Silly Rubber: An Implicit Material Point Method for Simulating Non-equilibrated Viscoelastic and Elastoplastic Solids

Yu Fang, Minchen Li, Ming Gao, Chenfanfu Jiang

Contents

1 Viscoelasticity

In this section, we briefly discuss some continuum mechanics background for the viscoelasticity proposed and discussed in this work. For elastoplasticity we refer the readers to the previous works of [4, 2, 10, 11, 1, 3]. Specifically, the supplemental document of [4] stands as good introductory material for graphics researchers new to continuum mechanics simulation techniques. Since we focus on materials such as rubber polymers and solid foams with large elastic deformation, we assume finite strain elasticity; however, we first introduce a simple one-dimensional linear model as motivation.

1.1 Motivation: one-dimensional infinitesimal case

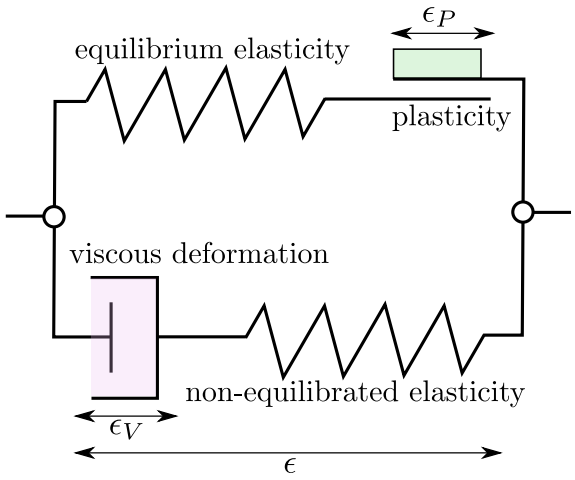

Figure 1: **1D rheological model**

As illustrated in Fig. 1, the 1D model consists of two parallel components. The top one represents elastoplasticity that can be reduced to pure elasticity when the plastic element is removed. The plastic element characterizes the response due to permanent deformation. Notice the top part of the model is time-independent, thus we refer to the elastic spring as the equilibrium spring. On the other hand, the bottom component represents the viscoelasticity via a standard Maxwell element, which is usually used to denote a time-dependent elastic response. Specifically, the dashpot captures the viscous part of the deformation due to its rate of change and does not contribute to the total elastic potential energy. The bottom part is referred to as the non-equilibrium component due to the existence of this rate-dependent viscous deformation.

The "total" stress σ exerted on the external environment from the model should be equal to the sum of the stresses contributed by each component as

$$\sigma = \sigma_{\text{top}} + \sigma_{\text{bot}}.$$

σ_{top} can be computed from either the equilibrium elasticity or the plasticity while σ_{bot} can be computed from either the non-equilibrium elasticity or the viscous element from Newton's third law. Assuming the total strain of the model is ϵ, the strain of the plasticity is ϵ_P and the strain of the viscosity is ϵ_V, the stresses can be computed as,

$$\begin{aligned}\sigma_{\text{top}} &= E_E(\epsilon - \epsilon_p)\\ \sigma_{\text{bot}} &= E_N(\epsilon - \epsilon_V) = \eta\dot{\epsilon_V}\end{aligned}$$

where E_E, E_N and η are the spring and dashpot constants respectively. From examples shown in the main paper, we can clearly see that the elastoplastic component will still determine the overall behavior while the viscosity will decide the resistance to rapid deformation.

There are two takeaway observations from this model. First, the two parallel components are independent of each other; i.e., the two stresses can be separately computed from the two spring strains without considering the other component. Second, the total stress is the sum of the two stresses from the two independent components. In the next section, this simple model will be extended to a 3D finite strain regime.

1.2 Three-dimensional case

In 3D, an energy model would be required to encode the relationship between stress and strain. In the infinitesimal regime, we can still get the equilibrated/non-equilibrated spring strains by subtracting the plasticity/viscosity strain from the whole strain; in other words, additive decomposition of the strain is used to define the energy. On the other hand, in the finite regime, multiplicative decomposition of the deformation gradient (instead of strain) is preferred:

$$\boldsymbol{F} = \boldsymbol{F}_E \boldsymbol{F}_P = \boldsymbol{F}_N \boldsymbol{F}_V \tag{1}$$

From Eqn. (1), it seems that the elastoplastic and viscoelastic parts can be handled independently and the resulting stresses are superimposed to evolve the dynamic deformation (similar to the 1D case). A detailed proof is provided in [5].

Starting from the Clausius-Duhem inequality, the form of the second law of thermodynamics used in continuum mechanics, Reese and Govindjee [8] proved that there exists a pseudo dissipation potential Φ_V with an appropriate convex, positive form such that the total entropy does not decrease. The potential adopted in our method in the main paper is a simple linear case whereas complex nonlinear models are found in the literature.

