SIGGRAPH ASIA 2020 VIRTUAL

Pre-recorded sessions:
From 4 December 2020

Live sessions:
10 – 13 December 2020

SA2020.SIGGRAPH.ORG
#SIGGRAPHAsia | #SIGGRAPHAsia2020

A Moving Least Square Reproducing Kernel Particle Method for Unified Multiphase Continuum Simulation

Xiao-Song Chen[1] Chen-Feng Li[2] Geng-Chen Cao[1] Yun-Tao Jiang[1] Shi-Min Hu[1]

[1]Tsinghua University, China
[2]Swansea University, UK

Prifysgol Abertawe
Swansea University

Background

Multiphase, Multi-material Phenomenon in Daily Life

[Spaghetti](), Wikipedia.

[Dunking (biscuit)](), Wikipedia.

The Challenge

1. Coupling of different solvers

Reuse existing solvers
Hard to extend
Hard to allow varying materials

Deformable body

Rigid body … Cloth

Fluid Hair

2. Unified particle-based simulation

Monolithic data structure
Easier to extend for more materials
Allow transition among different materials

Unified particle representation

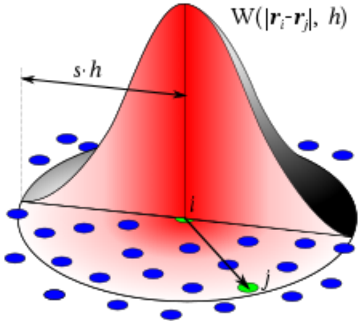
Particle-based Simulation

- Pure particle-based methods

e.g. SPH, PBF

Only Particles

Correction for consistency



SPH, Wikipedia.

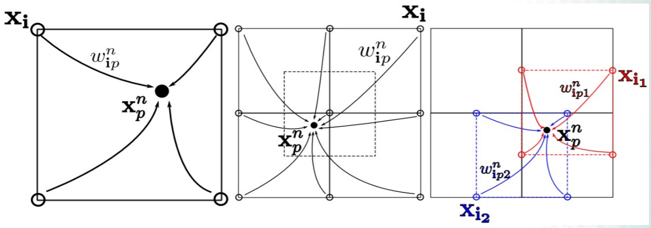
- Hybrid methods

e.g. FLIP, MPM

Particles + Background Grid

Transfer data between particles and grid

Consistency from grid regularity



Poly-PIC [Fu et al. 2017]

SPH Improvements

Standard SPH interpolation is inconsistency, which leads to incorrect behavior in simulation.

- Gradient Correction

Linear Consistent Gradient Correction [Bonet and Lok 1999]:

$$\nabla\widetilde{W}_i(\mathbf{x}_{ij}) = \mathbf{L}_i \nabla W(\mathbf{x}_{ij})$$

where $\mathbf{L}_\mathrm{i} = \left(\Sigma_j \nabla W(\mathbf{x}_{ij}) \otimes \mathbf{x}_{ji} V_j\right)^{-1}$

Used by [Peer et al. 2018; Gissler et al. 2020]

- Kernel Correction

Reproducing Kernel Particle Method (Moving Least Square Reproducing Kernel Method)

$$N_i(\mathbf{x}) = \mathbf{h}(0)^T \mathbf{M}^{-1}(\mathbf{x})\mathbf{h}(\mathbf{x}_i - \mathbf{x})\Phi(\mathbf{x}_i - \mathbf{x})V_i$$

where $\mathbf{M}(\mathbf{x}) = \Sigma_i \mathbf{h}(\mathbf{x}_i - \mathbf{x})\mathbf{h}(\mathbf{x}_i - \mathbf{x})^T \Phi(\mathbf{x}_i - \mathbf{x})V_i$

MLS used by [Müller et al. 2004; Gerszewski et al. 2009]

Our Method

We propose a MLSRK framework for unified simulation of multiphase material and phase-change phenomenon.

Our technical contributions include:

- A pure particle-based simulation framework, which uniformly handles different materials.
- Multiphase phenomenon is captured by integrating phase-field model in our MLSRK framework.

Related Works

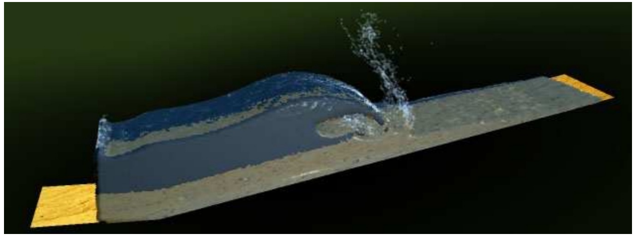
SIGGRAPH ASIA 2020 VIRTUAL

SPH Fluids

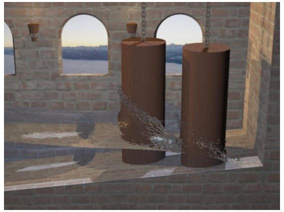
- Originally used for fluid simulation (hydrodynamics)

Incompressibility

[Müller et al. 2003]

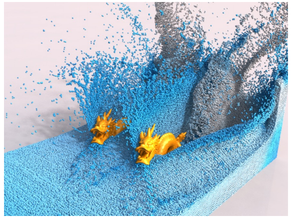


WCSPH
[Becker and Teschner 2007]



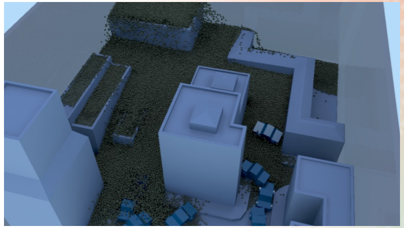
PCISPH
[Solenthaler and Pajarola 2009]

