SUPPLEMENTAL DOCUMENT: GPU OPTIMIZATION OF MATERIAL POINT METHODS

MING GAO, XINLEI WANG, KUI WU, ANDRE PRADHANA, EFTYCHIOS SIFAKIS, CEM YUKSEL, AND CHENFANFU JIANG

Contents

1. MLS-MPM Implicit Heat Solver.

1.1. Weak Form

MPM is a general spatial discretization scheme (just like FEM) and can be used to solve the heat equation. A straightforward application in computer graphics is melting and freezing, where mechanical material parameters depend on temperature. A more complex case is thermo-elastoviscoplasticity, where thermal expansion and frictional heating effects are taken into account through thero-mechanical coupled equations. Here we only focus on deriving the weak form discretization on MLS-MPM (Hu et al., 2018) for the heat conduction equation.

Let's start from the Eulerian-form heat equation

$$\rho C \frac{D\theta}{Dt} - \nabla \cdot \kappa \nabla \theta + q^{ext}(\mathbf{x}) = 0,$$

where $\theta(\mathbf{x}, t)$ is temperature, $\rho(\mathbf{x}, t)$ is density, $c(\mathbf{x}, t)$ is Eulerian specific heat capacity per unit mass (with its Lagrangian counter part $C(\mathbf{X}, t)$), $\kappa(\mathbf{x}, t)$ is heat conductivity, q^{ext} is external body heat source (let's ignore this term, this one is similar to gravity in the momentum equation and can encode effects such as radiation). The Eulerian weak form of this PDE is

$$\int_{\Omega^t} \rho c \frac{D\theta}{Dt} - \nabla \cdot \kappa \nabla \theta d\mathbf{x} = 0.$$

We follow the same way of discretizing the momentum equation using a Galerkin style weak form (Jiang et al., 2016). Commonly in MPM, we take an updated Lagrangian view and look at t^n. Now our temperature θ is like the velocity in the momentum equation. It relates to its Lagrangian counter-part Θ as $\theta^{n+1}(\mathbf{x}) = \Theta(\Phi^{-1}(\mathbf{x}, t^n), t^{n+1})$ and $\theta^n(\mathbf{x}) = \Theta(\Phi^{-1}(\mathbf{x}, t^n), t^n)$.

1.2. MLS Shape Functions

For any test function $w(\mathbf{x}, t^n)$ in the proper function space, the time discretization reveals

$$\begin{aligned} &\frac{1}{\Delta t} \int_{\Omega^{t^n}} w(\mathbf{x}, t^n) \rho(\mathbf{x}, t^n) c(\mathbf{x}, t^n) \left(\theta^{n+1}(\mathbf{x}) - \theta^n(\mathbf{x})\right) d\mathbf{x} \\ &= \int_{\partial\Omega^{t^n}} \left(w(\mathbf{x}, t^n) \kappa(\mathbf{x}, t^n) \nabla \theta(\mathbf{x}, t^n)\right) \cdot \mathbf{n} ds(\mathbf{x}) - \int_{\Omega^{t^n}} \nabla w(\mathbf{x}, t^n) \cdot \left(\kappa(\mathbf{x}, t^n) \nabla \theta(\mathbf{x}, t^n)\right) d\mathbf{x}. \end{aligned}$$

The next step is MLS style spacial discretization. Following (Hu et al., 2018), we adopt the MLS shape function $\Phi_{\mathbf{i}}(\mathbf{x})$ at each node near a particle to discretize both the test function and the physical fields. That is, we do

$$w^n = w^n_{\mathbf{i}} \Phi_{\mathbf{i}}, \quad \theta^n = \theta^n_{\mathbf{j}} \Phi_{\mathbf{j}}, \quad \theta^{n+1} = \theta^{n+1}_{\mathbf{j}} \Phi_{\mathbf{j}}. \tag{1}$$

We refer to (Hu et al., 2018) for the construction of the MLS shape functions in a way that is consistent with Element Free Galerkin (EFG) methods.