Given this potential energy, the standard prediction-correction method in elastoplasticity [4] can be extended to tackle viscoelasticity without many changes. Considering finite deformation, the ultimate goal is always to solve for the deformation gradient at the new time step t^{n+1}. However, as shown in [4], it is more straightforward, yet equivalent, to solve for the right Cauchy-Green strain $\boldsymbol{b}_N = \boldsymbol{F}_N \boldsymbol{F}_N^T$ at the new time step t^{n+1}. Its evolution equation can be written as

$$\frac{\boldsymbol{D}\boldsymbol{b}_N}{\boldsymbol{D}t} = (\nabla \boldsymbol{v})\boldsymbol{b}_N + \boldsymbol{b}_N(\nabla \boldsymbol{v}) + \mathcal{L}_v \boldsymbol{b}_N \tag{2}$$

where $\mathcal{L}_v \boldsymbol{b}_N$ is the Lie derivative encoding the deformation change due to the inelasticity. The first two terms correspond to solving for the trial state of $\boldsymbol{b}_N^{\text{tr}}$ by only considering the pure non-equilibrated elasticity part; meanwhile, the last term corresponds to the correction step where the viscosity is taken into consideration to rectify $\boldsymbol{b}_N^{\text{tr}}$ to $\boldsymbol{b}_N^{\text{n+1}}$. In other words, operator splitting is employed. Similar to Klar et al. [4], only isotropic models are considered, and the derivations is all performed in diagonal space due to the simplicity it brings.

For completeness, we also include some materials from the main paper to explain the correction step for viscoelasticity. It turns out that the correction from $\boldsymbol{F}_N^{\text{tr}}$ to $\boldsymbol{F}_N^{\text{n+1}}$ can also be explicitly written as a projection. In particular for isotropic materials, if we adopt the singular value decomposition $\boldsymbol{F}_N = \boldsymbol{U}_N \Sigma_N \boldsymbol{V}_N{}^T$ and introduce the principal diagonal Hencky strain $\boldsymbol{\epsilon}_N = \langle \log \Sigma_N \rangle$ where operator $\langle\rangle$ maps a diagonal matrix to its vector form, then the implicit discretization of $\frac{D\boldsymbol{b}_N}{Dt} = \boldsymbol{\mathcal{L}_v} \boldsymbol{b}_N$ can be shown to reduce to a potentially nonlinear equation system for $\boldsymbol{\epsilon}^{n+1}$ [6].

$$\boldsymbol{\epsilon}_N^{n+1} = \boldsymbol{\epsilon}_N^{\text{tr}} - \Delta t \frac{\partial \Phi_V}{\partial \tau_N}(\tau_N^{n+1}(\boldsymbol{\epsilon}_N^{n+1})), \tag{3}$$

where $\tau_N = \frac{\partial \Psi_N}{\partial \boldsymbol{\epsilon}_N}$ is the principal Kirchhoff stress in vector form.

Non-equilibrated elasticity. For the non-equilibrated energy density Ψ_N, we adopt the St. Venant Kirchhoff model with the Hencky strain measure, which can be expressed in the principal strain space Σ_N

as $\Psi_N(\Sigma_N) = \mu_N \mathrm{tr}\left((\log\Sigma_N)^2\right) + \frac{1}{2}\lambda_N\left(\mathrm{tr}(\log\Sigma_N)\right)^2$, with the corresponding principal Kirchhoff stress

$$\tau_N = \frac{\partial \Psi_N}{\partial \boldsymbol{\epsilon}_N} = 2\mu_N \boldsymbol{\epsilon}_N + \lambda_N \mathrm{tr}(\boldsymbol{\epsilon}_N)\mathbf{1}, \tag{4}$$

where $\mathbf{1}$ denotes the all-ones vector, μ_N and λ_N are the material Lamé parameters.

Dissipation potential. The pseudo-potential of dissipation $\Phi_V(\tau_N)$ is chosen to be quadratic in terms of the principal non-equilibrated stress: $\Phi_V(\tau_N) = \frac{1}{2\nu_d}|\mathrm{dev}(\tau_N)|^2 + \frac{1}{9\nu_v}(\tau_N \cdot \mathbf{1})^2$, where $\mathrm{dev}(\tau_N) = \tau_N - \frac{1}{d}(\tau_N \cdot \mathbf{1})\mathbf{1}$ is the deviatoric part of the principal stress and $d = 2$ or 3 is the problem dimension. ν_d and ν_v are viscosity parameters separately controlling the dissipation on the deviatoric and dilational parts of the non-equilibrated stress. The derivative of the dissipation potential is

$$\frac{\partial \Phi_V}{\partial \tau_N}(\tau_N) = \frac{1}{\nu_d}\mathrm{dev}(\tau_N) + \frac{2}{9\nu_v}\mathrm{tr}(\tau_N)\mathbf{1}, \tag{5}$$

which separately controls the viscous flow on the deviatoric and dilational non-equilibrated stress.