IISPH
[Ihmsen et al. 2013]

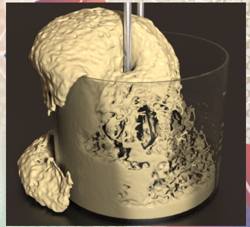


DFSPH
[Bender and Koschier 2015]

Other types of flow



[Alduán and Otaduy 2011]

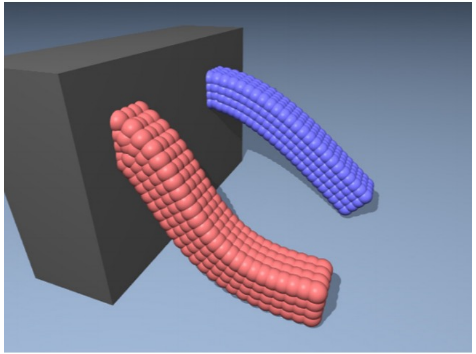


[Peer et al. 2015]

[Weiler et al. 2018]

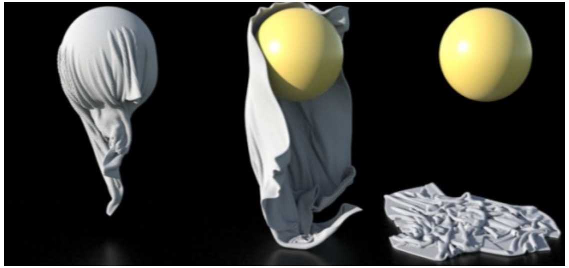
SPH Solids

- Later used for solid simulation

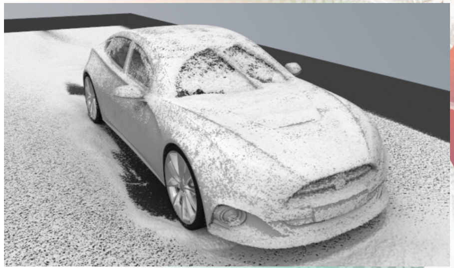


Corotated SPH
[Becker et al. 2009]

Gradient Correction

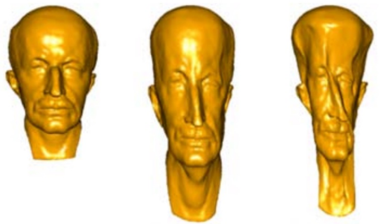


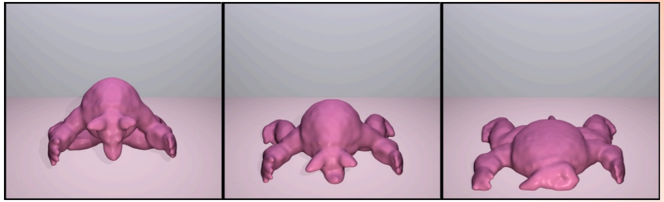
[Peer et al. 2018]



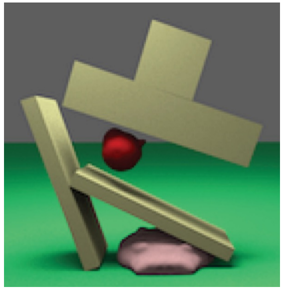
[Gissler et al. 2020]

MLS solids


[Müller et al. 2004]

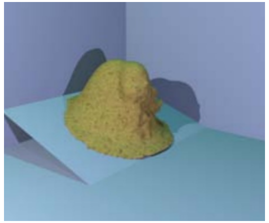

[Gerszewski et al. 2005]


[Pauly et al. 2005]

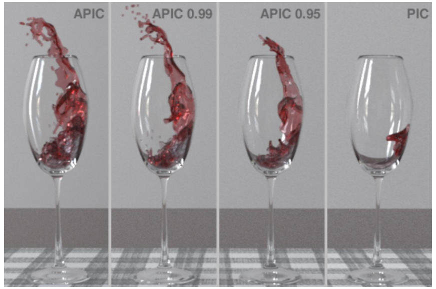

[Jones et el. 2014]

Hybrid Methods

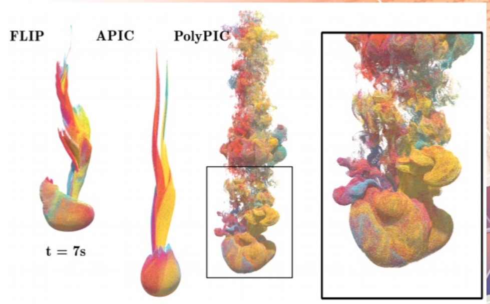
- PIC/FLIP, MPM



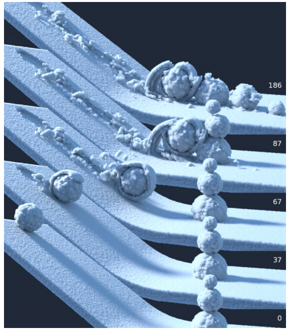
[Zhu and Bridson 2005]



APIC
[Jiang et al. 2015]



Poly-PIC
[Fu et al. 2015]

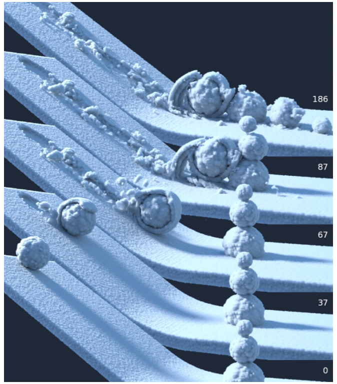


[Stomakhin et al. 2013]

