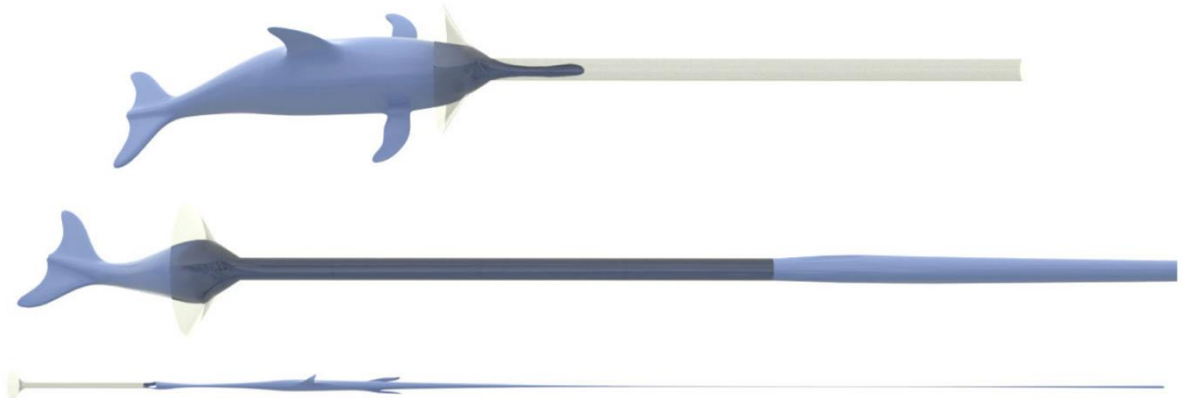
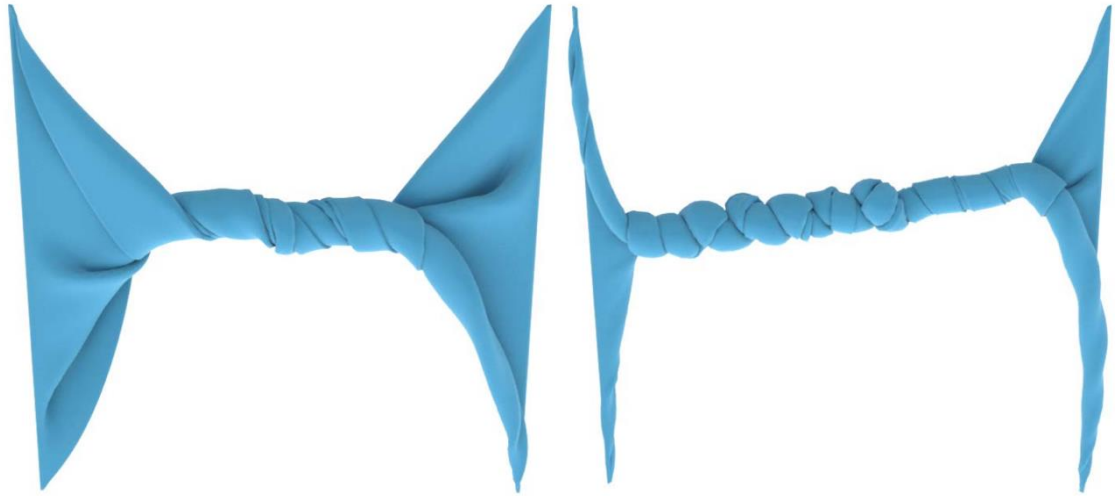
velocity magnitude (m/s)



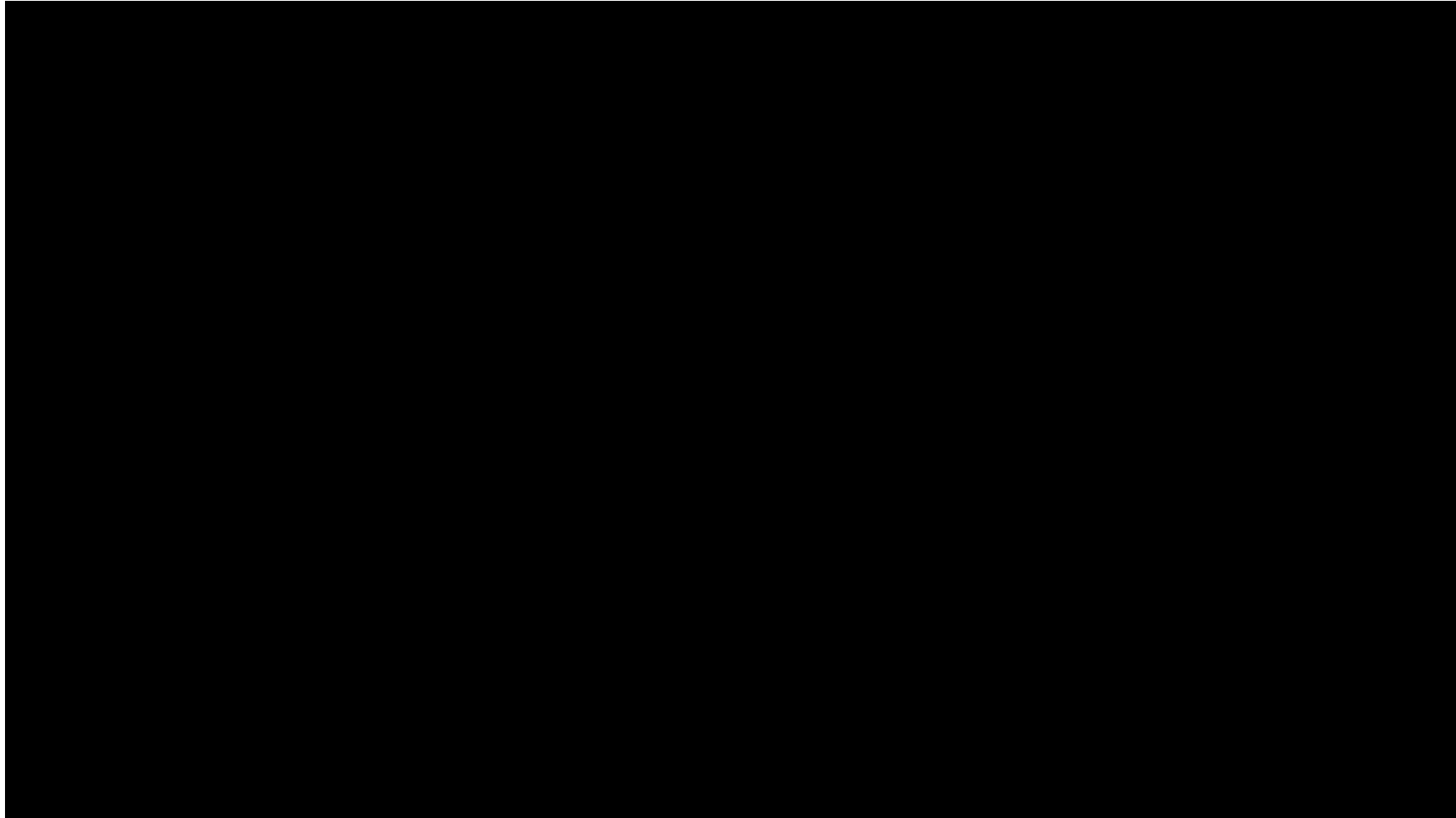
simulation



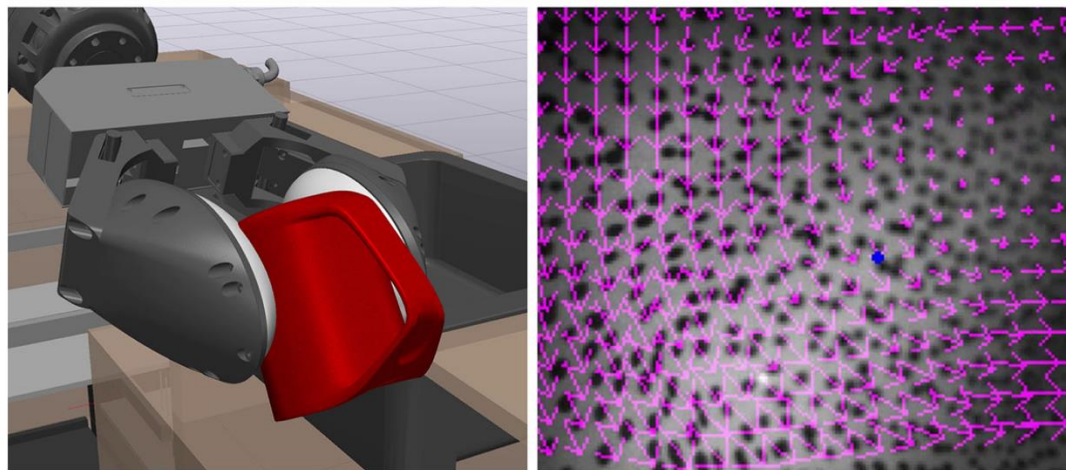
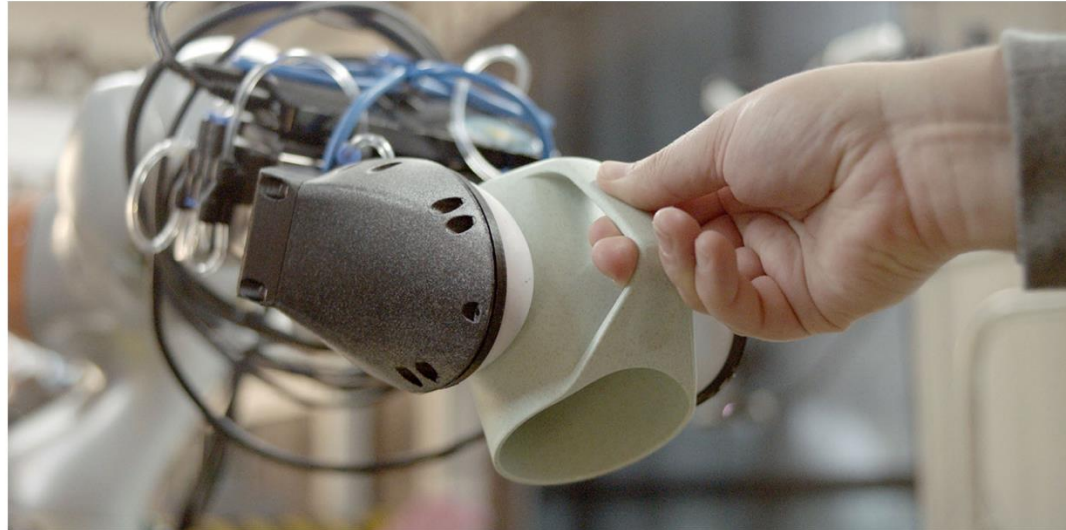
Results



Why Do We Need Contact Modeling? Robot Manipulation is All About Contact



Why Do We Need Contact Modeling? Simulation of Tactile Sensing



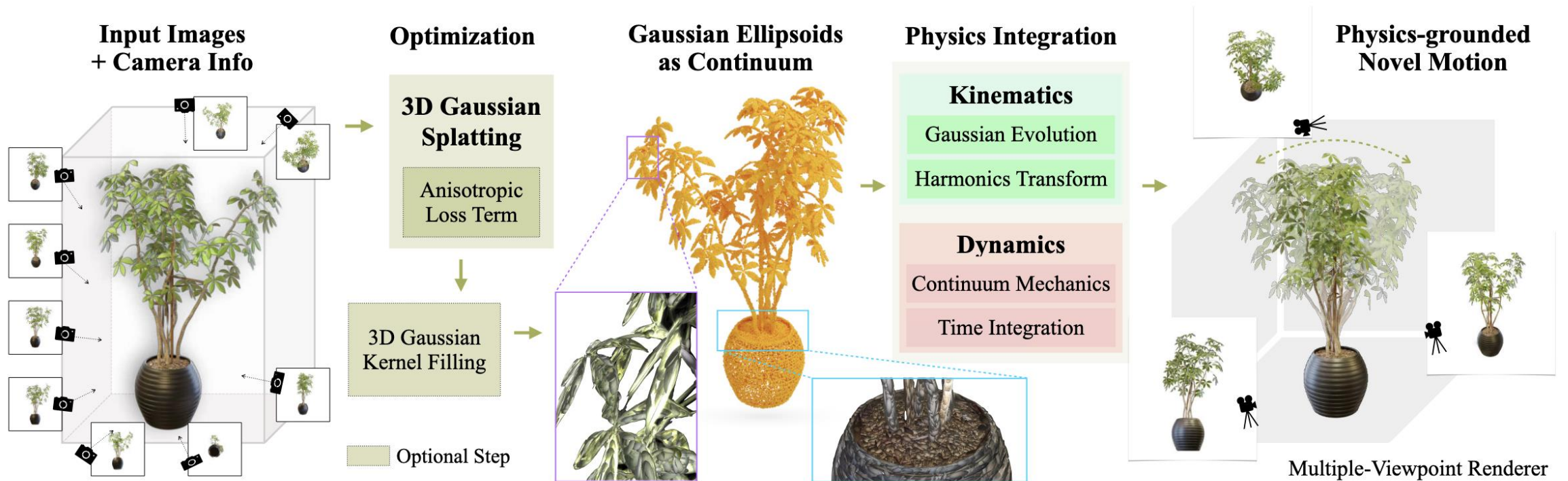
Content

- Lagrangian Mechanics
- Incremental Potential Contact Modeling
- Material Point Methods

How about Simulation of Complex Materials or Large Deformations?