Correcting the strain. Substituting Eq. (4) and (5) into (3) gives $\boldsymbol{\epsilon}_N^{n+1} = \boldsymbol{\epsilon}_N^{\mathrm{tr}} - \Delta t\left(\alpha\boldsymbol{\epsilon}_N^{n+1} + \beta\mathrm{tr}(\boldsymbol{\epsilon}_N^{n+1})\mathbf{1}\right)$, where $\alpha = \frac{2\mu_N}{\nu_d}$ and $\beta = \frac{2(2\mu_N + \lambda_N d)}{9\nu_v} - \frac{2\mu_N}{\nu_d d}$ are constants. It has an analytic solution of

$$\boldsymbol{\epsilon}_N^{n+1} = A\left(\boldsymbol{\epsilon}_N^{\mathrm{tr}} - B\mathrm{tr}(\boldsymbol{\epsilon}_N^{\mathrm{tr}})\mathbf{1}\right), \tag{6}$$

where $A = \frac{1}{1+\Delta t\alpha}$, $B = \frac{\Delta t\beta}{1+\Delta t(\alpha+d\beta)}$. Note that more complex nonlinear dissipation potentials for $\Phi_V(\tau_N)$ could also be chosen for more versatile dissipation behaviours, which could potentially cause the equations for $\boldsymbol{\epsilon}_N^{n+1}$ to become a nonlinear system that requires Newton's method for the solution.

Since $\boldsymbol{\epsilon}^{n+1} = \langle\log(\Sigma_N^{n+1})\rangle$, Eq. (6) defines an analytic and smooth projection $\boldsymbol{F}_N^{n+1} = \mathcal{Z}_N(\boldsymbol{F}_N^{\mathrm{tr}})$. Equivalently, the projection can be expressed in the principal space as $\Sigma_N^{n+1} = \hat{\mathcal{Z}}_N(\Sigma_N^{\mathrm{tr}})$, whose derivative is given by

$$\frac{\partial \hat{\mathcal{Z}}_N}{\partial \Sigma_N}(\Sigma_N^{\mathrm{tr}}) = \Sigma_N^{n+1} A\left((\Sigma_N^{\mathrm{tr}})^{-1} - B\mathbf{1}\left\langle(\Sigma_N^{\mathrm{tr}})^{-1}\right\rangle^T\right).$$

1.3 Discussion

Another popular approach to the visco-elastic-plastic model consists of a standard linear solid in series with a plastic element, i.e., $\boldsymbol{F} = \boldsymbol{F}_E\boldsymbol{F}_P\boldsymbol{F}_V$. However, this model would lead to the strong coupling of viscosity and elastoplasticity [5, 6], which largely complicates the solve. In contrast, in our model, the viscoelasticity and elastoplasticity are decoupled and can be tackled independently leading to a much simpler problem.

2 ADMM Solver

Considering that the equilibrated projection and the non-equilibrated projection are independent of each other and possess no fundamental differences with regards to our ADMM solver, we will drop the superscript $\star \in \{E, N\}$ for simplicity. Furthermore, we will first only consider the pure elasticity and defer the discussion of elastoplasticity/viscoelasticity to Section 2.4.

2.1 Formulation

Given a pure elasticity model, the time integration can be formulated by

$$\begin{aligned} \underset{d\boldsymbol{v},\boldsymbol{F}}{\mathrm{argmin}} \quad & \frac{1}{2}d\boldsymbol{v}^T\boldsymbol{M}d\boldsymbol{v} + \Psi(\boldsymbol{F}) - \Delta t d\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{g} \\ \text{subject to} \quad & \mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b}) = 0, \end{aligned}$$

where $\boldsymbol{M}$ is the nodal mass matrix, $\Psi(\boldsymbol{F})$ is the elastic potential energy, $\boldsymbol{g}$ is the gravitational acceleration, and the unknowns are both the particles' deformation gradient $\boldsymbol{F}$ and the nodal velocity change $d\boldsymbol{v}$. The definitions of $\boldsymbol{D}$ and $\boldsymbol{b}$ become clear if we write down how the deformation gradient is evolved as in the main paper. Notice $\mathbf{W}$ is the weight for the constraint connecting the two sets of unknowns. As has been shown in the literature, $\mathbf{W}$ plays a crucial role in improving the convergence of the ADMM solver.

The Lagrangian and augmented Lagrangian can be written as

$$L = \frac{1}{2} d\boldsymbol{v}^T \boldsymbol{M} d\boldsymbol{v} + \Psi(\boldsymbol{F}) - \Delta t d\boldsymbol{v}^T \boldsymbol{M}\boldsymbol{g} + \mathbf{y} : [\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})]$$
$$L_\rho = L + \frac{\rho}{2} ||\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})||_F^2$$

where $\mathbf{y}$ is the dual variable and ρ is the penalty coefficient. To us, ρ is the uniform scaling parameter shared by all particles while $\mathbf{W}$'s are more like the independent scaling parameters for every single particle.

ADMM alternates the updates of $\boldsymbol{F}$, $d\boldsymbol{v}$ and $\mathbf{y}$ as

$$\boldsymbol{F}^{n+1} = \underset{\boldsymbol{F}}{\operatorname{argmin}}\, L_\rho(\boldsymbol{F}, d\boldsymbol{v}^n, \mathbf{y}^n)$$
$$d\boldsymbol{v}^{n+1} = \underset{d\boldsymbol{v}}{\operatorname{argmin}}\, L_\rho(\boldsymbol{F}^{n+1}, d\boldsymbol{v}, \mathbf{y}^n)$$
$$\mathbf{y}^{n+1} = \underset{\mathbf{y}}{\operatorname{argmax}}\ \operatorname{Inf}\ \{L_\rho(\boldsymbol{F}^{n+1}, d\boldsymbol{v}^{n+1}, \mathbf{y}) - \frac{1}{2\rho}||\mathbf{y} - \mathbf{y}^n||_F^2\}$$

To avoid possible confusion, we stick to dual variable $\mathbf{y}$ instead of its scaled version as in [7].