MLS-MPM
[Hu et al. 2018]

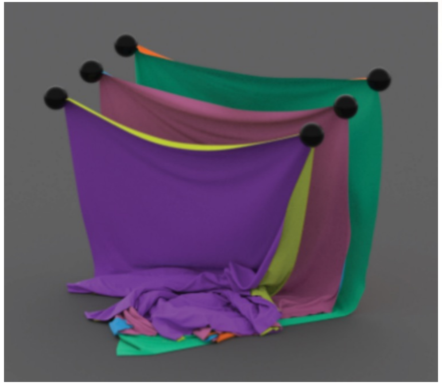
Hybrid Methods

- MPM for different materials

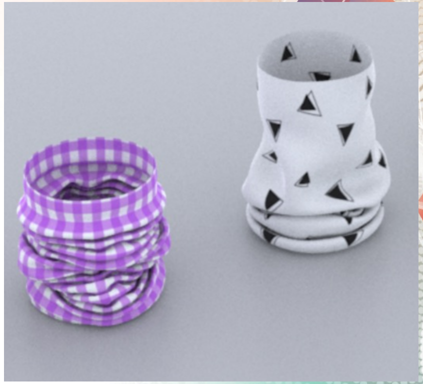


[Stomakhin et al. 2013]

[Klár et al. 2016]

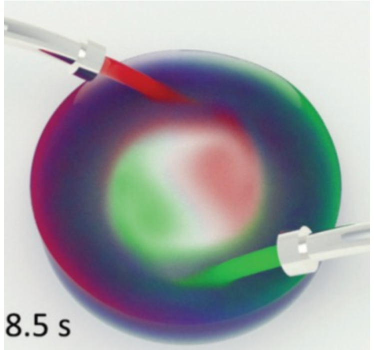


[Jiang et al. 2017]



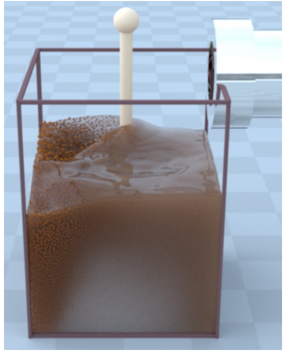
[Guo et al. 2017]

Multiphase Simulation

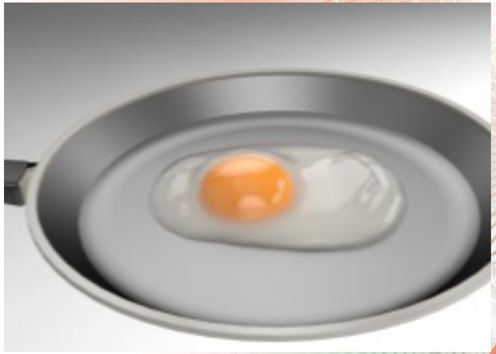


[Ren et al. 2014]

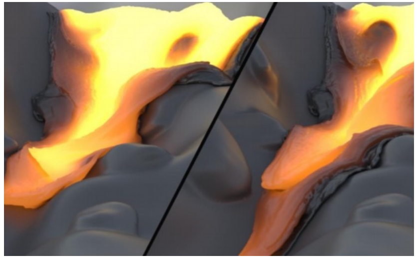
[Yang et al. 2015]



[Yan et al. 2016]



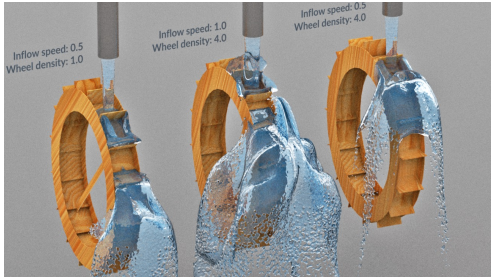
[Yang et al. 2017]



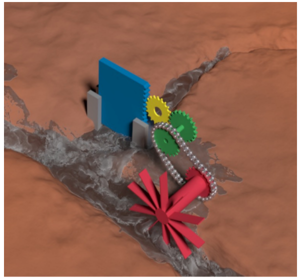
[Stomakhin et al. 2014]

[Ding et al. 2019]

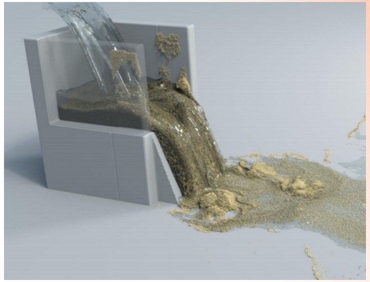
Coupling of multiple materials



[Hu et al. 2018]



[Gissler et al. 2018]



[Tampubolon et al. 2018]

[Guo et al. 2018]

[Fei et al. 2017]

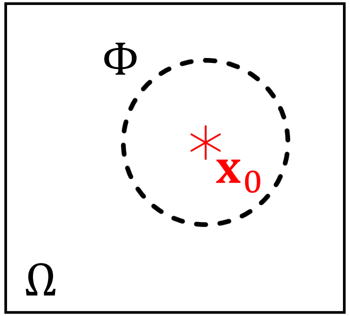
[Fei et al. 2018]

[Fei et al. 2019]

MLSRK introduction

MLSRK Interpolation in Continuous Space

MLSRK



Least Square interpolation:

In the neighborhood of point $\mathbf{x}_0$, approximate function u with a linear combination of basis functions, e.g.

$$\tilde{u} = c_1 + c_x x + c_y y.$$

Or in matrix form:

$$\tilde{u} = \mathbf{c}^T \mathbf{h} = \mathbf{h}^T \mathbf{c}$$

Minimize weighted error

$$E[\mathbf{c}] = \int_\Omega (\tilde{u} - u)^2 \Phi(\mathbf{x} - \mathbf{x}_0) \mathrm{d}\mathbf{x}$$

to get optimal basis coefficient $\mathbf{c} = \mathbf{M}^{-1}\mathbf{r}$, where

$$\boxed{\mathbf{M} = \int_\Omega \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x})^T \Phi(\mathbf{x} - \mathbf{x}_0) \mathrm{d}\mathbf{x},}$$

Moment matrix

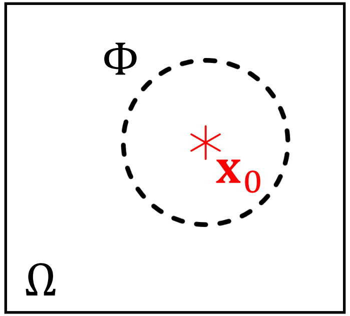
$$\mathbf{r} = \int_\Omega \mathbf{h}(\mathbf{x}) \Phi(\mathbf{x} - \mathbf{x}_0) u(\mathbf{x}) \mathrm{d}\mathbf{x}.$$

Estimate $\tilde{u}$ at point $\mathbf{x}_0$, we get the interpolated value

$$u^h = \tilde{u}(\mathbf{x}_0) = \mathbf{h}(\mathbf{x}_0)^T \mathbf{c} = \int_\Omega \mathbf{h}(\mathbf{x}_0)^T \mathbf{M}^{-1} \mathbf{h}(\mathbf{x}) \Phi(\mathbf{x} - \mathbf{x}_0) u(\mathbf{x}) \mathrm{d}\mathbf{x}$$

MLSRK Interpolation in Continuous Space

MLSRK



With fixed $\mathbf{x}_0$, we already have:

$$u^h = \tilde{u}(\mathbf{x}_0) = \mathbf{h}(\mathbf{x}_0)^T \mathbf{c} = \int_\Omega \mathbf{h}(\mathbf{x}_0)^T \mathbf{M}^{-1} \mathbf{h}(\mathbf{x}) \Phi(\mathbf{x} - \mathbf{x}_0) u(\mathbf{x}) \mathrm{d}\mathbf{x}$$

By moving point $\mathbf{x}_0$, we can interpolate function:

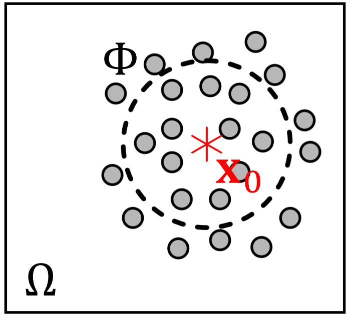
$$u^h(\mathbf{x}) = \int_\Omega \underbrace{\boxed{\mathbf{h}(\mathbf{x})^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{h}(\mathbf{x}')}}_{C(\mathbf{x},\mathbf{x}')} \Phi(\mathbf{x}' - \mathbf{x}) u(\mathbf{x}') \mathrm{d}\mathbf{x}'$$

Standard SPH Interpolation:

$$u^h(\mathbf{x}) = \int_\Omega \Phi(\mathbf{x}' - \mathbf{x}) u(\mathbf{x}') \mathrm{d}\mathbf{x}'$$

MLSRK Interpolation with Particles

MLSRK



In continuous space:

$$u^h(\mathbf{x}) = \int_\Omega \mathbf{h}(\mathbf{x})^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{h}(\mathbf{x}') \Phi(\mathbf{x}' - \mathbf{x}) u(\mathbf{x}') \mathrm{d}\mathbf{x}'$$

Sampled by particles:

$$u^h(\mathbf{x}) = \sum_{\boxed{i}} \mathbf{h}(\mathbf{x})^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{h}(\mathbf{x}_i) \boxed{\Phi(\mathbf{x}_i - \mathbf{x})} u(\mathbf{x}_\mathbf{i}) V_i$$

where $\mathbf{M}(\mathbf{x}) = \sum_{\boxed{i}} \mathbf{h}(\mathbf{x}_i) \mathbf{h}(\mathbf{x}_i)^T \boxed{\Phi(\mathbf{x}_i - \mathbf{x})} V_i$.

Sum for i in neighborhood of $\mathbf{x}$, where $\Phi \neq 0$.

Write in form of shape function:

$$u^h(\mathbf{x}) = \sum_i \boxed{N_i(\mathbf{x})} u_i$$ Shape function

where $N_i(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{h}(\mathbf{x}_i) \Phi(\mathbf{x}_i - \mathbf{x}) V_i$,

$\mathbf{b}(\mathbf{x})^T = \mathbf{h}(\mathbf{x})^T \mathbf{M}^{-1}(\mathbf{x})$.