Then

$$\begin{aligned}&\frac{1}{\Delta t}\int_{\Omega^{t^n}} w_{\mathbf{i}}^n \Phi_{\mathbf{i}} \rho(\mathbf{x},t^n)c(\mathbf{x},t^n)\theta_{\mathbf{j}}^{n+1}\Phi_{\mathbf{j}}d\mathbf{x} - \frac{1}{\Delta t}\int_{\Omega^{t^n}} w_{\mathbf{i}}^n \Phi_{\mathbf{i}} \rho(\mathbf{x},t^n)c(\mathbf{x},t^n)\theta_{\mathbf{j}}^{n}\Phi_{\mathbf{j}}d\mathbf{x}\\ &= \int_{\partial\Omega^{t^n}} \left(w_{\mathbf{i}}^n \Phi_{\mathbf{i}}\kappa(\mathbf{x},t^n)\theta_{\mathbf{j}}^n\nabla\Phi_{\mathbf{j}}\right)\cdot\mathbf{n}ds(\mathbf{x}) - \int_{\Omega^{t^n}} w_{\mathbf{i}}^n\nabla\Phi_{\mathbf{i}}\cdot\left(\kappa(\mathbf{x},t^n)\theta_{\mathbf{j}}^n\nabla\Phi_{\mathbf{j}}\right)d\mathbf{x}.\end{aligned}$$

Utilizing the MLS shape functions avoids differentiating B-spline kernels in high dimensions. More specifically, if a linear polynomial space with quadratic B-spline weighting is chosen for the MLS reconstruction, we have (Hu et al., 2018)

$$\nabla\Phi_i(\mathbf{x}_p) = D_p^{-1}N_{\mathbf{i}}(\mathbf{x}_p^n)(\mathbf{x}_i - \mathbf{x}_p^n),$$

where $D_p = \frac{1}{4}\Delta x^2$ for quadratic B-spline weighting in $N_{\mathbf{i}}(\mathbf{x})$.

1.3. Lumped Mass

Similarly to how mass matrix was definened in the momentum case, we can define a "thermal mass" matrix

$$\mathcal{M}_{ij}^n = \int_{\Omega^{t^n}} \Phi_i(\mathbf{x})\rho(\mathbf{x},t^n)c(\mathbf{x},t^n)\Phi_j(\mathbf{x})d\mathbf{x}$$

and rewrite the equation as

$$\begin{aligned}&\frac{1}{\Delta t}w_{\mathbf{i}}^n\theta_{\mathbf{j}}^{n+1}\mathcal{M}_{ij}^n - \frac{1}{\Delta t}w_{\mathbf{i}}^n\theta_{\mathbf{j}}^n\mathcal{M}_{ij}^n\\ &= \int_{\partial\Omega^{t^n}} \left(w_{\mathbf{i}}^n \Phi_{\mathbf{i}}\kappa(\mathbf{x},t^n)\theta_{\mathbf{j}}^n\nabla\Phi_{\mathbf{j}}\right)\cdot\mathbf{n}ds(\mathbf{x}) - \int_{\Omega^{t^n}} w_{\mathbf{i}}^n\nabla\Phi_{\mathbf{i}}\cdot\left(\kappa(\mathbf{x},t^n)\theta_{\mathbf{j}}^n\nabla\Phi_{\mathbf{j}}\right)d\mathbf{x}. \qquad (2)\end{aligned}$$

Note that

$$\begin{aligned}\mathcal{M}_{ij}^n &= \int_{\Omega^{t^n}} \Phi_i(\mathbf{x})\rho(\mathbf{x},t^n)c(\mathbf{x},t^n)\Phi_j(\mathbf{x})d\mathbf{x}\\ &\quad \text{Pull back from the Eulerian frame to the Lagrangian frame..}\\ &= \int_{\Omega^0} \Phi_i(\mathbf{x}(\mathbf{X}))R(\mathbf{X},t)C(\mathbf{X},t^n)\Phi_j(\mathbf{x}(\mathbf{X}))J(\mathbf{X},t)d\mathbf{X}\\ &= \int_{\Omega^0} \Phi_i(\mathbf{x}(\mathbf{X}))R(\mathbf{X},0)C(\mathbf{X},t^n)\Phi_j(\mathbf{x}(\mathbf{X}))d\mathbf{X}\\ &\quad \text{Adopting one point quadrature over particle domains..}\\ &\approx \sum_p m_p C_p \Phi_i(\mathbf{x}_p)\Phi_j(\mathbf{x}_p)\end{aligned}$$