Local step. For each particle p, we solve

$$\boldsymbol{F}^{n+1} = \underset{\boldsymbol{F}}{\operatorname{argmin}}\, \Psi(\boldsymbol{F}) + \mathbf{y} : (\mathbf{W}\boldsymbol{F}) + \frac{\rho}{2} ||\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})||_F^2$$

by computing the gradient,

$$0 = V\boldsymbol{P}^{n+1} + \mathbf{W}^T\mathbf{y} + \rho\mathbf{W}^T[\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})]$$
$$V\boldsymbol{P}^{n+1} + \rho\mathbf{W}^T\mathbf{W}\boldsymbol{F} = \rho\mathbf{W}^T[\mathbf{W}(\boldsymbol{D}d\boldsymbol{v} + \boldsymbol{b})] - \mathbf{W}^T\mathbf{y}$$
$$V\boldsymbol{P}^{n+1} + \rho\mathbf{W}^T\mathbf{W}\boldsymbol{F} = \rho\mathbf{W}^T[\mathbf{W}(\boldsymbol{D}d\boldsymbol{v} + \boldsymbol{b} - \frac{1}{\rho}\mathbf{W}^{-1}\mathbf{y})] \tag{7}$$

The local step is a nonlinear problem of $\boldsymbol{F}$. For efficiency, we solve it in the diagonal space with a standard Newton solver. Assuming the SVD decomposition of the right-hand side (excluding ρ and $\mathbf{W}$) of Eqn. (7) gives the two rotation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, we can apply left-multiplication with $\boldsymbol{U}^T$ and right-multiplication with $\boldsymbol{V}$ to both sides to convert to the diagonal space [7] as

$$V\hat{\boldsymbol{P}}^{n+1} + \rho\mathbf{W}^T\mathbf{W}\hat{\boldsymbol{F}} = \rho\mathbf{W}^T\mathbf{W}\boldsymbol{\Sigma}$$

where $\boldsymbol{\Sigma}$ is the corresponding singular values. Notice that we enforce W to have the form of wI, where w is a scalar, when applying Ruiz equilibration. To always guarantee the Newton direction is a descent direction, we need to do positive definite projection to the constitutive model's Hessian.

Global step. We solve globally

$$d\boldsymbol{v}^{n+1} = \underset{d\boldsymbol{v}}{\operatorname{argmin}}\, \frac{1}{2} d\boldsymbol{v}^T \boldsymbol{M} d\boldsymbol{v} - \Delta t d\boldsymbol{v}^T \boldsymbol{M}\boldsymbol{g} - \mathbf{y} : \mathbf{W}\boldsymbol{D}d\boldsymbol{v} + \frac{\rho}{2} ||\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})||_F^2$$

by computing the gradient,

$$0 = \boldsymbol{M}d\boldsymbol{v} - \Delta t\boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T\mathbf{W}^T\mathbf{y} - \rho\boldsymbol{D}^T\mathbf{W}^T[\mathbf{W}(\boldsymbol{F} - \boldsymbol{D}d\boldsymbol{v} - \boldsymbol{b})]$$
$$(\boldsymbol{M} + \rho\boldsymbol{D}^T\mathbf{W}^T\mathbf{W}\boldsymbol{D})d\boldsymbol{v} = \Delta t\boldsymbol{M}\boldsymbol{g} + \boldsymbol{D}^T\mathbf{W}^T\mathbf{y} + \rho\boldsymbol{D}^T\mathbf{W}^T[\mathbf{W}(\boldsymbol{F} - \boldsymbol{b})]$$

Notice the condition number of the "stiffness" matrix of this global solve is usually much better than the one of the standard implicit MPM.

Dual step. The problem in $\mathbf{y}$-step is non-smooth, thus we add a quadratic regularizer to the augmented Lagrangian and solve

$$\mathbf{y}^{n+1} = \underset{\mathbf{y}}{\text{argmax}} \ \text{Inf} \ \{L_\rho(\boldsymbol{F}^{n+1}, d\boldsymbol{v}^{n+1}, \mathbf{y}) - \frac{1}{2\rho}||\mathbf{y} - \mathbf{y}^n||_F^2\}$$

by computing the gradient and setting it to 0:

$$\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b}) - \frac{1}{\rho}(\mathbf{y}^{n+1} - \mathbf{y}^n) = 0$$
$$\mathbf{y}^{n+1} = \mathbf{y}^n + \rho[\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b})]$$

2.2 ADMM Residuals

We denote optimal primal variables by $\boldsymbol{F}^*$ and $d\boldsymbol{v}^*$, and the optimal dual variable by $\mathbf{y}^*$. The primal feasibility condition can be written as

$$\mathbf{W}(\boldsymbol{F}^* - \boldsymbol{D}d\boldsymbol{v}^* + \boldsymbol{b}) = 0$$

and dual feasibility conditions are

$$0 = \frac{\partial L_\rho}{\partial \boldsymbol{F}}$$
$$0 = \frac{\partial L_\rho}{\partial d\boldsymbol{v}}$$