MLSRK Interpolation with Particles

Shifted and scaled interpolation:

$$u^h(\mathbf{x}) = \sum_i N_i(\mathbf{x}) u_i$$

where $N_i(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{h}\left(\frac{\mathbf{x}_i - \mathbf{x}}{a}\right) \Phi\left(\frac{\mathbf{x}_i - \mathbf{x}}{a}\right) V_i,$

$$\mathbf{b}(\mathbf{x})^T = \mathbf{h}(\mathbf{0})^T \mathbf{M}^{-1}(\mathbf{x}),$$

$$\mathbf{M}(\mathbf{x}) = \sum_i \mathbf{h}\left(\frac{\mathbf{x}_i - \mathbf{x}}{a}\right) \mathbf{h}\left(\frac{\mathbf{x}_i - \mathbf{x}}{a}\right)^T \Phi\left(\frac{\mathbf{x}_i - \mathbf{x}}{a}\right) V_i\,.$$

Cartesian cubic spline kernel can be used:

$$\Phi(x, y) = \mathrm{cublicspline}(x)\ \mathrm{cublicspline}(y)$$

$$\Phi(x, y, z) = \mathrm{cublicspline}(x)\ \mathrm{cublicspline}(y)\ \mathrm{cublicspline}(z)$$

MLSRK Consistency

MLSRK interpolation reproduces basis function, because error is minimized to 0.

Consistency depends on basis selection. Linear consistency is usually enough.

Linear basis: $\mathbf{h}(\mathbf{x}) = [1\ \mathbf{x}]$

i.e.

$$\mathbf{h}([x]) = [1\ x]$$
$$\mathbf{h}([x\ y]) = [1\ x\ y]$$
$$\mathbf{h}([x\ y\ z]) = [1\ x\ y\ z]$$

Constant Consistency :

$$\sum_i N_i(\mathbf{x}) = 1, \sum_i \nabla N_i(\mathbf{x}) = \mathbf{0}$$

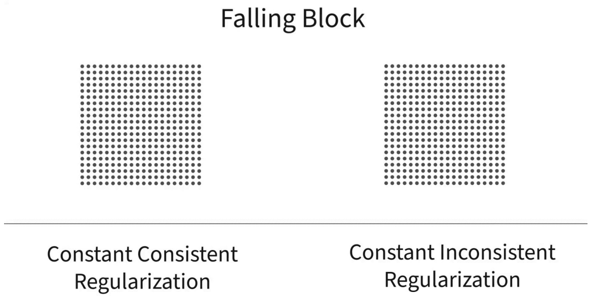
Linear Consistency :

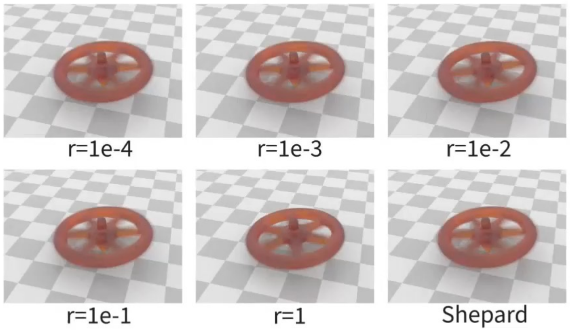
$$\sum_i N_i(\mathbf{x})(\mathbf{x}_i - \mathbf{x}) = \mathbf{0}, \sum_i \nabla N_i(\mathbf{x}) \otimes (\mathbf{x}_i - \mathbf{x}) = \mathbf{I}$$

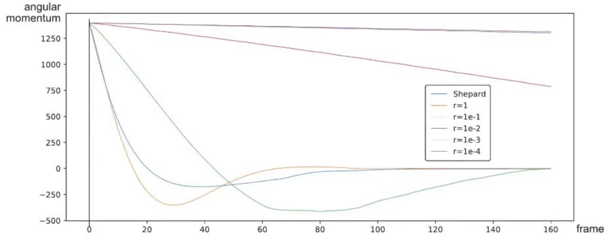
Regularization for Ill Particle Distribution

Moment matrix $\mathbf{M}(\mathbf{x}) = \sum_i \mathbf{h}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_i)^T \Phi(\mathbf{x}_i - \mathbf{x}) V_i$ can be singular for ill particle distribution (colinear/coplanar)

Regularization for linear coefficient $\mathbf{M}_r = a^d \begin{bmatrix} 0 & & & \\ & r & & \\ & & \ddots & \\ & & & r \end{bmatrix}$







The relation of SPH, MPM and MLSRK

All can handle complex continuum models.

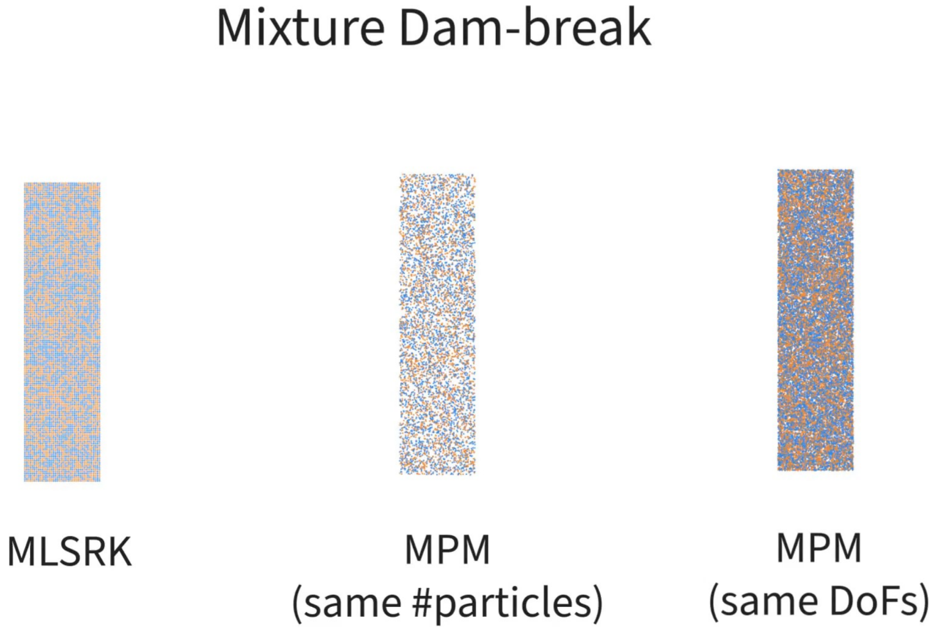
MLSRK is an improved SPH scheme

- Similar data structure & algorithm workflow
- Invert a small matrix per particle
- MLSRK uniformly discretizes PDEs, suitable for multiphase systems