Now in Equation 2 we choose

$$w_{\mathbf{i}}^n = \begin{cases} 1, & \mathbf{i} = \hat{\mathbf{i}} \\ 0, & \text{otherwise}\end{cases}$$

then

$$\sum_j \frac{1}{\Delta t}(\theta_j^{n+1} - \theta_j^n)\mathcal{M}_{\hat{i}j}^n = \int_{\partial\Omega^{t^n}} \Big(\Phi_{\hat{i}}\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) \cdot \mathbf{n} ds(\mathbf{x}) - \int_{\Omega^{t^n}} \nabla\Phi_{\hat{i}} \cdot \Big(\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) d\mathbf{x}. \tag{3}$$

This is the equation for node $\hat{i}$.

Inverting a full mass matrix is not possible in MPM due to its potential singularity. A common strategy is diagonal mass lumping, meaning replacing $\mathcal{M}$ with its diagonal row-sums. Let's call each new entry $\hat{\mathcal{M}}_i^n$, then

$$\begin{aligned}
\hat{\mathcal{M}}_i^n &= \sum_j \mathcal{M}_{ij}^n \\
&= \sum_j \int_{\Omega^{t^n}} \Phi_i(\mathbf{x})\rho(\mathbf{x}, t^n)c(\mathbf{x}, t^n)\Phi_j(\mathbf{x}) d\mathbf{x} \\
&= \int_{\Omega^{t^n}} \Phi_i(\mathbf{x})\rho(\mathbf{x}, t^n)c(\mathbf{x}, t^n) d\mathbf{x} \\
&\text{Pull back..} \\
&= \int_{\Omega^0} \Phi_i(\mathbf{x}(\mathbf{X}))R(\mathbf{X}, 0)C(\mathbf{X}, t^n) d\mathbf{X} \\
&\approx \sum_p m_p C_p \Phi_i(\mathbf{x}_p).
\end{aligned}$$

Now we use $\mathcal{M}_{\hat{i}j}^n \approx \hat{\mathcal{M}}_{\hat{i}}^n \delta_{\hat{i}j}$ to rewrite Equation 3:

$$\frac{1}{\Delta t}(\theta_{\hat{i}}^{n+1} - \theta_{\hat{i}}^n)\hat{\mathcal{M}}_{\hat{i}}^n = \int_{\partial\Omega^{t^n}} \Big(\Phi_{\hat{i}}\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) \cdot \mathbf{n} ds(\mathbf{x}) - \int_{\Omega^{t^n}} \nabla\Phi_{\hat{i}} \cdot \Big(\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) d\mathbf{x}. \tag{4}$$

Let's replace $\hat{i}$ with i to get the final equation for the temperature on node i:

$$\frac{1}{\Delta t}(\theta_i^{n+1} - \theta_i^n)\hat{\mathcal{M}}_i^n = \int_{\partial\Omega^{t^n}} \Big(\Phi_i\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) \cdot \mathbf{n} ds(\mathbf{x}) - \int_{\Omega^{t^n}} \nabla\Phi_i \cdot \Big(\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) d\mathbf{x}. \tag{5}$$

Note that mC is called thermal mass and is fixed on each particle.

