Notes on Various MPM Extrapolations

John A. Nairn

August 20, 2019

Contents

Chapter 1

MPM Extrapolations

1.1 The MPM Extrapolation Problem

Let lower case denote grid (or Eulerian frame) quantities and upper case denote particle (or Lagrangian frame) quantities. We define S to be a matrix of grid-based shape functions to extrapolate from nodes to any position (normally to a particle position). Thus:

$$\boldsymbol{Q}_p = \sum_i S_{pi}\boldsymbol{q}_i \tag{1.1}$$

$$1 = \sum_i S_{pi} \tag{1.2}$$

The first extrapolates any nodal quantity, q_i, to a particle quantity, Q_p, and S_{pi} is shape function for node i at position of particle p (note that many MPM papers use S_{ip}, but the reversed order makes more sense with S and transformation from grid to particles). The second requires partition of unity for the shape functions. As a linear transformation

$$\boldsymbol{Q} = \mathsf{S}\boldsymbol{q} \tag{1.3}$$

where $\boldsymbol{Q}$ is vector of particle properties, $\boldsymbol{q}$ is vector of nodal quantities, and S is the $N \times n$ transformation matrix (N particles and n nodes).

1.1.1 Extrapolate to the Grid — Lumped and Full Mass Matrix

Extrapolating properties like velocity or acceleration back to the grid requires, inverse of S. Becase S is not square, we need to use a pseudo inverse denoted here by S^+. The standard S^+ was found in original MPM paper by Sulsky using weighted least squares. Assuming particle velocities extrapolated from grid velocities are given by $\boldsymbol{V}^{(n)}_{g\to p} = \sum_i S_{pi}\boldsymbol{v}^{(n)}_i$, find the $\boldsymbol{v}^{(n)}_i$ to minimize the difference of this velocity with

initial particle velocity ($\boldsymbol{V}_p^{(n)}$) weighted by the particle mass:

$$\Omega = \sum_p M_p\left(\boldsymbol{V}_p^{(n)} - \boldsymbol{V}_{g\to p}^{(n)}\right)^2 = \sum_p M_p\left(\boldsymbol{V}_p^{(n)} - \sum_i S_{pi}\boldsymbol{v}_i^{(n)}\right)^2 \tag{1.4}$$

$$0 = \frac{d\Omega}{d\boldsymbol{v}_j^{(n)}} = -2\left(\sum_p S_{pj}M_p\boldsymbol{V}_p^{(n)} - \sum_i\sum_p M_p S_{pj}S_{pi}\boldsymbol{v}_i^{(n)}\right) \tag{1.5}$$

$$\sum_i m_{ji}^{(n)}\boldsymbol{v}_i^{(n)} = \sum_p S_{jp}^T P_p^{(n)} \tag{1.6}$$

$$\tilde{\mathsf{m}}\boldsymbol{v} = \mathsf{S}^T\boldsymbol{P} \tag{1.7}$$

$$\boldsymbol{v} = \tilde{\mathsf{m}}^{-1}\mathsf{S}^T\boldsymbol{P} \tag{1.8}$$

where $\tilde{m}_{ji}^{(n)} = \sum_p M_p S_{jp}^T S_{pi}$ is an element of the full (and symmetric) $n \times n$ mass matrix $\tilde{\mathsf{m}}$ and $\boldsymbol{P}$ is a vector of particle momenta. In standard MPM, $\tilde{\mathsf{m}}$ is replaced m, which is a diagonal lumped mass matrix, with i^{th} diagonal element defined by:

$$\boldsymbol{m}_i^{(n)} = \sum_j \tilde{m}_{ji}^{(n)} = \sum_p S_{pi}M_p = \sum_p S_{ip}^T M_p \quad \text{or} \quad \boldsymbol{m} = \mathsf{S}^T\boldsymbol{M} \quad \text{and} \quad \mathsf{m} = \text{diag}(\boldsymbol{m}) \tag{1.9}$$

where $\boldsymbol{m}$ is vector of nodal masses (distinguished from m matrix be serifs) and $\boldsymbol{M}$ is vector of particle masses. Writing particle momenta as $\boldsymbol{P} = \mathsf{M}\boldsymbol{V}$ where $\mathsf{M} = \text{diag}(\boldsymbol{M})$, the pseudo inverse matrix for a lumped mass matrix is:

$$\mathsf{S}^+ = \mathsf{m}^{-1}\mathsf{S}^T\mathsf{M} \quad \Longrightarrow \quad S_{ip}^+ = \sum_k\sum_l \frac{\delta_{ik}}{\boldsymbol{m}_i^{(n)}} S_{kl}^T\delta_{lp}M_p = \frac{S_{pi}M_p}{\boldsymbol{m}_i^{(n)}} \tag{1.10}$$