Comparison between MLS-MPM and MLSRK

MLSRK and MLS-MPM have similar theoretical foundation, but with different discretization

- MLS-MPM tracks DoFs on grid, MLSRK on particles
- Grid introduces extra complexity



MLSRK Discretization for Multiphase System

MLSRK discretization follows FEM:

1. Strong Form

   Differential Equation

2. Weak Form

   Integrational Equation

3. Galerkin Discretization

   Shape function as Trial function

Multiphase System:

- Momentum Conservation
- Phase Evolution

Momentum Conservation

- Cauchy momentum equation:

acceleration

Cauchy stress

$$\boxed{\rho}\boxed{\frac{D\mathbf{u}}{Dt}} = \boldsymbol{\nabla}\cdot\boxed{\boldsymbol{\sigma}} + \boxed{\rho}\boxed{\mathbf{g}}$$

gravity

density

Weak form:

$$\int_{\Omega}\left[\rho\left(\frac{Du}{Dt} - g\right)v + \nabla v\cdot\sigma\right]\mathrm{d}V - \oint_{\partial\Omega} n\cdot\sigma v\mathrm{d}S = 0$$

Galerkin discretization ($v = N_i$) & nodal integration (integration to summation):

$$\sum_j\left(\sum_k N_i(\mathbf{x}_k)N_j(\mathbf{x}_k)\rho_k V_k\right)(\dot{\mathbf{u}}_j - \mathbf{g}) = -\sum_j \nabla N_i(\mathbf{x}_j)\cdot\boldsymbol{\sigma}_j V_j$$

Mass lumping:

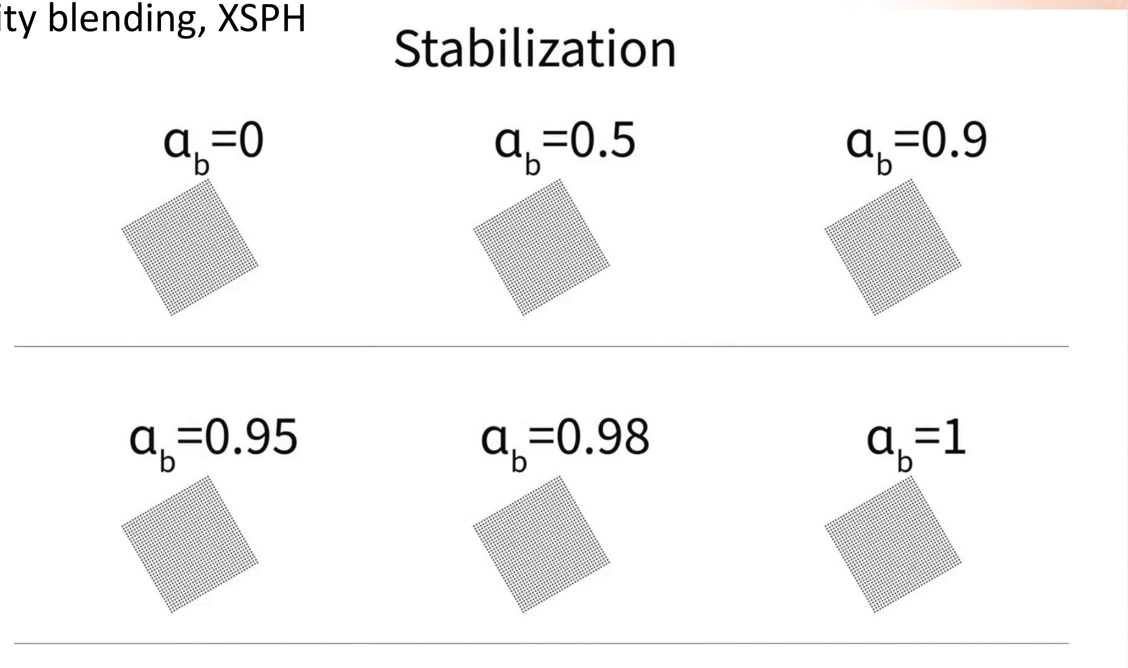
$$\dot{\mathbf{u}}_i = \frac{-\Sigma_j\boldsymbol{\nabla}N_i(\mathbf{x}_j)\cdot\boldsymbol{\sigma}_j V_j}{\Sigma_j N_i(\mathbf{x}_j)\rho_j V_j} + \mathbf{g}$$

Momentum Conservation

- Stabilization:

$$\mathbf{u}_i = \alpha_b \mathbf{u}_i + (1 - \alpha_b)\mathbf{u}^h(\mathbf{x}_i)$$