1.4. Heat Force

Now let's deal with the right hand side. Let's look the second one (volume integral):

$$\int_{\Omega^{t^n}} \nabla\Phi_i \cdot \Big(\kappa(\mathbf{x}, t^n)\theta_j^n \nabla\Phi_j\Big) d\mathbf{x} \approx \sum_p \big(\nabla\Phi_i(\mathbf{x}_p)\big) \cdot \Big(\kappa(\mathbf{x}_p, t^n)\theta_j^n \nabla\Phi_j(\mathbf{x}_p)\Big) V_p^n.$$

This is the 'internal force' for heat equation. Note that to go implicit we need to change the known θ_j^n to the unknown θ_j. We have

$$f_i(\theta) = \sum_p \big(\nabla\Phi_i(\mathbf{x}_p)\big) \cdot \big(\kappa_p \theta_j \nabla\Phi_j(\mathbf{x}_p)\big) V_p^n \tag{6}$$

$$= \sum_j \sum_p \left(\nabla \Phi_i(\mathbf{x}_p)\right) \cdot \left(\kappa_p \theta_j \nabla \Phi_j(\mathbf{x}_p)\right) V_p^n \tag{7}$$

$$= \sum_j \sum_p \kappa_p V_p^n \nabla \Phi_i(\mathbf{x}_p) \cdot \nabla \Phi_j(\mathbf{x}_p) \theta_j \tag{8}$$

$$= \sum_j H_{ij} \theta_j \tag{9}$$

where $H_{ij} = \sum_p \kappa_p V_p^n \nabla \Phi_i(\mathbf{x}_p) \cdot \nabla \Phi_j(\mathbf{x}_p)$. We can see the force is linear in θ. It is easy to verify that if we use a linear basis, and one particle per cell center, then this is equivalent to finite difference Laplacian operator.

The boundary term

$$h_i = \int_{\partial \Omega^{t^n}} \left(\Phi_i(\mathbf{x}) \kappa(\mathbf{x}, t^n) \theta_j^n \nabla \Phi_j \right) \cdot \mathbf{n} ds(\mathbf{x})$$

is the heat flux boundary condition. To apply this, quantity $\eta(\mathbf{x}, t^n) = (\kappa(\mathbf{x}, t^n) \nabla \theta) \cdot \mathbf{n}$ is specified at the boundary. This is analygoous to the traction boudnary conditon in MPM.

1.5. Summarization

Finally, let's summarize the implicit formulation of heat equation assuming no boundary heat flux:

$$\frac{1}{\Delta t}(\theta_i - \theta_i^n) \hat{\mathcal{M}}_i^n = -\sum_j \left(\sum_p \kappa_p V_p^n \nabla \Phi_i(\mathbf{x}_p) \cdot \nabla \Phi_j(\mathbf{x}_p) \right) \theta_j, \tag{10}$$

where θ_i is temperature of node i, $\hat{\mathcal{M}}_i^n = \sum_p m_p C_p \Phi_i(\mathbf{x}_p)$ is lumped thermal mass, C_p is specific heat capacity per unit mass, κ_p is heat conductivity. The resulting implicit system is SPD and can be efficiently solved with Conjugate Gradient on the GPU.

2. Unilateral Sand Constitutive model.

2.1. Notations

The letter $d \in \{2, 3\}$ denotes the spatial dimension of the problem. We use $\mathbf{F}$ to denote the deformation gradient of the flow-map $\boldsymbol{\varphi}$. For isotropic constitutive model, it is useful to work with the singular value decomposition of $\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T$ (Bonet and Wood, 2008). We write

$$\hat{\mathbf{F}} = \begin{pmatrix} \hat{f}_0 & 0 \\ 0 & \hat{f}_1 \end{pmatrix} \quad \text{(in 2D)}, \qquad \text{or} \qquad \hat{\mathbf{F}} = \begin{pmatrix} \hat{f}_0 & 0 & 0 \\ 0 & \hat{f}_1 & 0 \\ 0 & 0 & \hat{f}_2 \end{pmatrix} \quad \text{(in 3D)}. \tag{11}$$

Sometimes, it's convenient to use an abuse of notation and to write $\hat{\mathbf{F}}$ as a vector

$$\hat{\mathbf{F}} = \begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \end{pmatrix} \quad \text{(in 2D)}, \qquad \text{or} \qquad \hat{\mathbf{F}} = \begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \end{pmatrix} \quad \text{(in 3D)}. \tag{12}$$