We can define the primal residual as

$$\mathbf{r}^{n+1} = \mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b})$$

From the global step, we have

$$\begin{aligned}
0 &= \left.\frac{\partial L_\rho}{\partial d\boldsymbol{v}}\right|_{d\boldsymbol{v}^{n+1}} \\
&= \boldsymbol{M}d\boldsymbol{v}^{n+1} - \Delta t\boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T\mathbf{W}^T\mathbf{y}^n - \rho\boldsymbol{D}^T\mathbf{W}^T[\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b})] \\
&= \boldsymbol{M}d\boldsymbol{v}^{n+1} - \Delta t\boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T\mathbf{W}^T(\mathbf{y}^n + \rho[\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b})]) \\
&= \boldsymbol{M}d\boldsymbol{v}^{n+1} - \Delta t\boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T\mathbf{W}^T(\mathbf{y}^n + \rho\mathbf{r}^{n+1}) \\
&= \boldsymbol{M}d\boldsymbol{v}^{n+1} - \Delta t\boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T\mathbf{W}^T\mathbf{y}^{n+1}
\end{aligned}$$

so $d\boldsymbol{v}^{n+1}$ and $\mathbf{y}^{n+1}$ satisfy the second dual feasibility condition. Let's check the first one. From the local step, we have

$$\begin{aligned}
0 &= \left.\frac{\partial L_\rho}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^{n+1}} \\
&= V_p\boldsymbol{P}_p^{n+1} + \mathbf{W}^T\mathbf{y}^n + \rho\mathbf{W}^T[\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^n - \boldsymbol{b})] \\
&= V_p\boldsymbol{P}_p^{n+1} + \mathbf{W}^T\mathbf{y}^n + \rho\mathbf{W}^T[\mathbf{W}(\boldsymbol{F}^{n+1} - \boldsymbol{D}d\boldsymbol{v}^{n+1} - \boldsymbol{b}) + \mathbf{W}\boldsymbol{D}(d\boldsymbol{v}^{n+1} - d\boldsymbol{v}^n)] \\
&= V_p\boldsymbol{P}_p^{n+1} + \mathbf{W}^T\mathbf{y}^n + \rho\mathbf{W}^T[\mathbf{r}^{n+1} + \mathbf{W}\boldsymbol{D}(d\boldsymbol{v}^{n+1} - d\boldsymbol{v}^n)] \\
&= V_p\boldsymbol{P}_p^{n+1} + \mathbf{W}^T(\mathbf{y}^n + \rho\mathbf{r}^{n+1}) + \rho\mathbf{W}^T[\mathbf{W}\boldsymbol{D}(d\boldsymbol{v}^{n+1} - d\boldsymbol{v}^n)] \\
&= V_p\boldsymbol{P}_p^{n+1} + \mathbf{W}^T\mathbf{y}^{n+1} + \rho\mathbf{W}^T[\mathbf{W}\boldsymbol{D}(d\boldsymbol{v}^{n+1} - d\boldsymbol{v}^n)]
\end{aligned}$$

When the third term becomes zero, the first feasibility is satisfied. Thus we can define the dual residual as

$$\boldsymbol{s}^{n+1} = \rho\mathbf{W}^T[\mathbf{W}\boldsymbol{D}(d\boldsymbol{v}^{n+1} - d\boldsymbol{v}^n)]$$

2.3 Ruiz equilibration

As discussed in the main paper, several attempts for improving the ADMM convergence have been examined in our experiments. Particularly, we adopt the Ruiz equilibration [9] (originally designed for simple quadratic problems) and adapt it to our nonlinear problem.

Let's start from the Lagrangian,

$$\boldsymbol{L} = \frac{1}{2} d\boldsymbol{v}^T \boldsymbol{M} d\boldsymbol{v} + \Psi(\boldsymbol{F}) - \Delta t d\boldsymbol{v}^T \boldsymbol{M}\boldsymbol{g} + \mathbf{y} : [\mathbf{W}(\boldsymbol{F} - \boldsymbol{D} d\boldsymbol{v} - \boldsymbol{b})]$$

which is a nonlinear function of $\boldsymbol{F}$. To make it easier, we will try to derive the Ruiz equilibration from the linearized version,

$$L = \frac{1}{2} d\boldsymbol{v}^T \boldsymbol{M} d\boldsymbol{v} + \Psi(\boldsymbol{F}^n) + \left.\frac{\partial \Psi}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^n} : (\boldsymbol{F} - \boldsymbol{F}^n) - \Delta t d\boldsymbol{v}^T \boldsymbol{M}\boldsymbol{g} + \mathbf{y} : [\mathbf{W}(\boldsymbol{F} - \boldsymbol{D} d\boldsymbol{v} - \boldsymbol{b})]$$

From the optimality condition,

$$0 = V_p(\boldsymbol{P}_p^n + \left.\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^n} : (\boldsymbol{F} - \boldsymbol{F}_n)) + \mathbf{W}^T y \tag{8}$$

$$0 = \boldsymbol{M} d\boldsymbol{v} - \Delta t \boldsymbol{M}\boldsymbol{g} - \boldsymbol{D}^T \mathbf{W}^T \mathbf{y} \tag{9}$$

$$0 = \boldsymbol{F} - \boldsymbol{D} d\boldsymbol{v} - \boldsymbol{b} \tag{10}$$