Our Goal: Physically Reconstructing 3D Objects of Complex Materials or Large Deformations



Our Goal: Physically Reconstructing 3D Objects of Complex Materials or Large Deformations



Material Point Method: A Physical Model for Continuum Mechanics

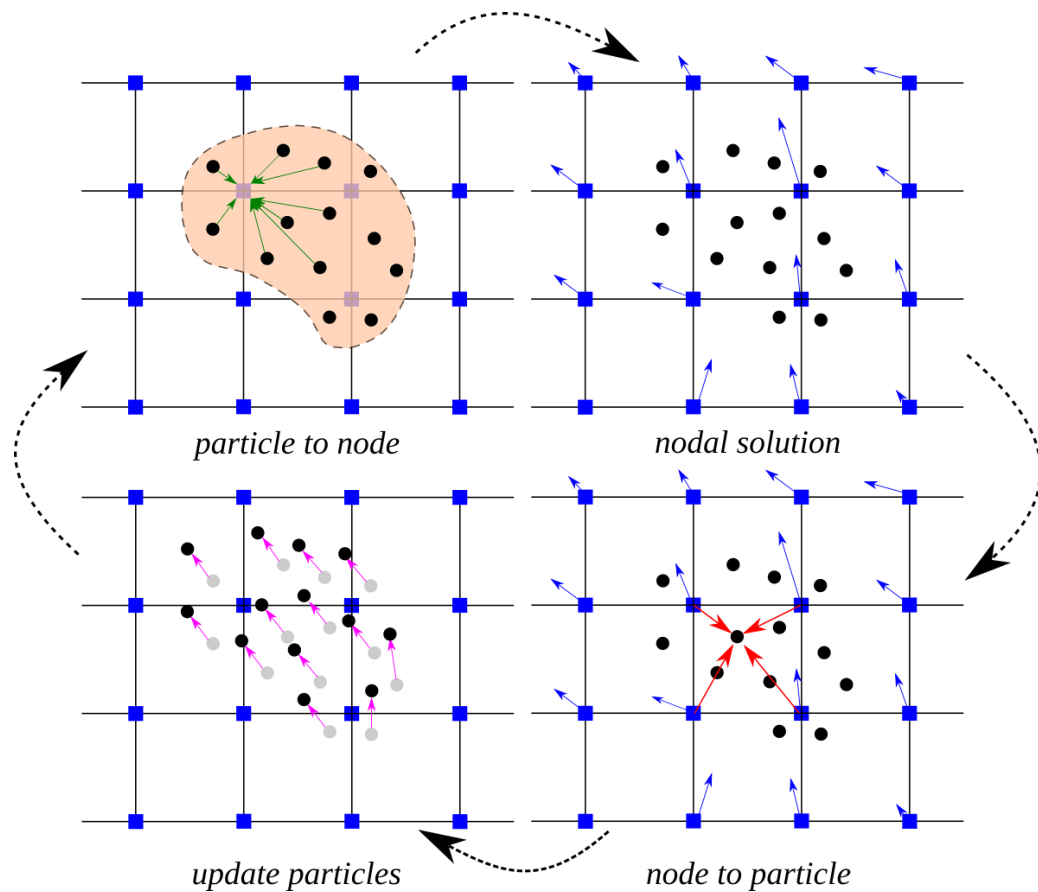


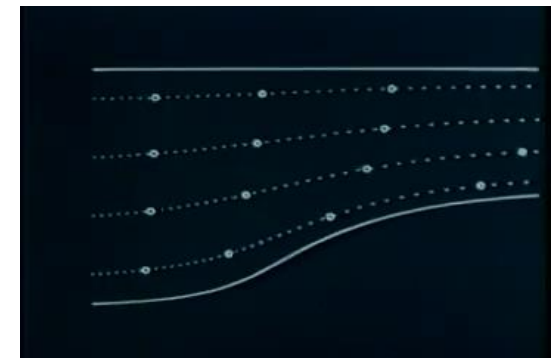
Image source
<https://www.geoelements.org/LearnMPM/mpm.html>

- A hybrid approach for modeling continuum mechanics
- Particles are Lagrangian representations: easier to track the motion
- Grids are Eulerian representations: easier to compute forces
- Lagrangian particle or Eulerian grid views the motion from different coordinate systems



Particle motion

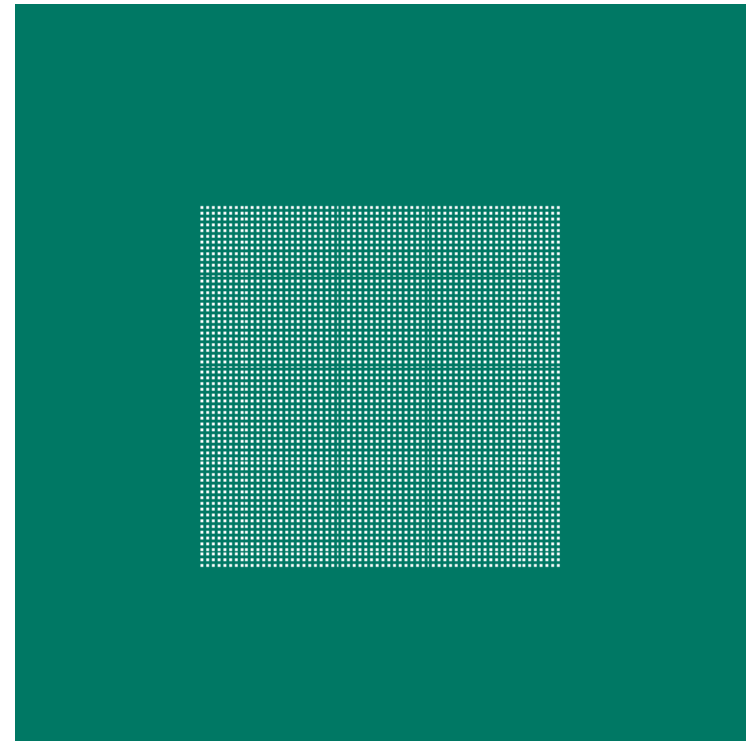
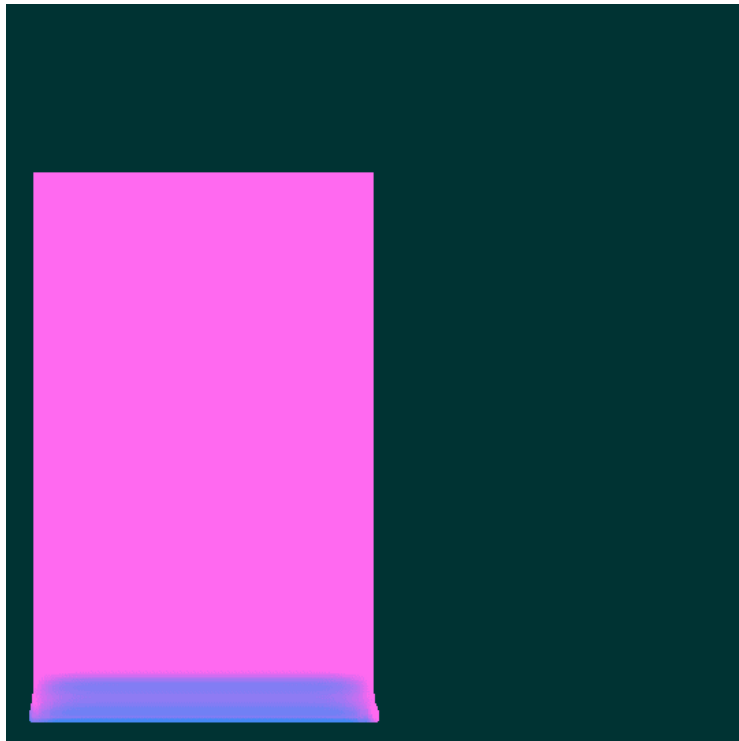
63 (Lagrangian representations)



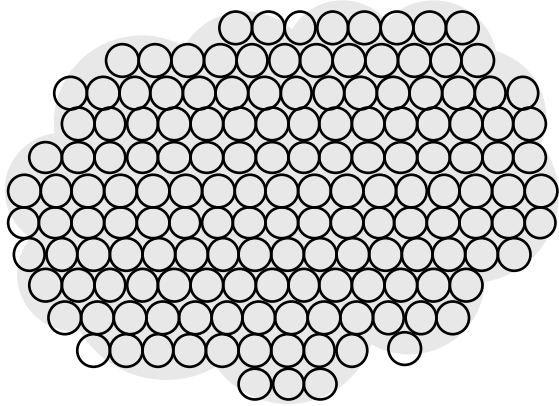
Grid motion

(Eulerian representations)

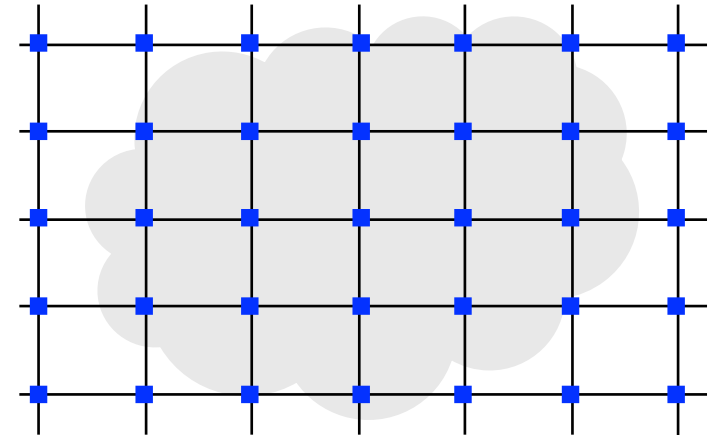
Material Point Method: A Physical Model for Continuum Mechanics



MPM Particles vs. Eulerian Grids



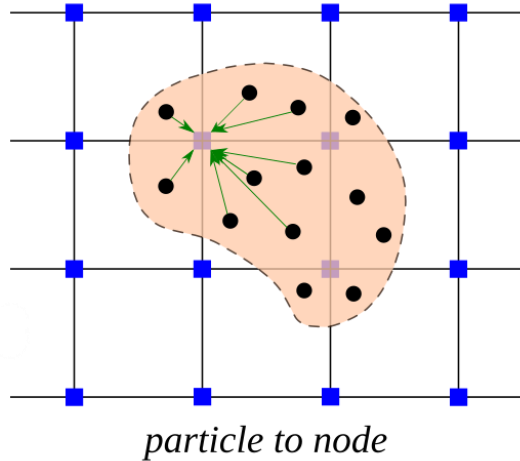
- MPM Particles are not individual particles, molecules, atoms or little spheres, but a continuous piece of material
- Particle motions are easier to track
- Particle motions are easier to advect
- Particles representations trivially enforce mass conservation



- Eulerian grids are fixed point in space where motions of particles passing through are measured
- It's easier to compute / exert forces on grids
- It's easier to handle collision / topological change / boundary conditions with grids
- Grid representations naturally handle spatial derivatives