Mast (2013) and Klár et al. (2016) have shown the usefulness of Hencky-strain in modeling granular material. The Hencky-strain $\mathbf{h}$ is the logarithm of the singular values of $\mathbf{F}$, i.e.

$$\hat{\mathbf{h}} = \log(\hat{\mathbf{F}}). \tag{13}$$

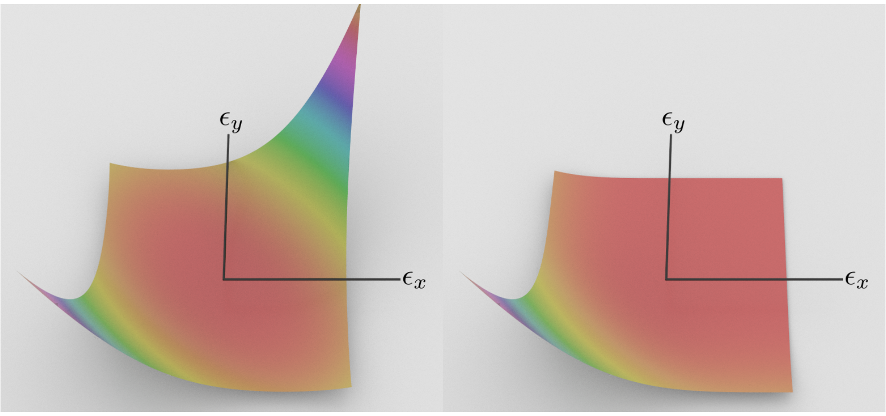

FIGURE 1. **Full v.s. unilateral.** Two-dimensional depiction of the full and unilateral quartic energy density function in the Hencky-strain space.

2.2. Energy density function

The original St. Venant-Kirchhoff with Hencky strain model is (Klár et al., 2016)

$$\hat{\psi}(\hat{\mathbf{F}}) = \mu \mathrm{tr}\Big((\log \hat{\mathbf{F}})^2\Big) + \frac{\lambda}{2}\Big(\mathrm{tr}(\log \hat{\mathbf{F}})\Big)^2. \tag{14}$$

In lieu of the original model, we propose a *quartic* energy density function

$$\hat{\psi}(\hat{\mathbf{F}}) = a\mu \sum_{i=0}^{d-1} (\log(\hat{f}_i))^4 + \frac{a\lambda}{2}\Big(\mathrm{tr}(\log(\hat{\mathbf{F}}))\Big)^4. \tag{15}$$

The coefficient a is chosen to be 6.254421582537118, which is the solution of the following minimization problem

$$a = \mathrm{argmin}_a \int_{0.25}^{1} \Big(ax^4 - \log(x)^2\Big)^2 \, \mathrm{d}x. \tag{16}$$

In order to mitigate numerical cohesion that occurs in a semi-implicit time integration scheme (where the elastic response is computed using an implicit time-integration scheme and plasticity is done as a post-process), we also consider the unilateral version of (15), namely

$$\hat{\psi}(\hat{\mathbf{F}}) = a\mu \sum_{i=0}^{d-1} (\log(\hat{f}_i))^4 H_{\{\log(\hat{f}_i)<0\}}(\log(\hat{f}_i)) + \frac{a\lambda}{2}\Big(\mathrm{tr}(\log(\hat{\mathbf{F}}))\Big)^4 H_{\{\mathrm{tr}(\log(\hat{\mathbf{F}}))<0\}}(\mathrm{tr}(\log(\hat{\mathbf{F}}))), \tag{17}$$

which is twice continuously differentiable. The function H_I denotes an indicator function, i.e. given a set $I \subseteq \mathbb{R}$, we have

$$H_I(x) = \begin{cases} 1 & \text{if } x \in I \\ 0 & \text{otherwise.} \end{cases}$$

2.3. Stress-strain relationship

Let $\mathbf{P}$ be the first Piola-Kirchhoff stress tensor. All of the models above are isotropic, and as such $\mathbf{P}$ can be written as

$$\mathbf{P} = \mathbf{U}\hat{\mathbf{P}}\mathbf{V}^\top, \tag{18}$$

with

$$\hat{\mathbf{P}} = \frac{\partial \hat{\psi}}{\partial \hat{\mathbf{F}}}. \tag{19}$$