We scale Eqn. (10) by $\mathbf{W}$ and write down the KKT as

$$\begin{pmatrix} \boldsymbol{Z} & & \mathbf{W}^T \\ & \boldsymbol{M} & -\boldsymbol{D}^T \mathbf{W}^T \\ \mathbf{W} & -\mathbf{W}\boldsymbol{D} & \end{pmatrix} \begin{pmatrix} \boldsymbol{F} \\ d\boldsymbol{v} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} sth \\ sth \\ sth \end{pmatrix} \tag{11}$$

where $\boldsymbol{Z}$ is a symmetric operator applied to $\boldsymbol{F}$ as

$$\boldsymbol{Z}\boldsymbol{F} = V_p(\left.\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}\right|_{\boldsymbol{F}^n} : \boldsymbol{F})$$

Now we can try to propose some scale to reformulate our problem. Denote the matrix in Eqn. (11) as

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{Z} & & \mathbf{W}^T \\ & \boldsymbol{M} & -\boldsymbol{D}^T \mathbf{W}^T \\ \mathbf{W} & -\mathbf{W}\boldsymbol{D} & \end{pmatrix}$$

and write the scaling matrix as

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{G} & & \\ & \mathbf{Q} & \\ & & \boldsymbol{R} \end{pmatrix}$$

where $\boldsymbol{G}$, $\mathbf{Q}$ and $\boldsymbol{R}$ are all diagonal matrices. Thus, we can scale the original problem using those three

diagonal matrices.

$$\bar{\boldsymbol{Z}} = \boldsymbol{G}\boldsymbol{Z}\boldsymbol{G} \tag{12}$$
$$\bar{\mathbf{W}} = \boldsymbol{R}\mathbf{W}\boldsymbol{G} \tag{13}$$
$$\bar{\boldsymbol{M}} = \mathbf{Q}\boldsymbol{M}\mathbf{Q} \tag{14}$$
$$\bar{\boldsymbol{D}} = \boldsymbol{G}^{-1}\boldsymbol{D}\mathbf{Q} \tag{15}$$
$$\bar{d\boldsymbol{v}} = \mathbf{Q}^{-1}d\boldsymbol{v} \tag{16}$$
$$\bar{\boldsymbol{F}} = \boldsymbol{G}^{-1}\boldsymbol{F} \tag{17}$$
$$\bar{\boldsymbol{b}} = \boldsymbol{G}^{-1}\boldsymbol{b} \tag{18}$$
$$\bar{\boldsymbol{g}} = \mathbf{Q}^{-1}\boldsymbol{g} \tag{19}$$
$$\bar{\mathbf{y}} = \boldsymbol{R}^{-1}\mathbf{y} \tag{20}$$
$$\tag{21}$$

[9] explained how to compute the scaling matrices by simply computing the infinite norm of each row $||K_i||_\infty$, and using $\frac{1}{\sqrt{||K_i||_\infty}}$ as the scale from an iterative process.

Algorithm 1 Ruiz equilibration

Initialization $\boldsymbol{S} = \boldsymbol{I}, \delta = 0$
While $||1 - \delta||_\infty > \epsilon_{\text{tol}}$
  For $i = 1..(N_n + 2 * N_p)$
    Do $\delta_i = \frac{1}{\sqrt{||K_i||_\infty}}$
  $\boldsymbol{S} = \mathbf{diag}(\delta)\boldsymbol{S}$
Return $\boldsymbol{S}$

The remaining difficulty is due to the non-linearity of the elasticity energy density function. Instead of solving the scaled local step, we can solve the original local problem and then scale the computed $\boldsymbol{F}$ to $\bar{\boldsymbol{F}}$ and use it in the global step and dual step.

2.4 Inelasticity

As discussed in the main paper, when the problem does not have a well-defined energy Ψ, we can instead write down the integral whose derivative corresponds to the projected force due to plasticity to replace the energy in all formulas. Moreover, all the derivations above can still be applied without any changes. The caveat here is that our local step cannot employ the standard line search method to ensure the strict decrease of the energy during the Newton steps. In practice, we never observed any problem in the local solves. We postulate that it is mostly because the non-smoothness and non-linearity within the local systems are already diminished compared to the original global non-smooth and non-linear problem [4].

3 Projection and Projection Derivatives for Plasticity

von-Mises. When the trial state is already inside the yield surface, no projection is required. Otherwise, the projection operation is defined by

$$\hat{\mathcal{Z}}(\Sigma) = e^{\epsilon - (||\text{dev}(\epsilon)|| - \frac{\sigma_y}{2\mu})\frac{\text{dev}(\epsilon)}{||\text{dev}(\epsilon)||}}$$

where σ_y is the yield stress and $\epsilon = \langle \log\Sigma \rangle$. The corresponding derivative is

$$(\frac{\partial \hat{\mathcal{Z}}}{\partial \Sigma})_{ij} = \hat{\mathcal{Z}}_i \Big(\Sigma^{-1} - \big(1 - \frac{\sigma_y}{2\mu \mathrm{dev}(\epsilon)}\big)\big(\Sigma^{-1} - \frac{\mathbf{1}\left\langle \Sigma^{-1} \right\rangle^T}{d}\big) - \frac{\sigma_y}{2\mu \mathrm{dev}(\epsilon)} \frac{\mathrm{dev}(\epsilon)}{||\mathrm{dev}(\epsilon)||} \big(\frac{\mathrm{dev}(\epsilon)}{||\mathrm{dev}(\epsilon)||}\big)^T \Sigma^{-1} \Big)_{ij}$$