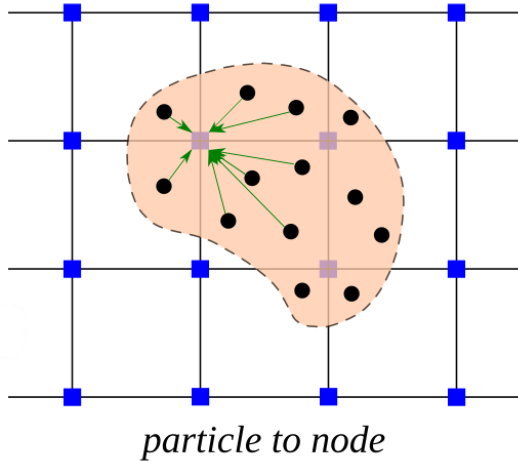
A Brief Overview of MPM

1. Aggregate particle motions to grid motions



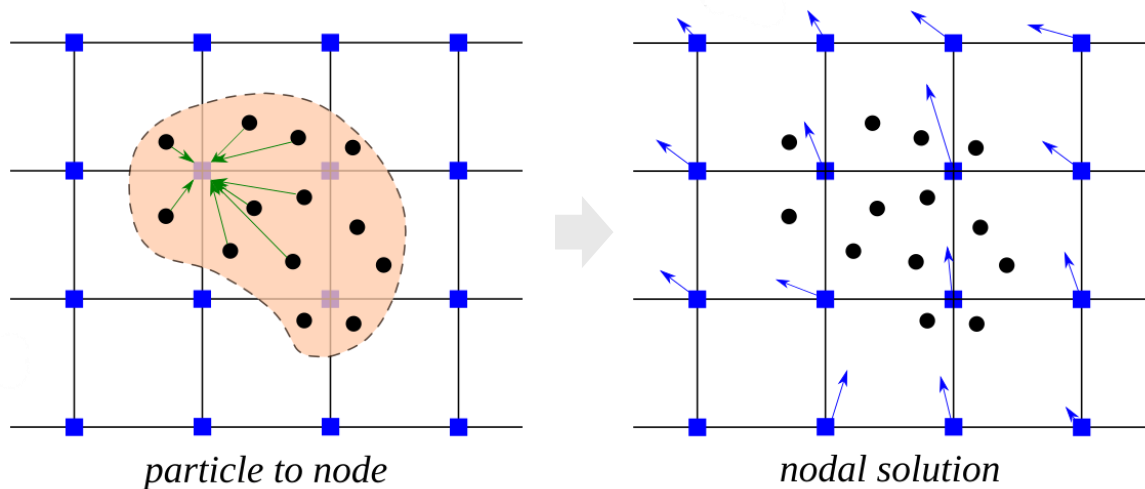
A Brief Overview of MPM

1. Aggregate particle motions to grid motions
2. Aggregate particle deformation to grid forces



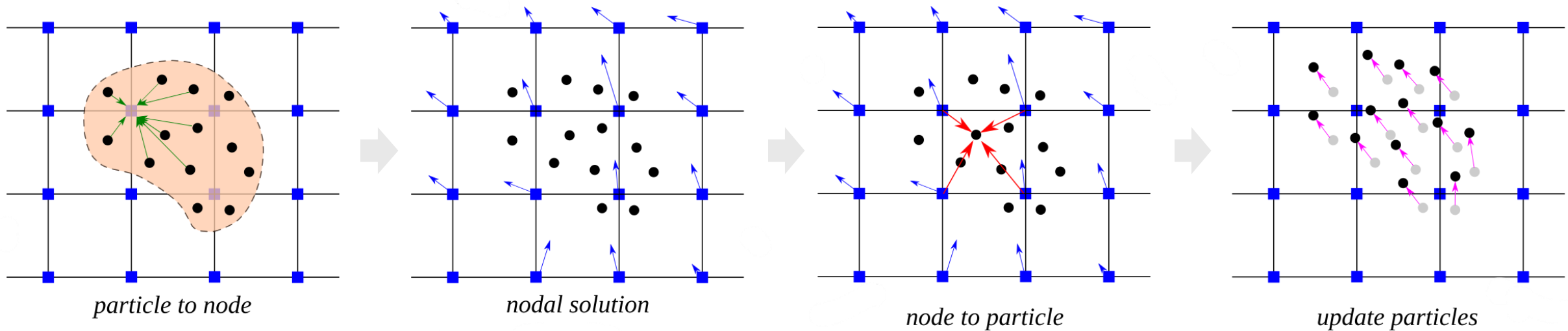
A Brief Overview of MPM

1. Aggregate particle motions to grid motions
2. Aggregate particle deformation to grid forces
3. Update grid motions based on calculated stress / strain and external forces



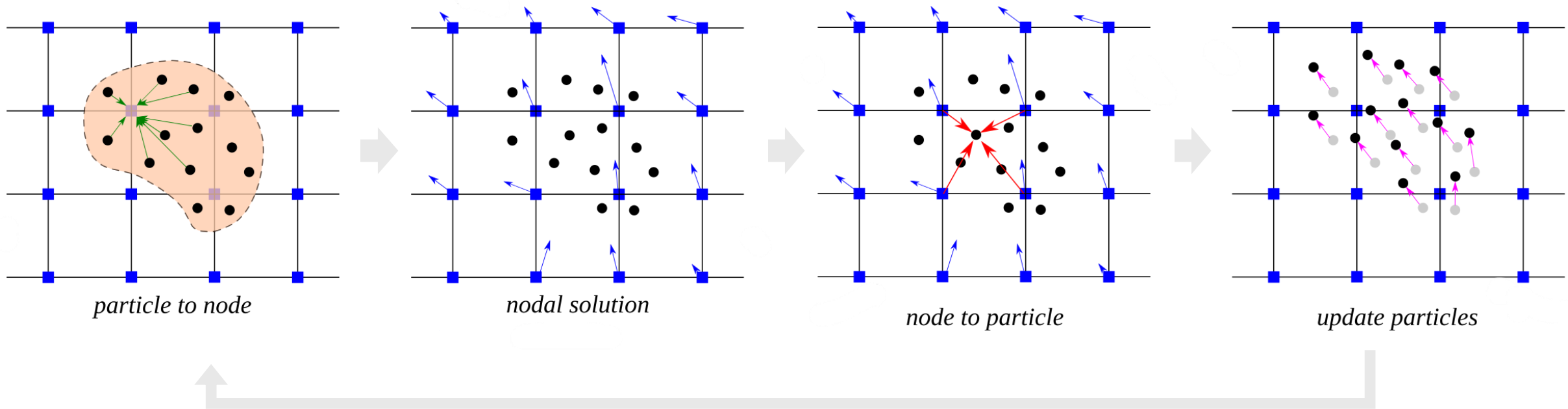
A Brief Overview of MPM

1. Aggregate particle motions to grid motions
2. Aggregate particle deformation to grid forces
3. Update grid motions based on calculated stress / strain and external forces
4. Update particle motions based on grid motions



A Brief Overview of MPM

1. Aggregate particle motions to grid motions
2. Aggregate particle deformation to grid forces
3. Update grid motions based on calculated stress / strain and external forces
4. Update particle motions based on grid motions
5. Repeat 1.



Let's Dig Into some Details

Continuum Mechanics

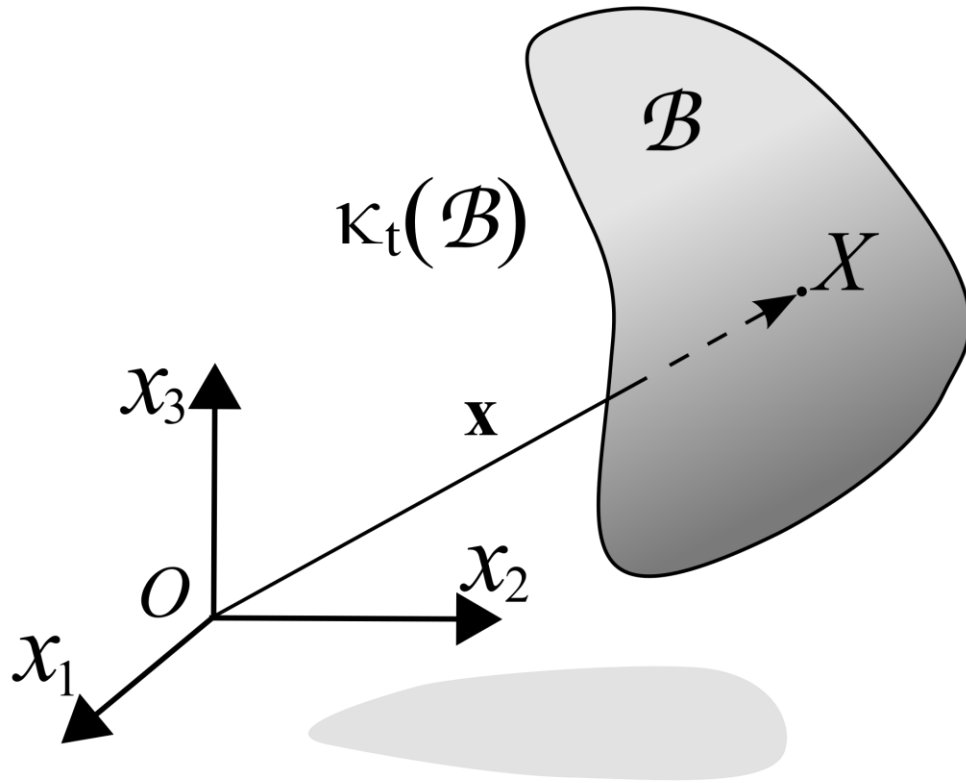
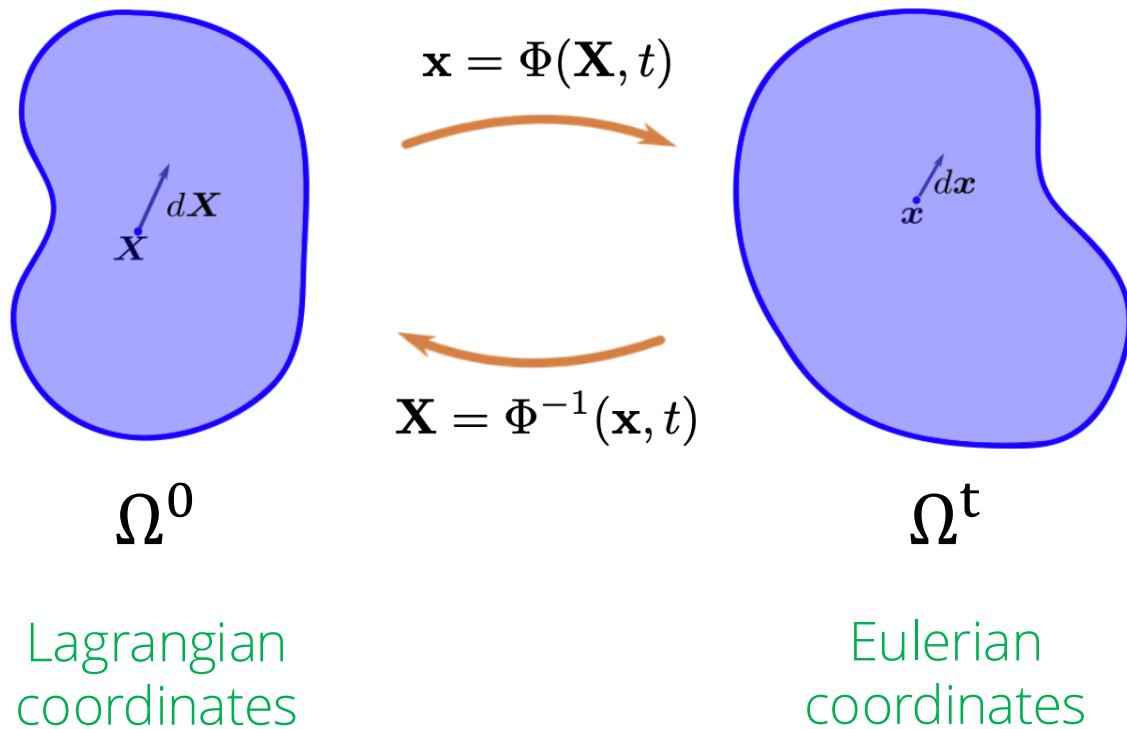


Image source: wiki

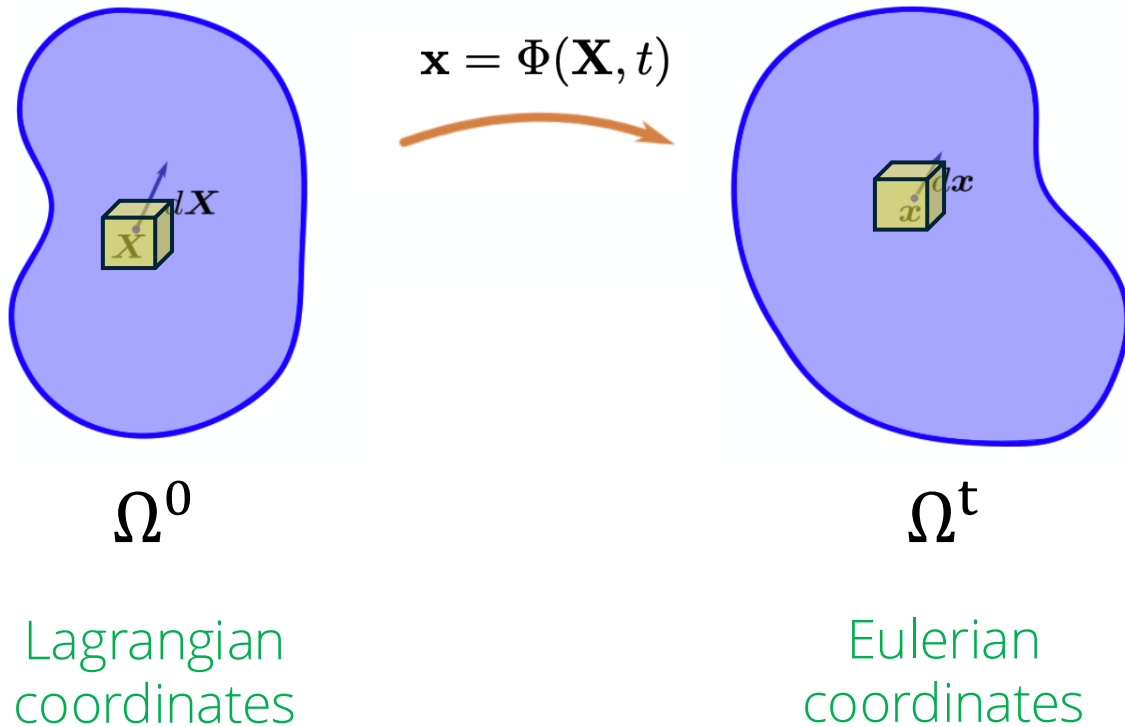
- A continuum is an area that can keep being divided and divided infinitely, while its particulate nature is ignored.
- This simplification enables studying the movement of matter on scales larger than the distances between particles
- For example, it studies the motion of a piece of materials, rather than the motion of individual atoms

