

# DP Elastoplasticity for Sand Animation - Reading Notes

Frank Madrid

March 9, 2018

## 1 Introduction

### 1.1 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 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 Mast with APIC transfers ?? which allows for more stable behavior with simulations that have higher numbers of particles per cell.

## 2 Background

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

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

and the **conservation of momentum** is

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

where  $\boldsymbol{\sigma}$  is the internal stress and  $\mathbf{g}$  gravity.

The law of conservation of mass indicates the rate of change of the density of an object with respect to time is equal net flow of  $\mathbf{v}$

**Definition 2.** An **Euler-Lagrange** equation is a second-order partial differential equation whose solutions are the functions for which a given functional is stationary.

**Definition 3.** The **deformation gradient** represents how deformed a material is locally. The deformation gradient  $\mathbf{F}$  evolves according to

$$\frac{D\mathbf{F}}{dt} = (\nabla \mathbf{v})\mathbf{F}.$$

## 2.1 Conservation of Mass

**Alternate Forms**

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

**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 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.

## 2.3 Kernel

Choosing a kernel  $N$  leads to trade offs with respect to smoothness, computational efficiency, and the width of the stencil.

Tensor product splines are preferred for their computational efficiency.

## 2.4 Elastic and Plastic Deformation Gradient

Plasticity is represented by factoring the deformation gradient into its *elastic* and *plastic* parts  $\mathbf{F} = \mathbf{F}^E \mathbf{F}^P$ . Consider a force which coils a metal rod. The **plastic part** of the deformation gradient  $\mathbf{F}^P$  represents the portion of the material's history that has been forgotten while the **elastic part** of the deformation gradient  $\mathbf{F}^E$  represents the strain or deformation of the metal coil. Thus to compute stress,  $\mathbf{F}^E$  should be used.

# 3 Overview

## 3.1 Transfer To Grid

During this step, we transfer mass and momentum from each particle to the grid and use this information to compute velocity on the grid.

### 3.2 Grid Update

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

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

After forces are applied, we can process the velocities for collisions  $\mathbf{v}_i^* \rightarrow \bar{\mathbf{v}}_i^{n+1}$  and then apply friction  $\bar{\mathbf{v}}_i^{n+1} \rightarrow \tilde{\mathbf{v}}_i^{n+1}$

### 3.3 Transfer To Particles

Velocities are interpolated back to particles using

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

?? uses normalized weights,

$$\mathbf{v}_i^n = \sum_p \mathbf{v}_p^n m_p \frac{w_{ip}^n}{m_i^n}$$

since weighting with  $w_{ip}^n$  does not result in momentum conservation while ?? does not. Does this have to do with the affine momentum  $\mathbf{B}$ ?

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

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

or the sum of the weighted post-collision velocities for each grid point  $i$  times the displacement vector of grid point  $i$  and the particular particle  $p$ .

### 3.4 Update Particle State

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 position of the particle after applying the velocities as a result of collisions but before applying friction.