Drucker-Prager. There are three different cases for Drucker-Prager. First, when the trial state is inside the yield surface, no projection is required. Second, when the trial state indicates the MPM particle is under tension, there should not exist any forces to prevent particles from free separation. Thus we project the trial state to the origin, i.e. a state of zero stress. In the last case, the projection is

$$\hat{\mathcal{Z}}(\Sigma) = e^{\epsilon - (||\mathrm{dev}(\epsilon)|| + \frac{d\lambda + 2\mu}{2\mu} \mathrm{tr}(\epsilon)\alpha) \frac{\mathrm{dev}(\epsilon)}{||\mathrm{dev}(\epsilon)||}}$$

where α is the friction coefficient. The corresponding derivative is

$$(\frac{\partial \hat{\mathcal{Z}}}{\partial \Sigma})_{ij} = \hat{\mathcal{Z}}_i \Big(\big(\frac{1+2p}{d}\mathbf{1} - \frac{p}{\mathrm{tr}(\epsilon)}\epsilon\big) \left\langle \Sigma^{-1} \right\rangle^T - p\big(\boldsymbol{I} - \frac{\mathrm{dev}(\epsilon)}{||\mathrm{dev}(\epsilon)||}\big(\frac{\mathrm{dev}(\epsilon)}{||\mathrm{dev}(\epsilon)||}\big)^T\big)\Sigma^{-1} \Big)_{ij}$$

where

$$p = \frac{\alpha(d\lambda + 2\mu)\mathrm{tr}(\epsilon)}{2\mu||\mathrm{dev}(\epsilon)||}$$

4 Pseudo-code

We now present the pseudo-code for our ADMM-based MPM inelasticity solver.

Algorithm 2 ADMM solver

procedure INITIAL_GUESS
  for all nodes i **do**
    $d\boldsymbol{v}_i \leftarrow (\sum_p \boldsymbol{A}_p w_{ip}^n)/(\sum_p w_{ip}^n)$
    $\hat{\boldsymbol{v}} \leftarrow d\boldsymbol{v}_i + \boldsymbol{v}_i^n$
    if COLLISION_DETECTED($\hat{\boldsymbol{v}}$, i) **then**
      $d\boldsymbol{v}_i \leftarrow$ BOUNDARY_PROJECT($\hat{\boldsymbol{v}}, i$) $- \boldsymbol{v}_i^n$
      RECORD_COLLISION(i)

procedure TIME_STEP
  BUILD_MATRIX
  INITIAL_GUESS
  while not reach max iterations
    LOCAL_STEP($\boldsymbol{F}_{E,p}$)
    LOCAL_STEP($\boldsymbol{F}_{N,p}$)
    GLOBAL_STEP($\delta\boldsymbol{v}$)
    DUAL_STEP
  for all particles p **do**
    $\boldsymbol{A}_p \leftarrow \sum_i d\boldsymbol{v}_i w_{ip}^n$

Algorithm 3 Local solver

function LOCAL_FUNCTION($\mathbf{z}$)
    $\mathbf{r} \leftarrow$ MATERIAL_PROJECT($\mathbf{z}$) $\mathbf{f} \leftarrow$ FIRST_PIOLA($\mathbf{r}$)
    return $V_p^0 \boldsymbol{f} + \rho \mathbf{W}_p \mathbf{W}_p(\mathbf{r} - \Sigma_p)$

function local_derivative($\mathbf{z}$)
    $\mathbf{r} \leftarrow$ material_project($\mathbf{z}$) $\mathbf{t} \leftarrow$ material_derivative($\mathbf{z}$) $\mathbf{g} \leftarrow$ first_piola_derivative($\mathbf{r}$)
    return $v_p^0 \mathbf{gt} + \rho \mathbf{w}_p \mathbf{w}_p \mathbf{1}$

function root_finding($\mathbf{z}$)
    while not reach max iterations
        $\boldsymbol{g} \leftarrow$ local_function($\mathbf{z}$)
        $\boldsymbol{p} \leftarrow$ local_derivative($\mathbf{z}$)
        step $\leftarrow \boldsymbol{p}^{-1}\boldsymbol{g}$
        if step small enough
            return z
        z += **step**
    return z

procedure local_step($\langle \boldsymbol{f}_p \rangle$)
    for all particles p **do**
        $\mathbf{c} \leftarrow \mathbf{wy}_p + \rho \mathbf{w}_p \mathbf{w}_p (\boldsymbol{f}_p^n + \sum_i \delta t (\boldsymbol{v}_i^n + d\boldsymbol{v}_i)(\nabla w_{ip}^n)^t \boldsymbol{f}_p^n)$
        $(\boldsymbol{u}_p, \boldsymbol{\sigma}_p, \boldsymbol{v}_p) \leftarrow \mathbf{svd}(\mathbf{c})$
        $\mathbf{z} \leftarrow \boldsymbol{u}_p^t \boldsymbol{f}_p \boldsymbol{v}_p$
        $\mathbf{z}_{\text{solved}} \leftarrow$ root_finding(z)
        $\boldsymbol{f}_p \leftarrow \boldsymbol{u}_p \mathbf{z}_{\text{solved}} \boldsymbol{v}^t$