Kinematics of Continuum Materials



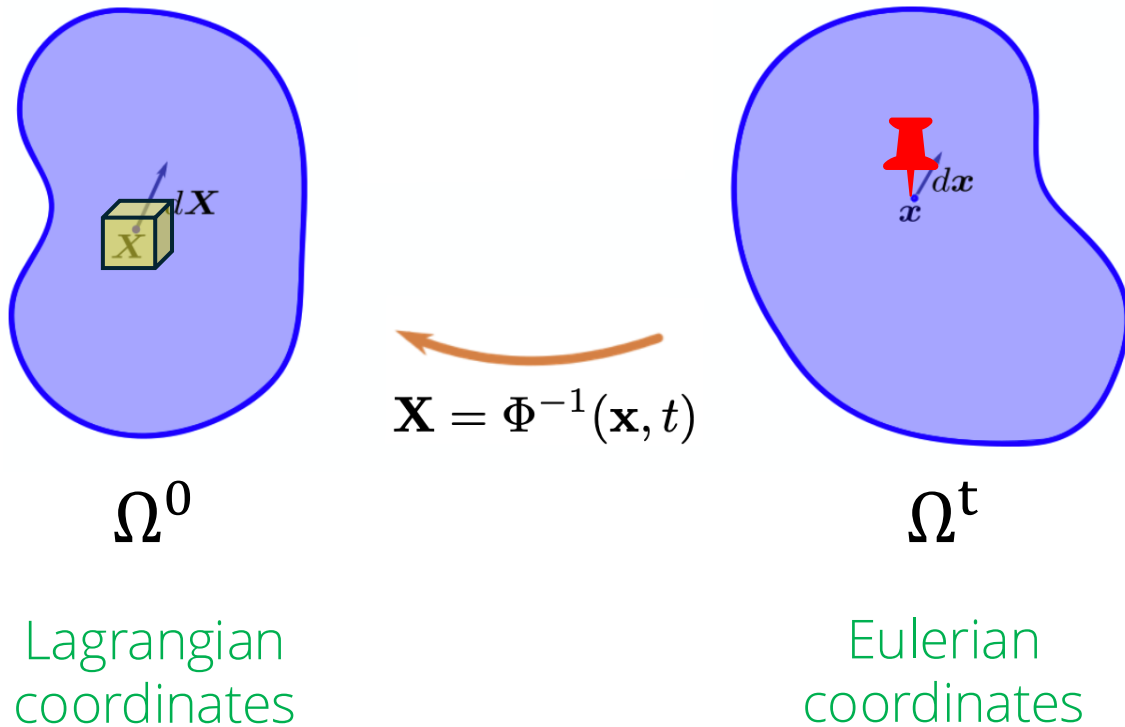
- In continuum mechanics, the deformation is usually represented with the material space \mathbf{X} , the world space \mathbf{x} and a deformation map $\phi(\mathbf{X}, t)$
- The material space \mathbf{X} denotes the coordinate that identifies **which piece of material**. By convention, \mathbf{X} is the position that piece occupied at time $t = 0$
- The world space \mathbf{x} denotes the coordinate that identifies a position in the space
- The deformation map $\mathbf{x} = \phi(\mathbf{X}, t)$ describes the chunk of material \mathbf{X} is currently sitting at location \mathbf{x} at time t

Lagrangian Kinematics of Continuum Materials



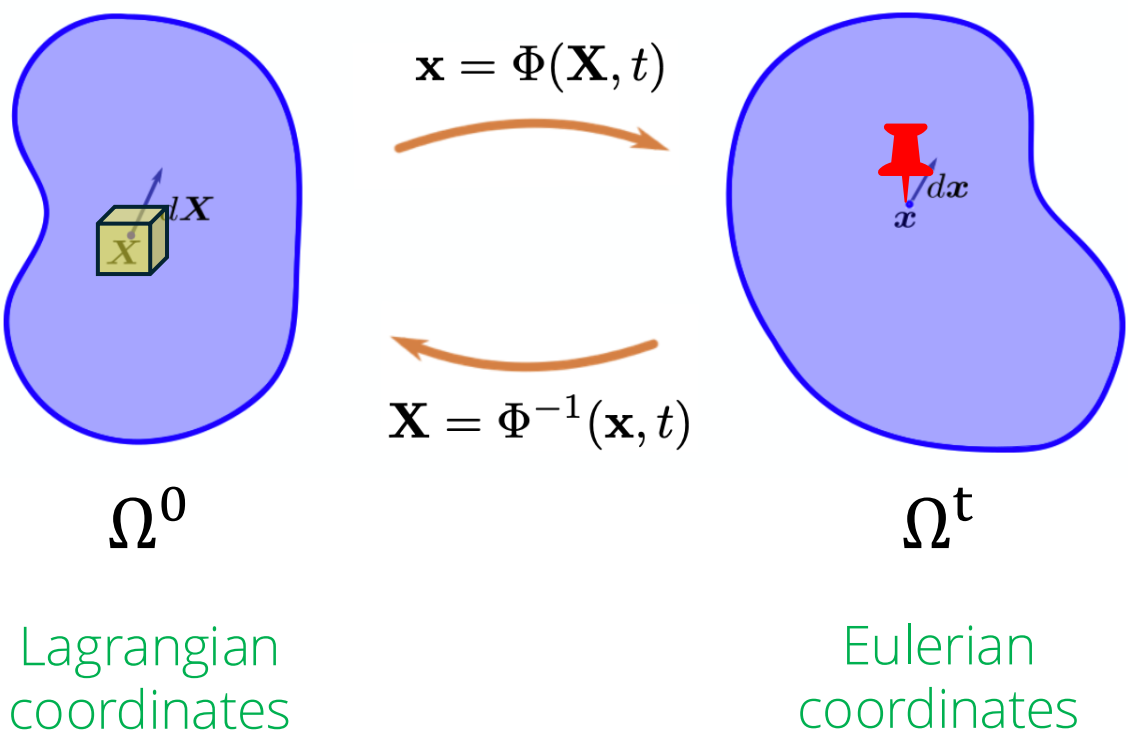
- The deformation map $\mathbf{x} = \Phi(\mathbf{X}, t)$ describes the chunk of material \mathbf{X} is currently sitting at location \mathbf{x} at time t
- With the deformation map, we naturally derive kinematics in Lagrangian coordinates:
 - Velocity: $\mathbf{V}(\mathbf{X}, t) = \frac{\partial \Phi(\mathbf{X}, t)}{\partial t}$
 - Acceleration: $\mathbf{A}(\mathbf{X}, t) = \frac{\partial^2 \Phi(\mathbf{X}, t)}{\partial t^2} = \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t}$
- In other words, we measure the velocity and acceleration on a fixed particle

Eulerian Kinematics of Continuum Materials



- The inverse deformation map $\mathbf{X} = \phi^{-1}(\mathbf{x}, t)$ traces back the chunk of material \mathbf{X} is currently sitting at location \mathbf{x} at time t
- The kinematics in Eulerian coordinates is non-intuitive:
 - Velocity: $\mathbf{v}(\mathbf{x}, t) = \frac{\partial \phi(\phi^{-1}(\mathbf{x}, t), t)}{\partial t}$
 - Acceleration: $\mathbf{a}(\mathbf{x}, t) = \frac{\partial^2 \phi(\phi^{-1}(\mathbf{x}, t), t)}{\partial t^2} \neq \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t}$
- In other words, we measure the velocity and acceleration at a fixed location \mathbf{x} at time t

Conversion between Lagrangian and Eulerian Kinematics



- From Eulerian to Lagrangian:
 - $v(\mathbf{x}, t) = V(\phi^{-1}(\mathbf{x}, t), t)$
 - $\mathbf{a}(\mathbf{x}, t) = \mathbf{A}(\phi^{-1}(\mathbf{x}, t), t)$
- From Lagrangian to Eulerian:
 - $V(\mathbf{X}, t) = v(\phi(\mathbf{X}, t), t)$
 - $\mathbf{A}(\mathbf{X}, t) = \mathbf{a}(\phi(\mathbf{X}, t), t)$
- Lagrangian acceleration is the derivative of Lagrangian velocity with respect to \mathbf{t} :

$$\mathbf{A}(\mathbf{X}, t) = \frac{\partial V(\mathbf{X}, t)}{\partial t}$$

- Eulerian acceleration is not the derivative of Eulerian velocity with respect to \mathbf{t} :

$$\mathbf{a}(\mathbf{x}, t) \neq \frac{\partial v(\mathbf{x}, t)}{\partial t}$$

Conversion between Lagrangian and Eulerian Kinematics

- Let's first derive Lagrangian acceleration with respect to Eulerian velocity:

$$\begin{aligned} A(\mathbf{X}, t) &= \frac{dV(\mathbf{X}, t)}{dt} = \frac{dv(\phi(\mathbf{X}, t), t)}{dt} \\ &= \frac{\partial v(\phi(\mathbf{X}, t), t)}{\partial t} + \frac{\partial v(\phi(\mathbf{X}, t), t)}{\partial \phi(\mathbf{X}, t)} \frac{\partial \phi(\mathbf{X}, t)}{\partial t} \\ &= \frac{\partial v(\phi(\mathbf{X}, t), t)}{\partial t} + \frac{\partial v(\phi(\mathbf{X}, t), t)}{\partial \mathbf{x}(\mathbf{X}, t)} \frac{\partial \phi(\mathbf{X}, t)}{\partial t} \end{aligned}$$

- Derive Eulerian acceleration from Lagrangian acceleration:

$$\mathbf{a}(\mathbf{x}, t) = A(\phi^{-1}(\mathbf{x}, t), t) = \frac{\partial v(\mathbf{x}, t)}{\partial t} + \frac{\partial v(\mathbf{x}, t)}{\partial \mathbf{x}} v(\mathbf{x}, t)$$

Conversion between Lagrangian and Eulerian Kinematics

- Derive Eulerian acceleration from Lagrangian acceleration:

$$\mathbf{a}(\mathbf{x}, t) = \mathbf{A}(\phi^{-1}(\mathbf{x}, t), t) = \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} + \frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x}, t)$$

- In other words, both temporal and spatial derivatives of velocity measured at location \mathbf{x} at time t attributes to the acceleration
- Imagine a **steady** river where the water speeds up as it enters a narrow channel.

At any fixed point in space the velocity never changes, so $\frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial t} = \mathbf{0}$. Yet a leaf floating downstream does **accelerate**, because it's being carried into regions of higher \mathbf{v} . That acceleration comes entirely from the spatial derivatives

$$\frac{\partial \mathbf{v}(\mathbf{x}, t)}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x}, t)$$

Take-away Message

- The motion can be measured in two coordinate systems: Lagrangian and Eulerian coordinates
- Lagrangian kinematics can be converted to Eulerian kinematics, and vice versa
- Eulerian acceleration results from both temporal and spatial derivatives of velocity measured at a fixed location at certain time

How to Describe Properties of Different Materials?



Cloth



Rope

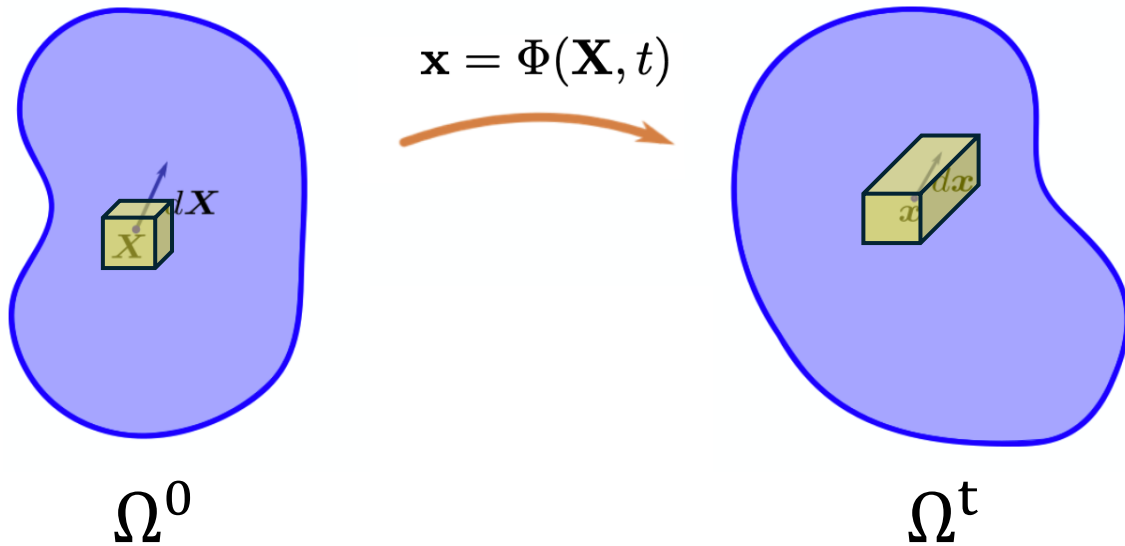


Fluid



Soft body

Deformation Gradient



- The deformation map $\mathbf{x} = \phi(\mathbf{X}, t)$ describes the chunk of material \mathbf{X} is currently sitting at location \mathbf{x} at time t
- The deformation gradient is defined as:

$$\mathbf{F}(\mathbf{X}, t) = \frac{\partial \phi(\mathbf{X}, t)}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}}$$