Chapter 2

General MPM Particle Update

2.1 Velocity Extrapolations

The first task in MPM time step n is to extrapolate from the particles to the grid. We begin by assuming a linear transformation

$$\boldsymbol{v}_i^{(n)} = \sum_p S_{ip}^+ \boldsymbol{V}_p^{(n)} \qquad \text{or} \qquad \boldsymbol{v} = \mathsf{S}^+ \boldsymbol{V} \tag{2.1}$$

where $\boldsymbol{v}$ and $\boldsymbol{V}$ are vectors of nodal and particle velocities, and S^+ is matrix of shape functions that ideally would be the pseudo-inverse of S. After extrapolating velocities to the grid MPM time step extrapolates particle stresses to the grid to find nodal forces, $\boldsymbol{f}_i^{(n)}$. The velocities and forces are used to update grid nodal positions and velocities using:

$$\tilde{\boldsymbol{x}}_i^{(n+1)} = \boldsymbol{x}_i + \boldsymbol{v}_i^{(n)} \Delta t + \frac{1}{2} \boldsymbol{a}_i^{(n)} (\Delta t)^2 \tag{2.2}$$

$$\tilde{\boldsymbol{v}}_i^{(n+1)} = \boldsymbol{v}_i^{(n)} + \boldsymbol{a}_i^{(n)} \Delta t \tag{2.3}$$

The $\tilde{(\cdot)}$ nomenclature indicate grid position and velocity at the end of time step n, but it is not the grid position and velocity for time step $n+1$. Instead, for time step $n+1$, the grid position is reset to $\boldsymbol{x}_i$ and the velocity is extrapolated to the new grid using updated particle positions and velocities. The $\boldsymbol{a}_i^{(n)}$ is a grid acceleration given by $\boldsymbol{a}_i^{(n)} = \boldsymbol{f}_i^{(n)} / m_i^{(n)}$ where $m_i^{(n)}$ is nodal mass in time step n.

Next, the updated grid values are used to update particle positions and velocity, which can be written generally as

$$\boldsymbol{X}_p^{(n+1)} = \boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n)} \Delta t + \frac{1}{2} \mathbb{A}_p^{(n)} (\Delta t)^2 = \boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n+1)} \Delta t - \frac{1}{2} \mathbb{A}_p^{(n)} (\Delta t)^2 \tag{2.4}$$

$$\boldsymbol{V}_p^{(n+1)} = \boldsymbol{V}_p^{(n)} + \mathbb{A}_p^{(n)} \Delta t \tag{2.5}$$

where $\mathbb{V}_p^{(n)}$ and $\mathbb{A}_p^{(n)}$ are velocity and acceleration extrapolated from the grid to the particle, by methods that are not yet determined or methods that can be potentially be selected to define various types of MPM. Whatever justification is used to select them, however, they are subject the physical consistency restriction that:

$$\mathbb{V}_p^{(n+1)} = \mathbb{V}_p^{(n)} + \mathbb{A}_p^{(n)} \Delta t \tag{2.6}$$

The above allows one to find particle updates for any definition of effective velocity and acceleration on the particles. Alternatively, if someone proposes particle updates, those can be reverse-engineered

into the effective terms that were assumed:

$$\mathbb{A}_p^{(n)} = \frac{\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}}{\Delta t} \tag{2.7}$$

$$\mathbb{V}_p^{(n)} = \frac{\boldsymbol{X}_p^{(n+1)} - \boldsymbol{X}_p^{(n)}}{\Delta t} - \frac{1}{2}\left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) \tag{2.8}$$

$$\mathbb{V}_p^{(n+1)} = \frac{\boldsymbol{X}_p^{(n+1)} - \boldsymbol{X}_p^{(n)}}{\Delta t} + \frac{1}{2}\left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) \tag{2.9}$$

If needed, the following identities always hold:

$$\begin{aligned} \mathbb{A}_p^{(n)}\Delta t &= \mathbb{V}_p^{(n+1)} - \mathbb{V}_p^{(n)} = \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)} \\ \mathbb{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n+1)} &= \mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)} \\ \left(\mathbb{A}_p^{(n)} - \boldsymbol{A}_{g\to p}^{(n)}\right)\Delta t &= \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n+1)} + \boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_p^{(n)} \end{aligned} \tag{2.10}$$