The Cauchy stress tensor is given by

$$\boldsymbol{\sigma} = J^{-1}\mathbf{P}\mathbf{F}^\top = J^{-1}\mathbf{U}\hat{\mathbf{P}}\mathbf{V}^\top\mathbf{V}\hat{\mathbf{F}}\mathbf{U}^\top = J^{-1}\mathbf{U}\hat{\mathbf{P}}\hat{\mathbf{F}}\mathbf{U}^\top, \tag{20}$$

which motivates us to define

$$\hat{\boldsymbol{\sigma}} := \hat{\mathbf{P}}\hat{\mathbf{F}}. \tag{21}$$

The Kirchhoff stress is $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. Arguably, this is a better stress measure for our purposes since we're avoiding a division by J. From (20), we have

$$\boldsymbol{\tau} = \mathbf{U}\hat{\boldsymbol{\sigma}}\mathbf{U}^\top \quad \Longrightarrow \quad \hat{\boldsymbol{\tau}} = J\boldsymbol{\sigma}.$$

The stress-strain relationship for the constitutive model defined by (15) is

$$\hat{\boldsymbol{\sigma}} = 4a\mu(\hat{\mathbf{h}})^3 + 2a\lambda\,\mathrm{tr}(\hat{\mathbf{h}})^3\mathbf{1}, \tag{22}$$

with $\mathbf{1}$ denoting the all-ones vector and $(\hat{\mathbf{h}})^3$ is defined component-wise.

2.4. Yield surface

To model sand, we use the Drucker-Prager yield criterion. For completeness, we'll include its form for both the Cauchy and Kirchhoff stress

$$\hat{y}(\hat{\boldsymbol{\sigma}}) = \operatorname{tr}(\hat{\boldsymbol{\sigma}})\alpha + \left\| \hat{\boldsymbol{\sigma}} - \frac{\operatorname{tr}(\hat{\boldsymbol{\sigma}})}{3}\mathbf{I} \right\|_F \leq 0, \qquad \text{(Cauchy)}, \tag{23}$$

$$\hat{y}(\hat{\boldsymbol{\tau}}) = \operatorname{tr}(\hat{\boldsymbol{\tau}})\alpha + \left\| \hat{\boldsymbol{\tau}} - \frac{\operatorname{tr}(\hat{\boldsymbol{\tau}})}{3}\mathbf{I} \right\|_F \leq 0, \qquad \text{(Kirchhoff)}. \tag{24}$$

Combined with (22), we can derive the yield function in terms of the Hencky-strain $\hat{\mathbf{h}}$. We denote this relationship by

$$\bar{y}(\hat{\mathbf{h}}) \leq 0. \tag{25}$$

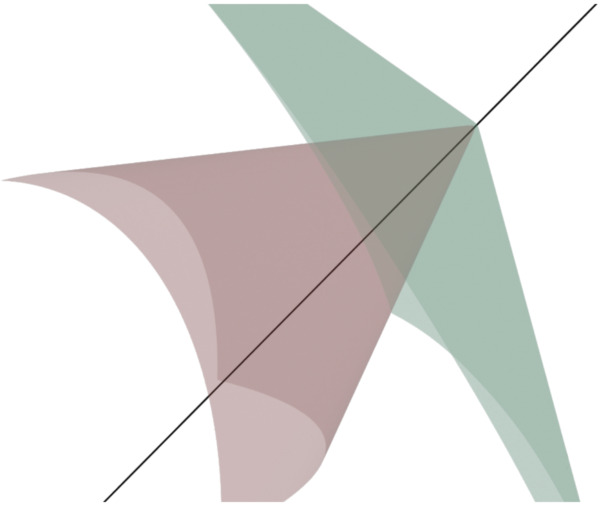

FIGURE 2. **Yield surface in Hencky-strain space.** The outer green surface is the yield function corresponding to the proposed quartic energy density and the inner red cone corresponds to the regular St. Venant-Kirchhoff with Hencky strain model. The line denotes the hydrostatic axis.