Deformation gradient describes the deformation of one axis relative to other axes

We Can Characterize Material-wise Dynamics using Strain Stress Relationship

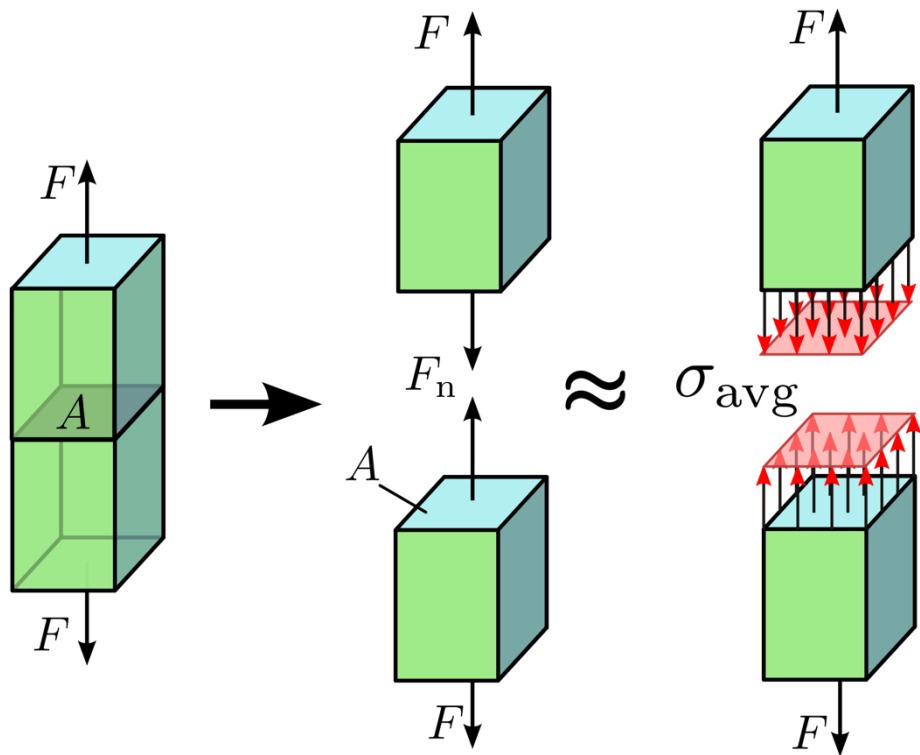


Image source: wikipedia

- Stress: the force per unit area on a body that tends to cause it to change shape

We Can Characterize Material-wise Dynamics using Strain Stress Relationship

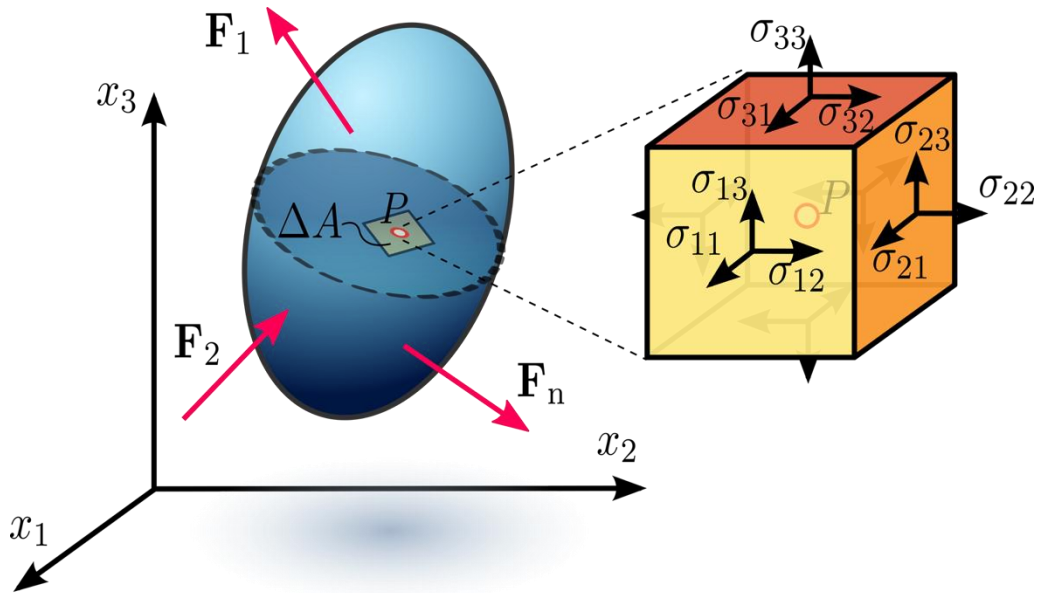


Image source: wikipedia

- Stress: the force per unit area on a body that tends to cause it to change shape
- Strain: the deformation or change in shape of a body subjected to applied forces
- First Piola-Kirchhoff stress \mathbf{P} can be derived as:

$$\mathbf{P} = \frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}}$$

while Cauchy stress $\boldsymbol{\sigma}$ is:

$$\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})} \frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}} \mathbf{F}^T$$

Strain energy density function $\Psi(\mathbf{F})$

- Each form of $\Psi(\mathbf{F})$ describe individual material properties
- For example, Neo-Hookean is excellent for modeling moderate to large, reversible strains in uniaxial tension, pure shear, and compression.

$$\Psi(\mathbf{F}) = \frac{\mu}{2} \left(\text{tr}(\mathbf{F}^T \mathbf{F}) - d \right) - \mu \log(J) + \frac{\lambda}{2} \log^2(J), \quad J = \det(\mathbf{F})$$

- For example, fixed corotated constitutive model is best for large deformation, large rotation, and non-linear elastic behavior of materials

$$\Psi(\mathbf{F}) = \hat{\Psi}(\boldsymbol{\Sigma}(\mathbf{F})) = \mu \sum_{i=1}^d (\sigma_i - 1)^2 + \frac{\lambda}{2} (J - 1)^2$$

$$\mathbf{F} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$$

Calculate SVD of \mathbf{F}

How to Model Dynamics of Different Materials?



Cloth



Rope



Fluid



Soft body

Governing Equations

Lagrangian view

Lagrangian mass density

$\det(\mathbf{F})$

$$\mathbf{R}(\mathbf{X}, t) \mathbf{J}(\mathbf{X}, t) = \mathbf{R}(\mathbf{X}, 0) \quad \text{Conservation of mass,}$$

$$\mathbf{R}(\mathbf{X}, 0) \frac{\partial \mathbf{V}}{\partial t} = \nabla^{\mathbf{X}} \cdot \mathbf{P} + \mathbf{R}(\mathbf{X}, 0) \mathbf{g} \quad \text{Conservation of momentum,}$$

Eulerian view

$$\mathbf{R}(\mathbf{X}, t) = \rho(\Phi(\mathbf{X}, t), t)$$

Eulerian mass density

$$\frac{D}{Dt} \rho(\mathbf{x}, t) + \rho(\mathbf{x}, t) \nabla^{\mathbf{x}} \cdot \mathbf{v}(\mathbf{x}, t) = 0 \quad \text{Conservation of mass,}$$

$$\rho(\mathbf{x}, t) \frac{D\mathbf{v}}{Dt} = \nabla^{\mathbf{x}} \cdot \boldsymbol{\sigma} + \rho(\mathbf{x}, t) \mathbf{g} \quad \text{Conservation of momentum,}$$

Calculating “Conservative of Momentum” is Challenging

- Conservation of momentum describes the change of velocity. However, we often do not have access to the derivatives of Piola-Kirchoff stress \mathbf{P} at every chunk of material \mathbf{X} . We need to approximate the formula

$$\mathbf{R}(\mathbf{X}, 0) \frac{\partial \mathbf{V}}{\partial t} = \nabla^{\mathbf{X}} \cdot \mathbf{P} + \mathbf{R}(\mathbf{X}, 0) \mathbf{g}$$

- Simplifications:
 - Ignore external forces (e.g. gravity)

$$\mathbf{R}(\mathbf{X}, 0) \frac{\partial \mathbf{V}}{\partial t} = \boxed{\nabla^{\mathbf{X}} \cdot \mathbf{P}}$$

This assumes the gradient of First Piola-Kirchoff stress \mathbf{P} is available at every \mathbf{X}

Representing “Conservative of Momentum” in a Weak Form

- Previously, we have

$$\mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}}{\partial t} = \nabla^{\mathbf{X}} \cdot \mathbf{P}$$

- With an arbitrary function $\mathbf{Q}(\cdot, t)$, we express the equation through integral:

$$\int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t} d\mathbf{X} = \int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \nabla^{\mathbf{X}} \cdot \mathbf{P} d\mathbf{X}$$

Representing “Conservative of Momentum” in a Weak Form

- Previously, we have

$$\mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}}{\partial t} = \nabla^{\mathbf{X}} \cdot \mathbf{P}$$

- With an arbitrary function $\mathbf{Q}(\cdot, t): \Omega^0 \rightarrow \mathbb{R}^d$, we express the equation through integral:

$$\int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t} d\mathbf{X} = \int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \nabla^{\mathbf{X}} \cdot \mathbf{P} d\mathbf{X}$$

$\mathbf{Q}(\mathbf{X}, t), \mathbf{V}(\mathbf{X}, t), \mathbf{X}$ are vectors
 \mathbf{P} is a tensor

Divergence

For a continuously differentiable vector field

$$\mathbf{F} = F_x \mathbf{i} + F_y \mathbf{j} + F_z \mathbf{k}$$

$$\begin{aligned} \nabla \cdot \mathbf{F} &= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \cdot (F_x, F_y, F_z) \\ &= \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \end{aligned}$$

For a continuously differentiable tensor field

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

$$\nabla \cdot \mathbf{A} = \begin{bmatrix} \frac{\partial A_{11}}{\partial x_1} + \frac{\partial A_{21}}{\partial x_2} + \frac{\partial A_{31}}{\partial x_3} \\ \frac{\partial A_{12}}{\partial x_1} + \frac{\partial A_{22}}{\partial x_2} + \frac{\partial A_{32}}{\partial x_3} \\ \frac{\partial A_{13}}{\partial x_1} + \frac{\partial A_{23}}{\partial x_2} + \frac{\partial A_{33}}{\partial x_3} \end{bmatrix}$$

Representing “Conservative of Momentum” in a Weak Form

- Previously, we have

$$\mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}}{\partial t} = \nabla^{\mathbf{X}} \cdot \mathbf{P}$$

- With an arbitrary function $\mathbf{Q}(\cdot, t)$, we express the equation through integral:

$$\int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \mathbf{R}(\mathbf{X}, t) \frac{\partial \mathbf{V}(\mathbf{X}, t)}{\partial t} d\mathbf{X} = \int_{\Omega^0} \mathbf{Q}(\mathbf{X}, t) \nabla^{\mathbf{X}} \cdot \mathbf{P} d\mathbf{X}$$

$\mathbf{Q}(\mathbf{X}, t), \mathbf{V}(\mathbf{X}, t), \mathbf{X}$ are vectors
 \mathbf{P} is a tensor



$$\int_{\Omega^0} Q_i(\mathbf{X}, t) R(\mathbf{X}, t) A_i(\mathbf{X}, t) d\mathbf{X} = \int_{\Omega^0} Q_i(\mathbf{X}, t) \boxed{P_{ij,j}(\mathbf{X}, t)} d\mathbf{X}$$

$\frac{\partial P_{ij}(\mathbf{X}, t)}{\partial X_j}$

Representing “Conservative of Momentum” in a Weak Form

$$\begin{aligned}
 \int_{\Omega^0} Q_i(\mathbf{X}, t) R(\mathbf{X}, 0) A_i(\mathbf{X}, t) d\mathbf{X} &= \int_{\Omega^0} Q_i(\mathbf{X}, t) \frac{\partial P_{ij}(\mathbf{X}, t)}{\partial X_j} d\mathbf{X} \\
 &= \int_{\Omega^0} \left(\frac{\partial (Q_i(\mathbf{X}, t) P_{ij}(\mathbf{X}, t))}{\partial X_j} - Q_{i,j}(\mathbf{X}, t) P_{ij}(\mathbf{X}, t) \right) d\mathbf{X}
 \end{aligned}$$

Representing “Conservative of Momentum” in a Weak Form

$$\begin{aligned}
 \int_{\Omega^0} Q_i(\mathbf{X}, t) R(\mathbf{X}, 0) A_i(\mathbf{X}, t) d\mathbf{X} &= \int_{\Omega^0} Q_i(\mathbf{X}, t) P_{ij,j}(\mathbf{X}, t) d\mathbf{X} \\
 &= \int_{\Omega^0} \left(\boxed{(Q_i(\mathbf{X}, t) P_{ij}(\mathbf{X}, t))_{,j}} - Q_{i,j}(\mathbf{X}, t) P_{ij}(\mathbf{X}, t) \right) d\mathbf{X} \\
 &= \boxed{\int_{\partial\Omega^0} Q_i(\mathbf{X}, t) P_{ij}(\mathbf{X}, t) N_j(\mathbf{X}, t) ds(\mathbf{X})} \\
 &\quad - \int_{\Omega^0} Q_{i,j}(\mathbf{X}, t) P_{ij}(\mathbf{X}, t) d\mathbf{X}.
 \end{aligned}$$