PIC/FLIP velocity blending, XSPH



Phase Evolution

order parameter chemical potential

- Cahn-Hillard Equation

$$\frac{\partial \boxed{\eta}}{\partial t} = \mathbf{\nabla} \cdot (\boxed{L}\mathbf{\nabla}\boxed{\mu_c})$$

degenerate mobility

Weak form:

$$\int_{\Omega} \left(\frac{\partial \eta}{\partial t} v + L \nabla \mu_c \cdot \nabla v \right) \mathrm{d}V - \oint_{\partial \Omega} v L \nabla \mu_c \cdot \mathbf{n} \mathrm{d}S = 0$$

Discretization:

$$\left(\frac{\partial \eta}{\partial t} \right)_i = - \frac{\Sigma_j \mathbf{\nabla} N_i(\mathbf{x_j}) \cdot \nabla \boldsymbol{\mu_c}(\mathrm{x_j}) L_j V_j}{\Sigma_j N_i(\mathbf{x_j}) V_j}$$

Phase Evolution

- Mixture Model

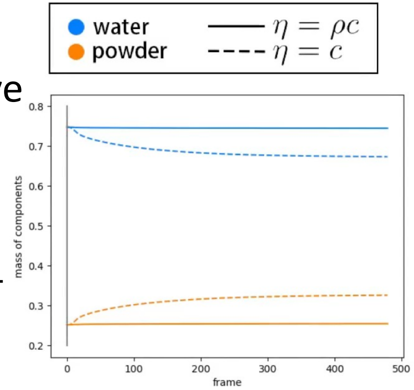
Each particle carries mass concentrations c of different components

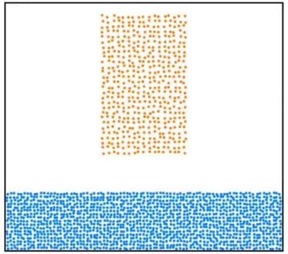
$\eta = c$ [Yang et al. 2015] do not conserve mass

We use $\eta = \rho c$:

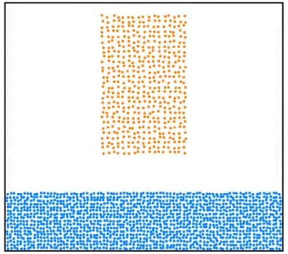
$$\dot{c}_{\boldsymbol{i}} = -\frac{1}{\rho_i}\frac{\Sigma_j \boldsymbol{\nabla} N_i(\mathbf{x}_j)\cdot\nabla\mu_c(\mathbf{x}_j)\tilde{L}_j\rho_j V_j}{\Sigma_j N_i(\mathbf{x}_j)V_j}$$

Mass Conservation





η=ρc



η=c

Video

Unified Multiphase Constitutive Model

Deformation Gradient Estimation

Constitutive models described with **deformation gradient $\mathbf{F}$**

Deformation gradient evolution:

$$\dot{\mathbf{F}} = (\nabla \mathbf{u})^T \mathbf{F},$$

where $(\nabla \mathbf{u})_i = \Sigma_j \nabla N_j(\mathbf{x}_{\boldsymbol{i}}) \otimes \mathbf{u}_j$.

Time step advance:

$$\mathbf{F}^{\boldsymbol{n}+\mathbf{1}} = \left(I + (\nabla \mathbf{u}^{\boldsymbol{n}+\mathbf{1}})^T \Delta \mathrm{t}\right)\mathbf{F}^{\mathbf{n}}$$

Deformation Gradient Estimation

Accumulated error can cause volume changing artifact for fluid.

- Volume correction for fluids

SPH volume estimation [Solenthaler and Pajarola 2008]

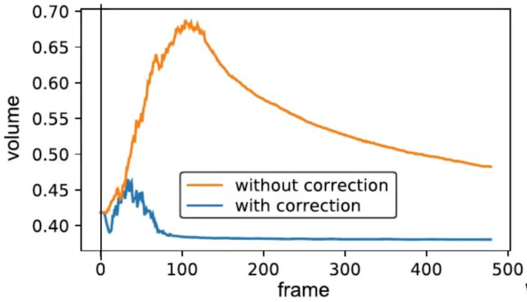
$$V_j = \frac{1}{\sum_j W_{ij}}.$$

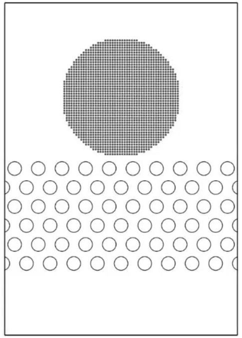
Corrected deformation gradient is

$$\tilde{F} = \left(\frac{V_i}{V_i^0 J}\right)^{1/d} F,$$

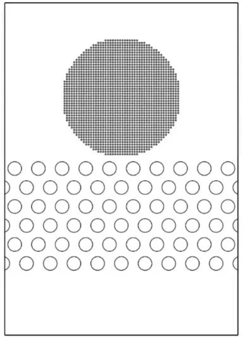
where $J = \det(F)$, so that $\tilde{J} V_i^0 = V_i$.

