Problem Statement & Main Contributions

Background
- Granular flows ubiquitous in nature and many industrial processes
- No underlying governing equations for general granular flow exist!
- Simulations with the Discrete Element Method (DEM, Cundall et al., 1979)
  Granular flow simulation data with open-source DEM software LIGGGHTS (Kloos et al., 2012)
- LIGGGHTS allows simulation of particulate flows
- complex mesh-based wall geometries
- enables simulation of relevant industrial processes
- interest in machine learning models that can predict simulation trajectories
- Gaining speedup by machine learning models

Compared to previous work (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2020)
Focus on learning 3D granular particle flow simulations with artificial geometric boundary conditions

Main contributions:
- Triangular geometric boundaries for Graph Neural Networks (GNNs)
- Orientation independence of normal vectors
- Compare and analyze simulated processes

Orientation Independence

Particle - Wall Interactions
- Normal vector components as features to describe walls
- Vector representation is orientation dependent, while particle - wall interactions do not depend on this representation!

Just using both orientations has problem that an order still is there.
⇒ Define partial ordering,

$$ f_i(m) = \sum_{m' = m}^{3} \epsilon^{i-1} \epsilon(m_i + 1) $$

$$ n_i = f_i(m) $$

$$ n_0 = f_0(m) $$

$$ \epsilon^{\text{expr}(m)} = \begin{cases} n_i - n_0 & \text{if } n_0 \leq n_i \\ -n_0 & \text{otherwise} \end{cases} $$

Mixing Entropy

- proposed by Lai et al. (1975)
- quantity extend of particle mixing
- local entropy \( S(x_{klm}, t) \) at grid cell \( x_{klm} \)
- splintering particles into two classes \( +1, -1 \) at a certain time step \( t \)

$$ S(t) = \sum_{t=0}^{N} \sum_{m=0}^{3} f_i(x_{klm}, t) f_{i+1}(x_{klm}, t) $$

Analysis of ML Simulation Outputs

Application to Granular Flow Data

Triangular Geometric Boundaries

- geometry described by triangular mesh
- static boundary particles in large number of additional particles
- insert virtual particles as needed into graph
- needs distance from particles to triangles
- usage of algorithms as adopted from Eberly (1999) (see figure)

References

Eberly D. (1999). Distance between point and triangle in 3D.
Kloos C. et al. (2012). Models, algorithms and validation for open-source DEM and CFD-DEM.
Pfaff T. et al. (2020). Learning Mass-Based Simulation with Graph Networks.
Sanchez-Gonzalez A. et al. (2020). Learning to Simulate Complex Physics with Graph Networks.