2.2 Matrix Form

Let upper case vectors be particle properties and lower case grid properties. Let $\mathbb{V}^{(n)}$ and $\mathbb{A}^{(n)}$ be vectors of velocity and acceleration extrapolated to the particle. The task is to find $\mathbb{V}^{(n)}$, $\mathbb{V}^{(n+1)}$ and $\mathbb{A}^{(n)}$ and use them for particle position and velocity updates. Next start with a generalized MPM update, which be used to reverse engineer effective particle velocity and accelerations, as:

$$\boldsymbol{V}^{(n+1)} = \mathsf{P}\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \tag{2.11}$$

where P is some projection tensor. In standard FLIP MPM, $\mathsf{P} = \mathsf{I}$, but ideally P would be a filter that removes null space noise from the particle velocities or $(\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)}$ would be the null space velocities. Reverse engineering particle acceleration gives

$$\mathbb{A}^{(n)}\Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)} = \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} \tag{2.12}$$

Note that a second interpretation of P is as damping of the extrapolated acceleration or

$$\mathbb{A}^{(n)}\Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)} = \left(\mathsf{S}\boldsymbol{a} - \alpha(\mathsf{P})\boldsymbol{V}^{(n)}\right)\Delta t \tag{2.13}$$

where $\alpha(\mathsf{P}) = \frac{1}{\Delta t}(\mathsf{I}-\mathsf{P})$ is a damping based on current particle velocities. It is a first order term (because of Δt in denominator). The acceleration term in position update is needed to capture this term.

2.3 General Results in Terms of $\mathbb{V}^{(n)}$

If we treat $\mathbb{V}^{(n)}$ as only undetermined quantity and add grid and particle damping to the acceleration, the update equations can be written in terms of eventual assunmption for $\mathbb{V}^{(n)}$ as:

$$\mathbb{A}^{(n)}\Delta t = \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} - \alpha_g\Delta t\mathsf{S}\boldsymbol{v} - \alpha_p\Delta t\boldsymbol{V}^{(n)} \tag{2.14}$$

$$\mathbb{V}^{(n+1)} = \mathbb{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} - \alpha_g\Delta t\mathsf{S}\boldsymbol{v} - \alpha_p\Delta t\boldsymbol{V}^{(n)} \tag{2.15}$$

$$\boldsymbol{V}^{(n+1)} = \mathsf{P}\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t - \alpha_g\Delta t\mathsf{S}\boldsymbol{v} - \alpha_p\Delta t\boldsymbol{V}^{(n)} \tag{2.16}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left[\mathbb{V}^{(n)} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t - \frac{1}{2}(\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} - \frac{1}{2}\alpha_g\Delta t\mathsf{S}\boldsymbol{v} - \frac{1}{2}\alpha_p\Delta t\boldsymbol{V}^{(n)}\right]\Delta t \tag{2.17}$$

Introducing $\boldsymbol{A}^{(0)}_{damp} = \alpha_g \mathsf{S}\boldsymbol{v} + \alpha_p \boldsymbol{V}^{(n)}$, the general updates become

$$\begin{aligned}
\mathbb{A}^{(n)}\Delta t &= -(\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.18)\\
\mathbb{V}^{(n+1)} &= \mathbb{V}^{(n)} - (\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.19)\\
\boldsymbol{V}^{(n+1)} &= \mathsf{P}\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.20)\\
\boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \left[\mathbb{V}^{(n)} - \frac{1}{2}\boldsymbol{V}^{(n)} + \frac{1}{2}\left(\mathsf{P}\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t\right)\right]\Delta t && (2.21)\\
&= \boldsymbol{X}^{(n)} + \left[\mathbb{V}^{(n)} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right]\Delta t && (2.22)
\end{aligned}$$

Errors between Lagrangian position change and velocities are:

$$\begin{aligned}
\frac{\boldsymbol{X}^{(n+1)} - \boldsymbol{X}^{(n)}}{\Delta t} &= \mathbb{V}^{(n)} - \frac{1}{2}(\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)} + \frac{1}{2}\left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.23)\\
\frac{\boldsymbol{V}^{(n)} + \boldsymbol{V}^{(n+1)}}{2} &= \frac{1}{2}(\mathsf{I}+\mathsf{P})\,\boldsymbol{V}^{(n)} + \frac{1}{2}\left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.24)\\
(\text{error}) &= \mathbb{V}^{(n)} - \boldsymbol{V}^{(n)} && (2.25)
\end{aligned}$$

Once the projection operator is chosen, the above update determines the particle velocity update, but the position update depends on what is assumed for the effective velocity. The following sections give possible methods. Lastly, some result suggested second order terms, which appear only in position update, can cause instabilities (radial compression in axisymmetry). Truncating second terms might help, but the choice of how to truncate is not clear. The truncation can also be implement by altering the choice of $\mathbb{V}^{(n)}$, which might be the best tactic.