Algorithm 4 Global solver

procedure BUILD_MATRIX
  $\boldsymbol{M} \leftarrow \boldsymbol{0}$
  for all particles p **do**
    for all nodes i **do**
      for all nodes j **do**
$\boldsymbol{M}(i,j)$ += $\rho\Delta t\Delta t\mathbf{W}_p\mathbf{W}_p(\nabla w_{ip}^n)^T\boldsymbol{F}_{E,p}^n(\boldsymbol{F}_{E,p}^n)^T\nabla w_{jp}^n$
        $\boldsymbol{M}(i,j)$ += $\rho\Delta t\Delta t\mathbf{W}_p\mathbf{W}_p(\nabla w_{ip}^n)^T\boldsymbol{F}_{N,p}^n(\boldsymbol{F}_{N,p}^n)^T\nabla w_{jp}^n$
procedure BUILD_RIGHT_HAND_SIDE
  $\boldsymbol{R} \leftarrow \boldsymbol{0}$
  for all nodes i **do**
    $\boldsymbol{R}_i$ += $\Delta t m_i\ \boldsymbol{g}$
  for all particles p **do**
    $\boldsymbol{B}_{E,p} \leftarrow \rho\mathbf{W}_{E,p}(\boldsymbol{F}_p^n - \sum_i \Delta t\boldsymbol{v}_i^n(\nabla w_{ip}^n)^T\boldsymbol{F}_{E,p}^n)$
    $\boldsymbol{B}_{N,p} \leftarrow \rho\mathbf{W}_{N,p}(\boldsymbol{F}_p^n - \sum_i \Delta t\boldsymbol{v}_i^n(\nabla w_{ip}^n)^T\boldsymbol{F}_{N,p}^n)$
  for all nodes i **do**
    $\boldsymbol{R}_i$ += $\mathbf{W}_{E,p}\sum_p \Delta t\boldsymbol{B}_{E,p}(\boldsymbol{F}_{E,p}^n)^T\nabla w_{ip}^n$
    $\boldsymbol{R}_i$ += $\mathbf{W}_{N,p}\sum_p \Delta t\boldsymbol{B}_{N,p}(\boldsymbol{F}_{N,p}^n)^T\nabla w_{ip}^n$

procedure GLOBAL_STEP($\langle d\boldsymbol{v}\rangle$)
  BUILD_RIGHT_HAND_SIDE
  Find $\langle d\boldsymbol{v}_{\text{solved}}\rangle$ so $\boldsymbol{M}\langle d\boldsymbol{v}_{\text{solved}}\rangle = \boldsymbol{R}$
▷ Solve using MINRES with large tolerance
  $\langle d\boldsymbol{v}\rangle \leftarrow \langle d\boldsymbol{v}_{\text{solved}}\rangle$

References

[1] Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG)*, 37(4):149, 2018.

[2] Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Transactions on Graphics (TOG)*, 36(6):223, 2017.

[3] Ming Gao, Xinlei Wang, Kui Wu, Andre Pradhana, Eftychios Sifakis, Cem Yuksel, and Chenfanfu Jiang. Gpu optimization of material point methods. In *SIGGRAPH Asia 2018 Technical Papers*, SIGGRAPH Asia '18, pages 254:1–254:12, 2018.

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

[5] B Nedjar. Frameworks for finite strain viscoelastic-plasticity based on multiplicative decompositions. part i: Continuum formulations. *Computer Methods in Applied Mechanics and Engineering*, 191(15-16):1541–1562, 2002.

[6] B Nedjar. Frameworks for finite strain viscoelastic-plasticity based on multiplicative decompositions. part ii: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 191(15-16):1563–1593, 2002.

[7] Matthew Overby, George E Brown, Jie Li, and Rahul Narain. ADMM projective dynamics: Fast simulation of hyperelastic models with dynamic constraints. *IEEE Trans. Vis. Comput. Graph.*, 23(10):2222–2234, October 2017.

[8] Stefanie Reese and Sanjay Govindjee. A theory of finite viscoelasticity and numerical aspects. *Int. J. Solids Struct.*, 35(26):3455–3482, September 1998.

[9] B Stellato, G Banjac, P Goulart, A Bemporad, and S Boyd. OSQP: An operator splitting solver for quadratic programs. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 339–339, 2018.

[10] Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun. Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph*, 34(5):160:1–160:20, 2015.

[11] Yonghao Yue, Breannan Smith, Peter Yichen Chen, Maytee Chantharayukhonthorn, Ken Kamrin, and Eitan Grinspun. Hybrid grains: Adaptive coupling of discrete and continuum simulations of granular media. In *SIGGRAPH Asia 2018 Technical Papers*, SIGGRAPH Asia '18, pages 283:1–283:19, 2018.