Druker-Prager Elastoplasticity for Sand Animation

Reading Notes

Frank Madrid

March 12, 2018

1 Paper Contributions

- Builds on the work of **??** and develops an implicit version of their Drucker-Prager-based elastoplasticity model for granular materials.
- Uses the Material Point Method (MPM) to discretize the Drucker-Prager Elastoplasticity model since it provides a natural and efficient way of treating contact, topological change and history dependent behavior.
- Replace the particle/grid transfers used by **??** with APIC transfers **??** which allows for more stable behavior with simulations that have higher numbers of particles per cell.

2 Concepts

2.1 Conservation of Mass

For any system closed to all transfers of matter and energy, the mass of the system must remain constant over time.

Definition 1. Let the density and velocity of our object be described at each location by $\rho(x, y, r)$ and $\mathbf{v}(x, y, z)$, then the **conservation of mass** is

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{v} = 0.$$

where $\frac{D\rho}{Dt}$ is the *material derivative* and describes the rate of change of density with respect to time while $\nabla \cdot \mathbf{v}$ is the div $\mathbf{v}$. **Recall**: If $\mathbf{F}(x, y, z)$ is the velocity of a fluid, then div $\mathbf{F}(x, y, z)$ represents the net rate of change of the mass of fluid flowing from the point (x, y, z) per unit volume.

Interpretation For a given closed surface in the system, the change in time of the mass enclosed by the surface is equal to the mass that traverses the surface, positive if matter goes in and negative if matter goes out.

2.2 Elastic and Plastic Deformation Gradient

Definition 2. The deformation gradient $\frac{d\mathbf{F}}{dt}$ is a measure of how a material has locally rotated and deformed due to its motion. The deformation gradient is factored into its elastic $\mathbf{F}^E$ and plastic $\mathbf{F}^P$ parts such that $\mathbf{F} = \mathbf{F}^E\mathbf{F}^P$.

The plastic deformation represents the portion of the material's history that has been *forgotten* (cannot be recovered) while the elastic deformation is the strain, or the tendency to revert back to its original form.

2.3 Constitutive Model

2.4 Discretization

Discretizations are typically either Eulerian or Lagrangian, which differ by their frame of reference.

An **Eulerian** description computes quantities of interest at fixed locations in space which feature fixed grids and are ideal for handling collisions and changes in topology.

A **Lagrangian** description uses quantities that move with the material being described and tend to use moving particles often connected by a mesh which provides a straightforward way to determine how deformed the material is.

The **Material Point Method** stores information on Lagrangian particles, that is each particle stores its mass and velocity, but computes forces using a fixed Eulerian grid. Since MPM uses two distinct representations, information must be transferred between them.

2.5 Particle States

In the MPM model, state representations are stored on particles. Each particle maintains its mass m_p, position x_p, velocity v_p, affine momentum $\mathbf{B}_p$ which is used for APIC transfers and approximates the spatial derivative of the grid velocity field at the end of the previous time step, and the elastic and plastic components of the deformation gradient.

2.6 Weights

During the information transformation process between particle and grid representations, each particle p and grid node i is associated with a weight w_{ip}^n at time index n which determines how strongly the particle and node interact.

Weights are computed based on a kernel $w_{ip} = N(\mathbf{x}_p{}^n - \mathbf{x}_i{}^n)$, where $\mathbf{x}_p{}^n$ and $\mathbf{x}_i{}^n$ are the locations of the particle and grid node locations.

2.7 Kernel

3 Algorithm Overview

3.1 Transfer To Grid

During this step, mass and momentum are transferred from each particle to the grid and which is used to compute velocity on the grid. Mass is transferred from each particle to its neighboring grid nodes using

$$m_i^n = \sum_p w_{ip}^n m_p$$

or the mass at any particular grid node i is the weighted sum of the masses of each particle p where $w_{ip}^n = N(\mathbf{x}_p^n - \mathbf{x}_i^n)$.