Divergence theorem

$$\boxed{\iiint_V (\nabla \cdot \mathbf{F}) dV = \iint_S (\mathbf{F} \cdot \hat{\mathbf{n}}) dS}$$

Representing “Conservative of Momentum” in a Weak Form

- Let $\mathbf{T}(\cdot, t)$ be the boundary force per unit reference area with $\mathbf{T}_i(\mathbf{X}, t) = \mathbf{P}_{ij}(\mathbf{X}, t)\mathbf{N}_j(\mathbf{X}, t)$

In Lagrangian configuration:

$$\int_{\Omega^0} Q_i(\mathbf{X}, t) \mathbf{R}(\mathbf{X}, 0) \mathbf{A}_i(\mathbf{X}, t) d\mathbf{X} = \int_{\partial\Omega^0} Q_i T_i ds(\mathbf{X}) - \int_{\Omega^0} Q_{i,j} P_{ij} d\mathbf{X}.$$

In Eulerian configuration:

$$\int_{\Omega^t} q_i(\mathbf{x}, t) \rho(\mathbf{x}, t) \mathbf{a}_i(\mathbf{x}, t) d\mathbf{x} = \int_{\partial\Omega^t} q_i t_i ds(\mathbf{x}) - \int_{\Omega^t} q_{i,k} \sigma_{ik} d\mathbf{x},$$

$$\int_{\Omega^0} Q_i(\mathbf{X}, t) \mathbf{R}(\mathbf{X}, 0) \mathbf{A}_i(\mathbf{X}, t) d\mathbf{X} = \int_{\partial\Omega^t} q_i t_i ds(\mathbf{x}) - \int_{\Omega^t} q_{i,k} \sigma_{ik} d\mathbf{x}$$

Representing “Conservative of Momentum” in a Weak Form

$$\int_{\Omega^t} q_\alpha \rho a_\alpha dx = \int_{\partial\Omega^t} q_\alpha t_\alpha ds - \int_{\Omega^t} \frac{\partial q_\alpha}{\partial x_\beta} \sigma_{\alpha\beta} dx$$

$$A_\alpha(X, t^n) \approx \frac{1}{\Delta t} (V_\alpha^{n+1}(X) - V_\alpha^n(X))$$

$$a(x, t) = A(\phi^{-1}(x, t), t)$$



Define

This is not Eulerian velocity!

$$v^n(x) = V(\phi^{-1}(x, t^n), t^n)$$

$$v^{n+1}(x) = V(\phi^{-1}(x, t^n), t^{n+1})$$

$$\frac{1}{\Delta t} \int_{\Omega^{t^n}} q_\alpha(x, t^n) \rho(x, t^n) (v_\alpha^{n+1}(x) - v_\alpha^n(x)) dx = \int_{\partial\Omega^{t^n}} q_\alpha t_\alpha ds - \int_{\Omega^{t^n}} \frac{\partial q_\alpha}{\partial x_\beta} \sigma_{\alpha\beta} dx$$

Still, the integrals are hard to solve.
We need to approximate the solution.

Represent a Continuous Function via Basis Functions

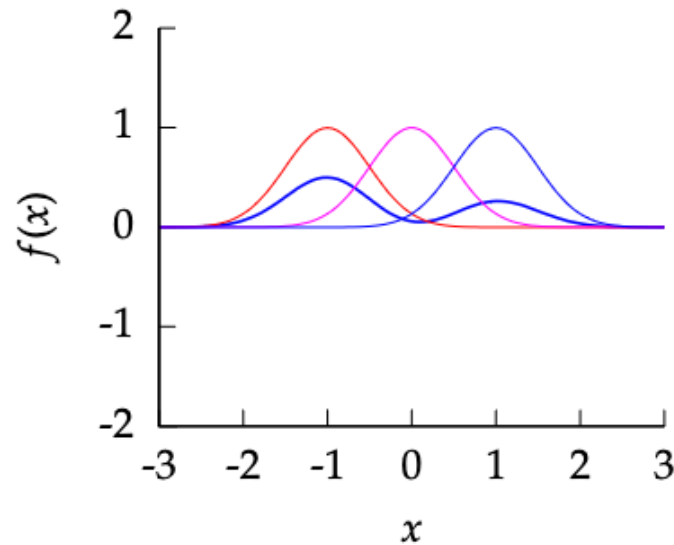
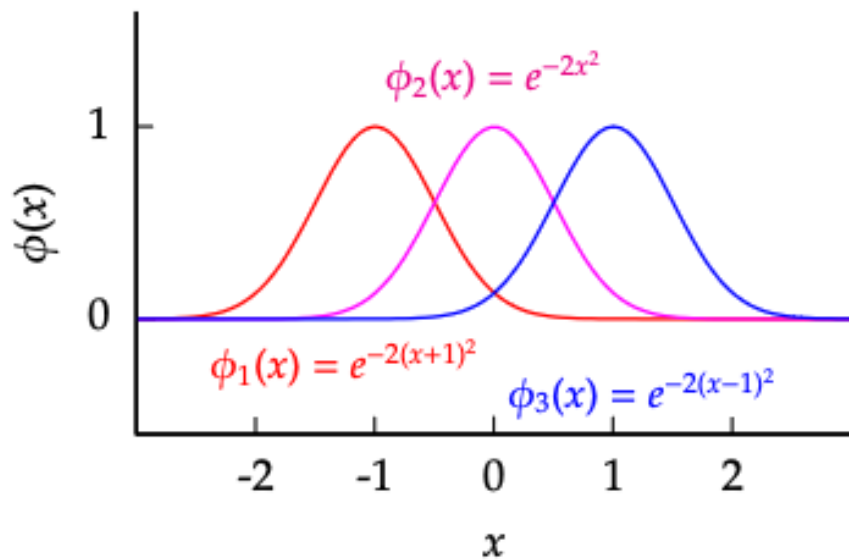


Figure: Function from radial basis with weights $w_1 = 0.50596$, $w_2 = -0.046315$, $w_3 = 0.26813$.

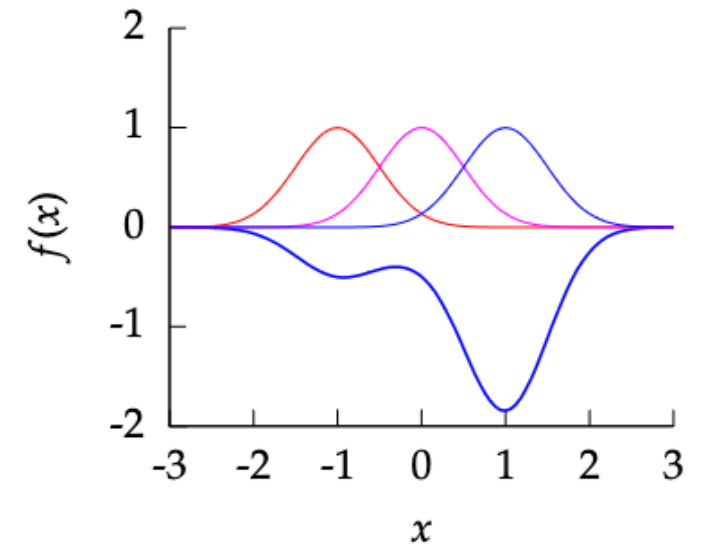
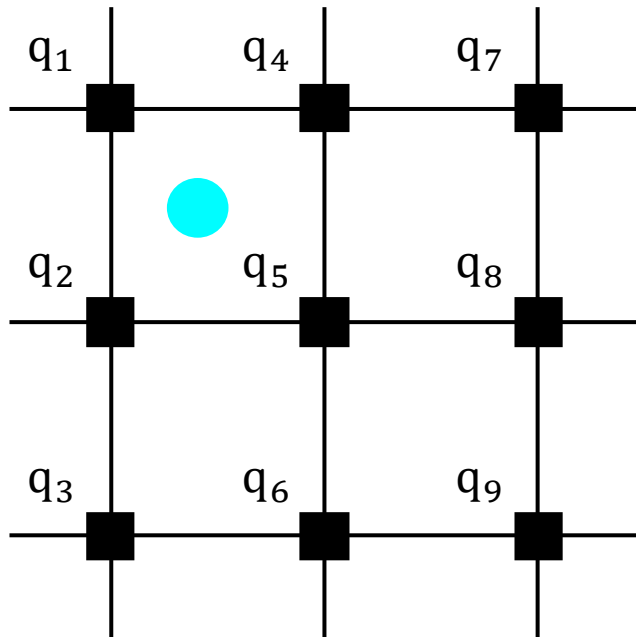


Figure: Function from radial basis with weights $w_1 = -0.47518$, $w_2 = -0.18924$, $w_3 = -1.8183$.

Represent a Continuous Function via Basis Functions



$$q(\mathbf{x}) = \sum_i q_i N_i(\mathbf{x})$$

- The choice of interpolation function $N_i(\mathbf{x})$

$$N_i(\mathbf{x}_p) = N\left(\frac{1}{h}(x_p - x_i)\right)N\left(\frac{1}{h}(y_p - y_i)\right)N\left(\frac{1}{h}(z_p - z_i)\right),$$

- The choice of kernel function $N(\mathbf{x})$

$$N(\mathbf{x}) = \begin{cases} \frac{1}{2}|\mathbf{x}|^3 - |\mathbf{x}|^2 + \frac{2}{3} & 0 \leq |\mathbf{x}| < 1 \\ \frac{1}{6}(2 - |\mathbf{x}|)^3 & 1 \leq |\mathbf{x}| < 2 \\ 0 & 2 \leq |\mathbf{x}| \end{cases} \quad N(\mathbf{x}) = \begin{cases} \frac{3}{4} - |\mathbf{x}|^2 & 0 \leq |\mathbf{x}| < \frac{1}{2} \\ \frac{1}{2}\left(\frac{3}{2} - |\mathbf{x}|\right)^2 & \frac{1}{2} \leq |\mathbf{x}| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |\mathbf{x}| \end{cases}$$

This is a low-rank approximation of the original function. No need to keep track of every function's value at each position but those on the grid nodes

Representing “Conservative of Momentum” in a Weak Form

$$\frac{1}{\Delta t} \int_{\Omega^{t^n}} q_\alpha(x, t^n) \rho(x, t^n) (v_\alpha^{n+1}(x) - v_\alpha^n(x)) dx = \int_{\partial\Omega^{t^n}} q_\alpha t_\alpha ds - \int_{\Omega^{t^n}} \frac{\partial q_\alpha}{\partial x_\beta} \sigma_{\alpha\beta} dx$$



$$q_\alpha(x) = \sum_i q_{i\alpha} N_i(x)$$

$$v_\alpha^n(x) = \sum_j v_{j\alpha}^n N_j(x)$$

$$v_\alpha^{n+1}(x) = \sum_j v_{j\alpha}^{n+1} N_j(x)$$

i, j, k denote the grid index
 α, β, γ denote the channel index

$$\frac{1}{\Delta t} \int_{\Omega^{t^n}} \left(\sum_i q_{i\alpha} N_i(x) \right) \rho(x, t^n) \left(\sum_j v_{j\alpha}^{n+1} N_j(x) - \sum_j v_{j\alpha}^n N_j(x) \right) dx$$

Reorganize the Lef-Hand Side of the Equation

$$\begin{aligned} & \frac{1}{\Delta t} \int_{\Omega^{t^n}} \left(\sum_{\mathbf{i}} q_{\mathbf{i}\alpha} N_{\mathbf{i}}(x) \right) \rho(x, t^n) \left(\sum_{\mathbf{j}} v_{\mathbf{j}\alpha}^{n+1} N_{\mathbf{j}}(x) - \sum_{\mathbf{j}} v_{\mathbf{j}\alpha}^n N_{\mathbf{j}}(x) \right) dx \\ &= \frac{1}{\Delta t} \int_{\Omega^{t^n}} \sum_{\mathbf{i}} \sum_{\mathbf{j}} \left[q_{\mathbf{i}\alpha} N_{\mathbf{i}}(x) \rho(x, t^n) (v_{\mathbf{j}\alpha}^{n+1} - v_{\mathbf{j}\alpha}^n) N_{\mathbf{j}}(x) \right] dx \\ &= \frac{1}{\Delta t} \sum_{\mathbf{i}} \sum_{\mathbf{j}} q_{\mathbf{i}\alpha} (v_{\mathbf{j}\alpha}^{n+1} - v_{\mathbf{j}\alpha}^n) \underbrace{\int_{\Omega^{t^n}} N_{\mathbf{i}}(x) \rho(x, t^n) N_{\mathbf{j}}(x) dx}_{m_{\mathbf{ij}}^n} \end{aligned}$$