Galton Board





Without volume correction



With volume correction

Constitutive Models of Multiphase Continua

- Hyperelasticity

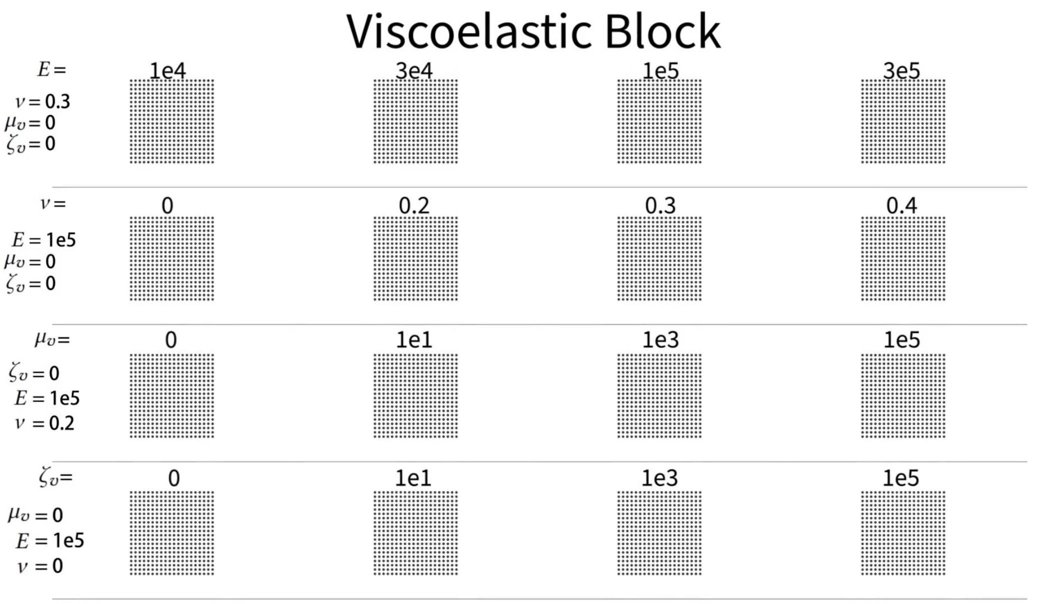
Neo-Hookean material:

$$\boldsymbol{\sigma}_e = \frac{1}{J}\left(\mu_e\left(\mathbf{F}\mathbf{F}^T - \mathbf{I}\right) + \lambda_e \log(J)\,\mathbf{I}\right)$$

- Viscosity

Newtonian viscosity:

$$\boldsymbol{\sigma}_v = \mu_v\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T - \frac{2}{d}(\nabla\cdot\mathbf{u})\mathbf{I}\right] + \zeta_v(\nabla\cdot\mathbf{u})\mathbf{I}$$



Constitutive Models of Multiphase Continua

- Elastoplasticity

Elastoplastic multiplicative decomposition $\mathbf{F} = \mathbf{F}_e \mathbf{F}_p$

Return mapping to determine $\mathbf{F}_e$

1. Critical singular value [Stomakhin et al. 2013]
2. Drucker-Prager condition [Klár et al. 2016; Tampubolon et al. 2017]

Rolling Snowball

Shooting Armadillo

Constitutive Models of Multiphase Continua

Material Strength change

- Use concentration to decide yield condition parameters.

Biscuit

Constitutive Models of Multiphase Continua

- Volume Change in Phase Evolution

Total volume can change when two materials mix up.

Multiplicative decomposition:

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_p \mathbf{F}_r$$

where $\mathbf{F}_r = (\rho^0_{\text{rest}} / \rho_{\text{rest}})^{1/d} \mathbf{I}$

Starch Powder Dissolution

Algorithm Workflow

In a time step:

1. Cache correction coefficient
2. Calculate stress from deformation gradient and velocity gradient
3. Calculate acceleration from stress and update velocity
4. Calculate chemical potential and update concentration
5. Update deformation gradient and position.

A simple implementation using less than 200 lines of C++ code is provided.

ALGORITHM 1: Multiphase MLSRK Workflow

repeat

  for *each particle* **do**
    find neighborhood particles inside kernel support range;
    calculate MLSRK correction coefficient $\mathbf{b}$ and its gradient for the particle neighborhood from Eqn. (8);
  end
  for *each particle* **do**
    calculate stress tensor σ from deformation gradient $\mathbf{F}$ and velocity gradient $\nabla\mathbf{u}$ with Eqns. (22) and (23);
  end
  for *each particle* **do**
    calculate acceleration from stress σ using Eqn. (11), update velocity $\mathbf{u}_{n+1} = \mathbf{u}_n + \mathbf{a}_n \Delta t$;
    apply boundary condition to velocity;
  end
  for *each particle* **do** `// optional for phase evolution`
    calculate chemical potential μ_c from concentration with Eqn. (16);
  end
  for *each particle* **do** `// optional for phase evolution`
    update mass concentration c according to chemical potential with Eqn. (15);
  end
  for *each particle* **do**
    calculate velocity gradient $\nabla\mathbf{u}$ from Eqn. (18), update deformation gradient $\mathbf{F}^{n+1} = (\mathbf{I} + (\nabla\mathbf{u}^{n+1})^T \Delta t)\mathbf{F}^n$ following Eqn. (17);
    perform return mapping according to certain yield condition; `// optional for elastoplasticity`
  end
  for *each particle* **do**
    update position $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{u}^h_{n+1}\Delta t$ and perform velocity blending in Eqn. (12);
  end

until *simulation stops*;

Other Results

Pseudo-dog

Other Results

Tearing Wet Paper

Conclusion and Future Work

An extensible pure particle-based simulation framework is established for multiphase continuum.

- Is particle-based, thus is simple for implementation,
- Uniformly handles various materials using continuum model,
- Captures various multiphase phenomenon using phase-field model.

Limitation and Future Work

- Explicit time scheme reduces performance for stiff and viscous materials
  - implicit scheme, parallel acceleration
- Artificial fracture can happen for large deformation
  - variable kernel radius, Lagrangian formulation
- Percolation is not captured
  - advanced phase field model

Thanks for your attention!
Welcome for questions!