2.5. Trial and projected strain

We adopt the following notation

	Principal strain	Hencky strain	Cauchy stress
trial state	$\hat{\mathbf{F}}^{\text{tr}}$	$\hat{\mathbf{h}}^{\text{tr}}$	$\hat{\boldsymbol{\sigma}}^{\text{tr}}$
projected state	$\hat{\mathbf{F}}$	$\hat{\mathbf{h}}$	$\hat{\boldsymbol{\sigma}}$

The relationship between the trial state and the projected state is given by

$$\hat{\mathbf{h}} = \delta\gamma \hat{\mathbf{G}}|_{\hat{\mathbf{h}}} + \hat{\mathbf{h}}^{\mathrm{tr}}. \tag{26}$$

where

$$\hat{\mathbf{G}} = \mathrm{dev}\left(\frac{\partial \hat{y}}{\partial \hat{\boldsymbol{\tau}}}\right). \tag{27}$$

which is completely determined by the yield surface.

2.6. Solving the system of equations

In the projection step, there are $d+1$-unknowns, namely $\delta\gamma$ and the projected strain $\hat{\mathbf{h}}$. Equations (26) and (25) gives a total of $(d+1)$-equations to be satisfied.

2.6.1. Solving the system of equations in three-dimensions. It can be proven that the solution $\mathbf{h}$ lies in the space of $\mathrm{span}\{(1,1,1)^\top, \mathrm{dev}(\hat{\mathbf{h}}^t)\}$. The deviatoric part of $\mathbf{h}$ is

$$\mathrm{dev}(\hat{\mathbf{h}}^t) = \begin{pmatrix} -2\hat{h}_0 + \hat{h}_1 + \hat{h}_2 \\ \hat{h}_0 - 2\hat{h}_1 + \hat{h}_2 \\ \hat{h}_0 + \hat{h}_1 - 2\hat{h}_2 \end{pmatrix}.$$

So we can write

$$\hat{\mathbf{h}} = p\mathbf{o} + q\mathrm{dev}(\hat{\mathbf{h}}^t). \tag{28}$$

We can substitute this to (26) and solve for p (see `mathematica_tech_doc.nb`) to get

$$p = \frac{\hat{h}^t_0 + \hat{h}^t_1 + \hat{h}^t_2}{3}. \tag{29}$$

The yield function as a function of p and q is quite long. Its form and its derivative is attached in the `mathematica` document. One can solve for the zero of this function using Newton's method.

The algorithm pseudocode is

```
$p \leftarrow (h^{\mathrm{tr}}_0 + h^{\mathrm{tr}}_1 + h^{\mathrm{tr}}_2)/3$
**if** $y(\mathbf{h}^{\mathrm{tr}}) \leq 0$ **then**
    return
**else if** $p \geq 0$ **then**
    $\mathbf{h} \leftarrow (0,0,0)$
**else**
    $\mathbf{dev} \leftarrow (-2h^{\mathrm{tr}}_0 + h^{\mathrm{tr}}_1 + h^{\mathrm{tr}}_2, h^{\mathrm{tr}}_0 - 2h^{\mathrm{tr}}_1 + h^{\mathrm{tr}}_2, h^{\mathrm{tr}}_0 + h^{\mathrm{tr}}_1 - 2h^{\mathrm{tr}}_2)$
    $q \leftarrow -1./3.$
    solve $q$ using Newton
    $\mathbf{h} \leftarrow p(1,1,1) + q\,\mathbf{dev}$
```

References.

J. Bonet and R. Wood. 2008. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press.

Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans Graph* 37, 4 (2018).

C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle. 2016. The material point method for simulating continuum materials. In *ACM SIGGRAPH 2016 Course*. 24:1–24:52.

G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Trans Graph* 35, 4 (2016), 103:1–103:12.

C. M. Mast. 2013. *Modeling landslide-induced flow interactions with structures using the material point method.* Ph.D. Dissertation. University of Washington.