Customize the Function $q(\mathbf{x}, t)$

$$\frac{1}{\Delta t} \sum_{\mathbf{i}} \sum_{\mathbf{j}} q_{\mathbf{i}\alpha} (v_{\mathbf{j}\alpha}^{n+1} - v_{\mathbf{j}\alpha}^n) \underbrace{\int_{\Omega^{t^n}} N_{\mathbf{i}}(x) \rho(x, t^n) N_{\mathbf{j}}(x) dx}_{m_{\mathbf{ij}}^n}$$

$$q_{\mathbf{i}\alpha} = \begin{cases} 1 & \text{if } \mathbf{i} = \hat{\mathbf{i}} \text{ and } \alpha = \hat{\alpha} \\ 0 & \text{otherwise} \end{cases}$$

$$= \frac{1}{\Delta t} \sum_{\mathbf{j}} (v_{\mathbf{j}\hat{\alpha}}^{n+1} - v_{\mathbf{j}\hat{\alpha}}^n) m_{\hat{\mathbf{i}}\mathbf{j}}^n$$

$$\hat{m}_{\mathbf{i}} = \sum_{\mathbf{j}} m_{\mathbf{ij}}^n \quad m_{\hat{\mathbf{i}}\mathbf{j}}^n \rightarrow m_{\hat{\mathbf{i}}} \delta_{\hat{\mathbf{i}}\mathbf{j}}$$

$$= \frac{m_{\hat{\mathbf{i}}}}{\Delta t} (v_{\hat{\mathbf{i}}\hat{\alpha}}^{n+1} - v_{\hat{\mathbf{i}}\hat{\alpha}}^n)$$

$$(mv)_{\mathbf{i}\alpha} = m_{\mathbf{i}} v_{\mathbf{i}\alpha}$$

$$= (mv)_{\hat{\mathbf{i}}\hat{\alpha}}^{n+1} - (mv)_{\hat{\mathbf{i}}\hat{\alpha}}^n$$

Representing “Conservative of Momentum” in a Weak Form

$$\frac{1}{\Delta t} \int_{\Omega^{t^n}} q_\alpha(x, t^n) \rho(x, t^n) (v_\alpha^{n+1}(x) - v_\alpha^n(x)) dx = \int_{\partial\Omega^{t^n}} q_\alpha t_\alpha ds - \int_{\Omega^{t^n}} \frac{\partial q_\alpha}{\partial x_\beta} \sigma_{\alpha\beta} dx$$

$$q_{\mathbf{i}\alpha} = \begin{cases} 1 & \text{if } \mathbf{i} = \hat{\mathbf{i}} \text{ and } \alpha = \hat{\alpha} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{m}_{\mathbf{i}} = \sum_{\mathbf{j}} m_{\mathbf{ij}}^n \quad m_{\hat{\mathbf{i}}\mathbf{j}}^n \rightarrow m_{\hat{\mathbf{i}}} \delta_{\hat{\mathbf{i}}\mathbf{j}}$$

$$(mv)_{\mathbf{i}\alpha} = m_{\mathbf{i}} v_{\mathbf{i}\alpha}$$

$$(mv)_{\mathbf{i}\alpha}^{n+1} - (mv)_{\mathbf{i}\alpha}^n = \Delta t \left[\int_{\partial\Omega^{t^n}} N_{\mathbf{i}}(x) t_\alpha(x, t^n) ds - \int_{\Omega^{t^n}} \frac{\partial N_{\mathbf{i}}}{\partial x_\beta}(x) \sigma_{\alpha\beta}(x, t^n) dx \right]$$

$\mathbf{i}, \mathbf{j}, \mathbf{k}$ denote the grid index
 α, β, γ denote the channel index

The integrals are hard to solve. We need to approximate the solution.

Approximation Method 2: Quadrature Rule

- Gauss–Legendre quadrature: given \mathbf{n} sampled points, we can approximate the continuous integrals by carefully selecting quadrature weights \mathbf{w}_i

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

- For continuum mechanics, we sample material points to approximate the integrals of each grid nodes

$$\int_{\Omega^{t^n}} \frac{\partial N_{\mathbf{i}}}{\partial x_{\beta}}(\mathbf{x}) \sigma_{\alpha\beta}(\mathbf{x}, t^n) d\mathbf{x} \approx \sum_p \frac{\partial N_{\mathbf{i}}}{\partial x_{\beta}}(\mathbf{x}_p^n) \sigma_{p,\alpha\beta}^n \boxed{V_p^n}$$

Volume for material point \mathbf{p}

Representing “Conservative of Momentum” in a Weak Form

$$(mv)_{i\alpha}^{n+1} - (mv)_{i\alpha}^n = \Delta t \left[\int_{\partial\Omega^{t^n}} N_i(x) t_\alpha(x, t^n) ds - \int_{\Omega^{t^n}} \frac{\partial N_i}{\partial x_\beta}(x) \sigma_{\alpha\beta}(x, t^n) dx \right]$$

How to derive the volume for material point \mathbf{p}

$$\int_{\Omega^{t^n}} \frac{\partial N_i}{\partial x_\beta}(x) \sigma_{\alpha\beta}(x, t^n) dx \approx \sum_p \frac{\partial N_i}{\partial x_\beta}(x_p^n) \sigma_{p,\alpha\beta}^n V_p^n$$

This is the Cauchy stress defined in the Eulerian coordinate system, while the rest terms are Lagrangian representations...

Representing “Conservative of Momentum” in a Weak Form

$$\text{Define } (\nabla_{\mathbf{w}_{ip}})_{\beta} = \frac{\partial N_i}{\partial x_{\beta}}(\mathbf{x}_p^n)$$

$$\mathbf{V}_p^n = \mathbf{J}_p^n \mathbf{V}_p^0$$

$$\int_{\Omega^{t^n}} \frac{\partial N_i}{\partial x_{\beta}}(\mathbf{x}) \sigma_{\alpha\beta}(\mathbf{x}, t^n) dx \approx \sum_p \frac{\partial N_i}{\partial x_{\beta}}(\mathbf{x}_p^n) \sigma_{p,\alpha\beta}^n \mathbf{V}_p^n$$

$$\sigma_{\alpha\beta} = \frac{1}{J_p^n} P_{\alpha\gamma} F_{\beta\gamma}$$

$$\sigma = \frac{1}{J} P F^T$$

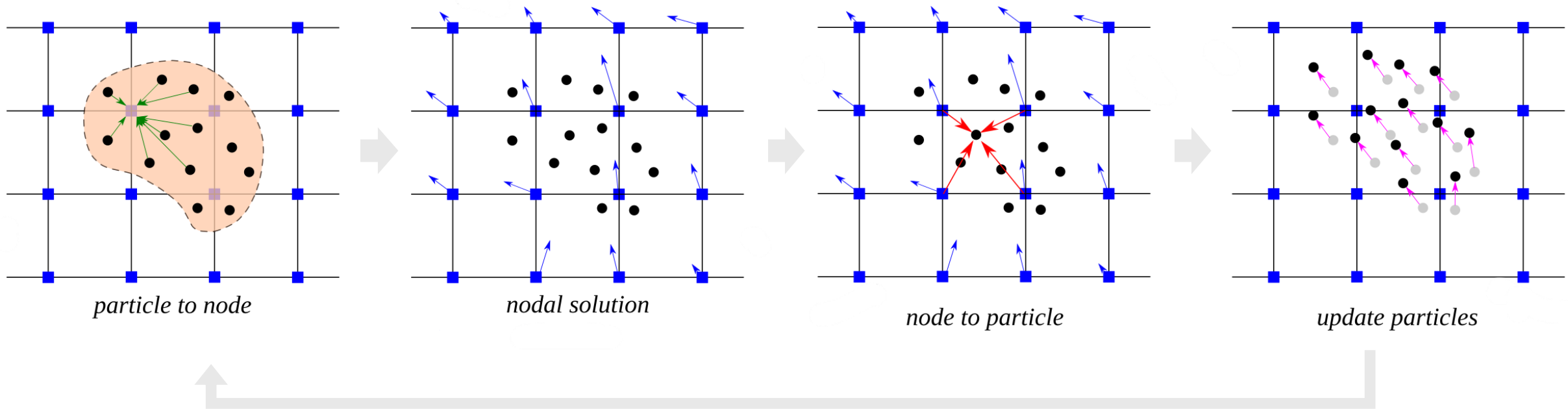
Momentum Update Rule from “Conservative of Momentum”

$$(mv)_{\mathbf{i}\alpha}^{n+1} - (mv)_{\mathbf{i}\alpha}^n = \Delta t \left[\int_{\partial\Omega^{t^n}} N_{\mathbf{i}} t_{\alpha} ds - \sum_p V_p^0 P_{p,\alpha\gamma}^n F_{p,\beta\gamma}^n (\nabla w_{\mathbf{i}p})_{\beta} \right]$$

- This update rule updates the momentum of each node on the Eulerian grid based on the deformation of material particles

Recap: A Brief Overview of MPM

1. Aggregate particle motions to grid motions
2. Aggregate particle deformation to grid forces
3. Update grid motions based on calculated stress / strain and external forces
4. Update particle motions based on grid motions
5. Repeat 1.

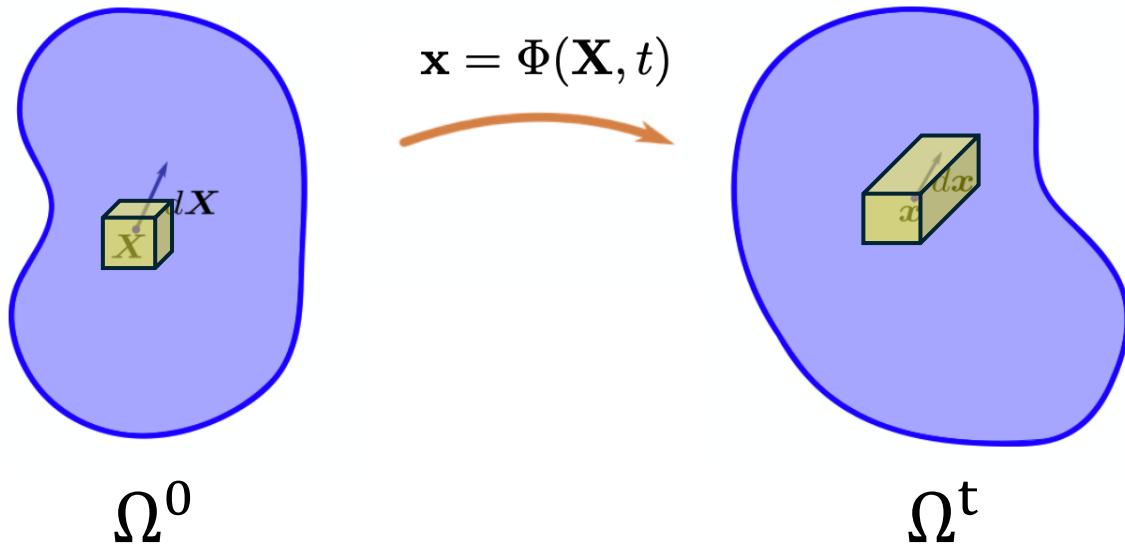


But How to Update the Deformation of Material Particles?

$$(mv)_{\mathbf{i}\alpha}^{n+1} - (mv)_{\mathbf{i}\alpha}^n = \Delta t \left[\int_{\partial\Omega^{t^n}} N_{\mathbf{i}} t_{\alpha} ds - \sum_p V_p^0 P_{p,\alpha\gamma}^n F_{p,\beta\gamma}^n (\nabla w_{\mathbf{i}p})_{\beta} \right]$$

- This update rule updates the momentum of each node on the Eulerian grid based on the deformation of material particles

Recap: Deformation Gradient



- The deformation map $\mathbf{x} = \phi(\mathbf{X}, t)$ describes the chunk of material \mathbf{X} is currently sitting at location \mathbf{x} at time t
- The deformation gradient is defined as:

$$\mathbf{F}(\mathbf{X}, t) = \frac{\partial \phi(\mathbf{X}, t)}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}}$$

Deformation gradient describes the deformation of one axis relative to other axes

Deformation Gradient Evolution

- The Lagrangian time derivative of F :

$$\frac{\partial}{\partial t} F(X, t) = \frac{\partial}{\partial t} \frac{\partial \phi(X, t)}{\partial X} = \frac{\partial}{\partial X} \frac{\partial \phi(X, t)}{\partial t} = \frac{\partial V(X, t)}{\partial X}$$

Deformation Gradient Evolution

- The Lagrangian time derivative of F :

$$\begin{aligned}\frac{\partial}{\partial t} F(X, t) &= \frac{\partial}{\partial t} \frac{\partial \phi(X, t)}{\partial X} = \frac{\partial}{\partial X} \frac{\partial \phi(X, t)}{\partial t} = \frac{\partial V(X, t)}{\partial X} \\ &= \frac{\partial V(X, t)}{\partial \phi(X, t)} \frac{\partial \phi(X, t)}{\partial X}\end{aligned}$$

Deformation Gradient Evolution

- The Lagrangian time derivative of F :

$$\begin{aligned}\frac{\partial}{\partial t} F(X, t) &= \frac{\partial}{\partial t} \frac{\partial \phi(X, t)}{\partial X} = \frac{\partial}{\partial X} \frac{\partial \phi(X, t)}{\partial t} = \frac{\partial V(X, t)}{\partial X} \\ &= \frac{\partial V(X, t)}{\partial \phi(X, t)} \frac{\partial \phi(X, t)}{\partial X} \\ &= \frac{\partial v(\phi(X, t), t)}{\partial x} F(X, t)\end{aligned}$$