Since the momentum contribution from particle p to node i is $w_{ip}^n m_p \bar{\mathbf{v}}_p(\mathbf{x}_i^n)$, the full form of the velocity transfer is

$$\mathbf{v}_i^n = \frac{1}{m_i^n} \sum_p w_{ip}^n m_p (\mathbf{v}_p^n + \ldots (\mathbf{x}_i^n - \mathbf{x}_p^n))$$

3.2 Grid Update

The grid velocities are updated by applying forces and processing for collisions and scripted objects.

$${\mathbf{v}_i}^i = {\mathbf{v}_i}^n + \frac{\Delta t}{{m_i}^n} \mathbf{f}_i(\langle {\mathbf{F}_p}^{E,n} \rangle)$$

where $\mathbf{f}_i(\langle \mathbf{F}_p^{E,n} \rangle)$ are the forces at grid node i as a result of the elastic forces in the deformation gradient. After forces are applied, we can process the velocities for collisions ${\mathbf{v}_i}^* \to \bar{\mathbf{v}}_i^{n+1}$ then apply friction $\bar{\mathbf{v}}_i^{n+1} \to \tilde{\mathbf{v}}_i^{n+1}$

3.3 Transfer To Particles

Velocities are interpolated back to particles using

$${\mathbf{v}_p}^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_i^{n+1}.$$

The transfer for ${B_p}^{n+1}$ is

$${B_p}^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_i^{n+1} ({\mathbf{x}_i}^n - {\mathbf{x}_p}^n)^T.$$

3.4 Update Particle States

Recall: The particle state includes information about the particle's mass, position, velocity, affine momentum and deformation gradient. Mass is constant, while velocity and affine momentum were updated in a previous step. Leaving position and deformation gradient state information.

Particle positions are updated by interpolating moving grid node positions

$$x_p^{n+1} = \sum_i w_{ip}^n \bar{\mathbf{x}}_i^{n+1}$$

where $\bar{\mathbf{x}}_i^{n+1}$ is the particle's position after applying velocities as a result of collisions. Since $\frac{D\mathbf{F}}{Dt} = (\nabla \mathbf{v})\mathbf{F}$ is just a normal time derivative,

$$\mathbf{F}_p^{n+1} = \mathbf{F}_p^n + \delta t (\nabla \mathbf{v})_p \mathbf{F}_p^n$$

where

$$(\nabla \mathbf{v})_p = \sum_p \bar{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T$$

3.5 Plasticity and Hardening

3.6 Implicit Velocity Update

3.7 Initialization

4 Forces

Forces are obtained by differentiating a discretization of the potential energy with respect to the motion of grid nodes. Potential energy is computed by integrating the energy density with respect to volume,

$$\Psi = \int_\Omega \psi dV$$

The potential energy is discretized over a sum on particles,

$$\Psi = \sum_p V_p^0 \psi(\mathbf{F}_p^E)$$

where the force on a particular particle is

$$\mathbf{f}_p = -\frac{\partial \Psi}{\partial x_p}$$

The potential energy of the material can be expressed in terms of the locations of the grid nodes. This relationships are used to compute forces on the grid nodes.

$$\Psi(\langle \mathbf{x}_i \rangle) = \sum_p V_p^0 \psi\big(\mathbf{F}_p^E(\langle \mathbf{x}_i \rangle)\big)$$

$$\Delta \mathbf{F}_p^E(\langle \mathbf{x}_i \rangle) = \mathbf{F}_p^{E,n} \sum_i (\mathbf{x}_i - \mathbf{x}_i^n)(\nabla w_{ip}^n)^T$$

4.1 Constitutive Model

The model is written in terms of the singular value decomposition $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ as

$$\psi(\mathbf{F}) = \mu \text{trace}((\ln \mathbf{\Sigma})^2) + \frac{1}{2}\lambda(\text{trace}(\ln \mathbf{\Sigma}))^2$$