Update Rule the Deformation of Material Particles

$$\frac{F_p^{n+1} - F_p^n}{\Delta t} \approx \frac{\partial v^{n+1}}{\partial x}(x_p^n) F_p^n$$



$$F_p^{n+1} = F_p^n + \Delta t \frac{\partial v^{n+1}}{\partial x}(x_p^n) F_p^n = \left(I + \Delta t \frac{\partial v^{n+1}}{\partial x}(x_p^n) \right) F_p^n$$

$$v^{n+1}(x) = \sum_{\mathbf{i}} v_{\mathbf{i}}^{n+1} N_{\mathbf{i}}(x)$$



$$\frac{\partial v_{\alpha}^{n+1}}{\partial x_{\beta}}(x) = \sum_{\mathbf{i}} v_{\mathbf{i}\alpha}^{n+1} \frac{\partial N_{\mathbf{i}}}{\partial x_{\beta}}(x)$$

$$\frac{\partial v^{n+1}}{\partial x}(x_p^n) = \sum_{\mathbf{i}} v_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p})^T$$

$$F_p^{n+1} = \left(I + \Delta t \sum_{\mathbf{i}} v_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p})^T \right) F_p^n$$

A Full Algorithm of MPM

1. **Particle to grid transfer (P2G).** Using the APIC formula from Section 10.1, the first step is to transfer particle quantities to the grid. In particular, this step computes grid mass and momentum.
2. **Compute grid velocities.** $\mathbf{v}_i = \frac{m_i \mathbf{v}_i}{m_i}$. For nodes with mass equals to 0, m_i and \mathbf{v}_i are manually reset to 0. One would traditionally think this requires a floating point threshold when implementing this step. However in practice directly comparing against 0.0f will not produce any unstable behavior or overflow. This is because even though you may get large forces on small mass nodes, the effect is scaled by the weight again when transferrring back to particles. In fact, hard coding a threshold will cause various problems such as drifting of total momentum.
3. **Identify grid degree of freedoms.** This step is important for implementation efficiency. We label the grid nodes with nonzero masses to be the actually degree of freedoms. All other nodes will remain static and are not considered being part of the solver unknowns.
4. **Compute explicit grid forces \mathbf{f}_i^n** using Equation 189.

A Full Algorithm of MPM

5. **Grid velocity update** using Equation 183. This step should take the boundary conditions or collision objects into account. In the case of explicit integration, each nodal velocity can be independently set to the desired value due to Dirichlet boundary conditions or rigid object collisions. See Section 12.1 for more details on how to deal with collision objects.
6. **Update particle deformation gradient** using Equation 181. Note that we never actually moved the grid or computed any new grid \mathbf{x} . The motion is imaginary and only velocities are explicitly stored.
7. **Grid to particle transfer (G2P)**. This step computes new particle velocities \mathbf{v}_p^{n+1} and affine matrices \mathbf{B}_p^{n+1} with the scheme given in Section 10.1.
8. **Particle advection**. Finally particles are advected with their new velocities: $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}$. Note that this is only true when APIC is used. In a FLIP or FLIP-PIC-blending transfer scheme, $\mathbf{x}_p^{n+1} = \sum_i \mathbf{x}_i^{n+1} w_{ip}$ should be used instead (note that these two formulas are equivalent in the PIC or APIC case).

Check “The Material Point Method for Simulating Continuum Materials” by Chenfanfu Jiang et al for more details

PhysGaussian: Physics-Integrated 3D Gaussians for Generative Dynamics

Tianyi Xie^{1*} Zeshun Zong^{1*} Yuxing Qiu^{1*} Xuan Li^{1*}

Yutao Feng^{2,3} Yin Yang³ Chenfanfu Jiang¹

¹ UCLA, ² Zhejiang University, ³ University of Utah



PhysGaussian: A Unified Simulation-Rendering Pipeline

- Previous approaches to physics-based visual content generation require multiple stages:
 1. Construct the geometry
 2. Transform to simulation-ready representations (e.g. Tetrahedral meshes)
 3. Simulate with physics
 4. Render
- Idea: “what you see is what you simulate”:
 - Use 3DGS for rendering and simulation

How to Simulate with 3DGS based on MPM?

- A quick recap of basic principles in MPM:
 - Deformation map $x = \phi(X, t)$
 - Deformation gradient $F(X, t) = \nabla_X \phi(X, t)$
 - Conservation of mass: $\int_{B_\epsilon^t} \rho(x, t) \equiv \int_{B_\epsilon^0} \rho(\phi^{-1}(X, t), 0)$
 - Conservation of momentum: $\rho(x, t) \dot{v}(x, t) = \nabla \cdot \sigma(x, t) + f^{ext}$
- A quick recap of update rules in MPM. For Eulerian grid i and Lagrangian particles p at time n :
 - Velocity update: $\frac{m_i}{\Delta t} (v_i^{n+1} - v_i^n) = -\sum_p V_p^0 \frac{\partial \Psi}{\partial F} (F_p^{E,n}) F_p^{E,n \top} \nabla \omega_{i,p}^n + f_i^{ext}$
 - Particle's position update: $x_p^{n+1} = x_p^n + \Delta t v_p^{n+1}$
 - Deformation gradient update: $F_p^{E,n+1} = (I + \Delta t \sum_i v_i^{n+1} \nabla \omega_{i,p}^n) F_p^{E,n}$

How to Simulate with 3DGS based on MPM?

- How to define particles?
 - 3DGS are particles, which deform like continuum materials
- Original 3DGS:

$$G_p(\mathbf{X}) = e^{-\frac{1}{2}(\mathbf{X} - \mathbf{X}_p)^T \mathbf{A}_p^{-1}(\mathbf{X} - \mathbf{X}_p)}$$

- Deformed 3DGS with the deformation map $\phi(X, t)$:

$$G_p(\mathbf{x}, t) = e^{-\frac{1}{2}(\underbrace{\phi^{-1}(\mathbf{x}, t) - \mathbf{X}_p}_{\text{deformed position}})^T \mathbf{A}_p^{-1}(\phi^{-1}(\mathbf{x}, t) - \mathbf{X}_p)}$$

A deformed Gaussian is not necessarily Gaussian

An Assumption of Local Affine Transformations

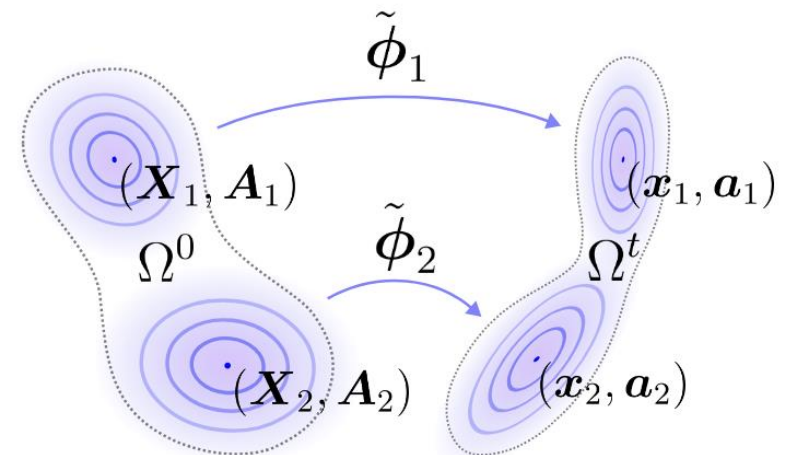
- The deformations are local affine transformations:

$$\tilde{\phi}_p(\mathbf{X}, t) = \mathbf{x}_p + \mathbf{F}_p(\mathbf{X} - \mathbf{X}_p)$$

We know affine transformed Gaussians are still Gaussians (check lecture 3)

- Deformed 3DGS with the deformation map $\phi(\mathbf{X}, t)$:

$$G_p(\mathbf{x}, t) = e^{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_p)^T (\mathbf{F}_p \mathbf{A}_p \mathbf{F}_p^T)^{-1} (\mathbf{x} - \mathbf{x}_p)}$$



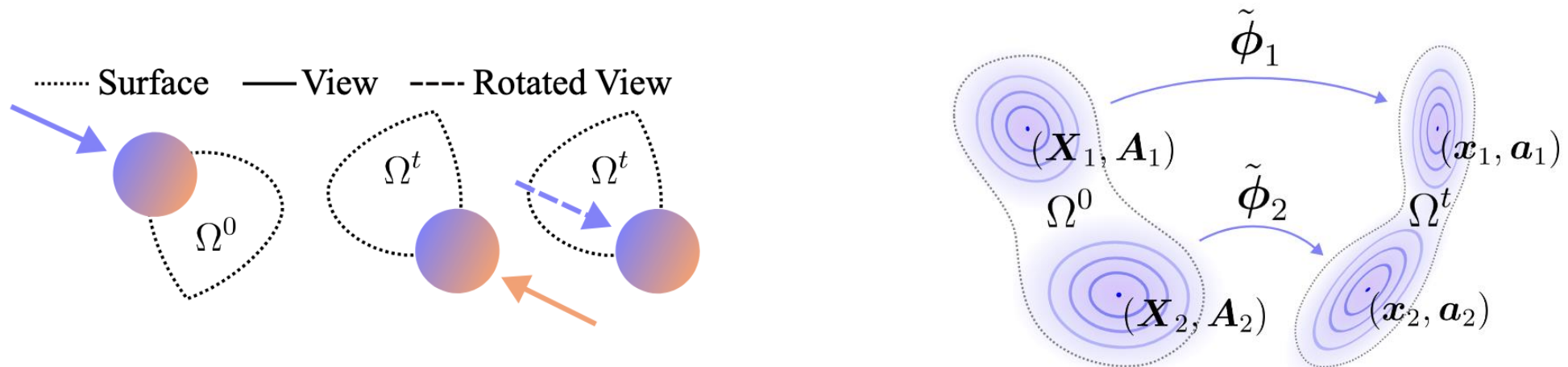
An Assumption of Local Affine Transformations

- The deformations are local affine transformations:

$$\tilde{\phi}_p(\mathbf{X}, t) = \mathbf{x}_p + \mathbf{F}_p(\mathbf{X} - \mathbf{X}_p)$$

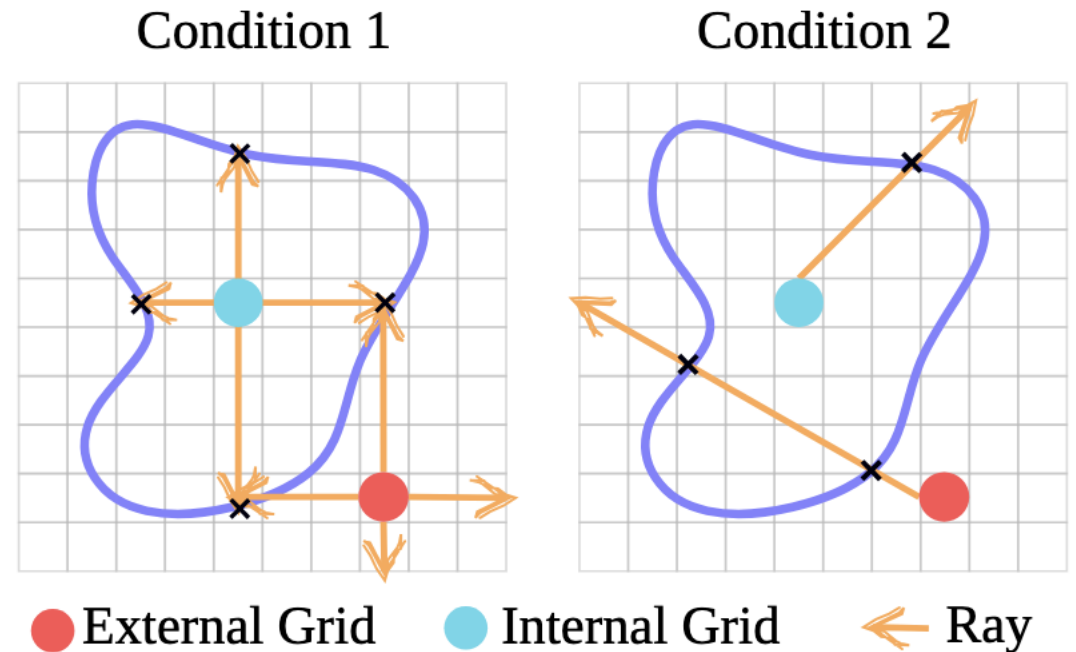
- Deformed 3DGS with the deformation map $\phi(X, t)$:

$$G_p(\mathbf{x}, t) = e^{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_p)^T (\mathbf{F}_p \mathbf{A}_p \mathbf{F}_p^T)^{-1} (\mathbf{x} - \mathbf{x}_p)}$$



How to Simulate with 3DGS based on MPM?

- How to define grids?
- When shooting a ray, we know if it intersects with the surface from the change of opacity
- Two patterns for defining internal and external grids



Results



What We Will Cover the Next Week

- Physical modeling
 - Learning-based physical modeling