2.3.1 Assume $\mathbb{V}^{(n)} = \boldsymbol{V}^{(n)}$

A tempting choice is to select $\mathbb{V}^{(n)} = \boldsymbol{V}^{(n)}$, because that choice eliminates the error between particle position change and its average velocity. The updates become:

$$\begin{aligned}
\boldsymbol{V}^{(n+1)} &= \mathsf{P}\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.26)\\
\boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \left[\frac{\boldsymbol{V}^{(n)} + \boldsymbol{V}^{(n+1)}}{2}\right]\Delta t && (2.27)
\end{aligned}$$

Whenever tried, this method gives poor results. The issue appears to be that particle velocities develop noise and that noise causes this position update to be very poor.

2.3.2 Assume $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$

The problem with $\mathbb{V}^{(n)} = \boldsymbol{V}^{(n)}$ was recognized in early days of FLIP modeling. The correction suggested by Brackbill (learned from after dinner talk at MPM Workshop in Oak Ridge in 2018) was essentially to use $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$. The updates become:

$$\begin{aligned}
\boldsymbol{V}^{(n+1)} &= \mathsf{P}\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t && (2.28)\\
\boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \left[\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right]\Delta t && (2.29)
\end{aligned}$$

Using $\tilde{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{a}\Delta t$ leads to

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left[\mathsf{S}\tilde{\boldsymbol{v}} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2} - \mathsf{S}\boldsymbol{a}\Delta t \right] \Delta t \tag{2.30}$$

The error in Lagrangian velocities (with sign change) is now

$$(\text{error}) = \boldsymbol{V}^{(n)} - \mathbb{V}^{(n)} = \boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v} = (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} \tag{2.31}$$

This error suggest a path to improve FLIP is to damp the zeroth-order error by using

$$\mathbb{A}^{(n)} = \mathsf{S}\boldsymbol{a} - \alpha(\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} \tag{2.32}$$

This approach is essentially the PIC method (if $\alpha = 1$ and $\mathsf{P} = \mathsf{S}\mathsf{S}^{+}$). In other words, PIC is a method that damps out error in Lagrangian velocity.

A common error by those using either FLIP (with $\mathsf{P} = \mathsf{I}$) or PIC (with $\mathsf{P} = \mathsf{S}\mathsf{S}^{+}$) is to think the velocity update changes but the position is the same. Because the $\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}$ depends on P, however, the position update for PIC differs from the position update for FLIP.

First Order

Some simulations (or maybe only radial compression in axisymmetry) are better by shifting the effective velocity to middle of time step and removing damping or by choosing $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} + (1/2)\mathsf{S}\boldsymbol{a}$, which is grid velocity at midpoint of time step without damping. The resulting position update is:

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left[\mathsf{S}\tilde{\boldsymbol{v}} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2} - \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t \right] \Delta t \tag{2.33}$$

$$= \boldsymbol{X}^{(n)} + \left[\mathsf{S}\tilde{\boldsymbol{v}} - \frac{1}{2}\boldsymbol{V}^{(n)} + \frac{1}{2}\mathsf{P}\boldsymbol{V}^{(n)} - \frac{1}{2}\boldsymbol{A}^{(0)}_{damp} \right] \Delta t \tag{2.34}$$

2.3.3 Assume $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)}$

An alternative view of Brackbill's suggestion is that it not revealing proper FLIP but rather recognizing that FLIP is inherently unstable, but can be fixed by using a smoothed velocity in the position update. If a good projection P is found, perhaps the best definition of MPM follows from $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)}$ or the particle velocity with null space removed (if P is non-null-space projector). The particle updates become:

$$\boldsymbol{V}^{(n+1)} = \mathsf{P}\boldsymbol{V}^{(n)} + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp} \right) \Delta t \tag{2.35}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left[\frac{3\mathsf{P}\boldsymbol{V}^{(n)} - \boldsymbol{V}^{(n)}}{2} + \frac{1}{2}\left(\left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp} \right) \Delta t \right) \right] \Delta t \tag{2.36}$$

$$= \boldsymbol{X}^{(n)} + \left[\mathsf{P}\boldsymbol{V}^{(n)} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2} \right] \Delta t \tag{2.37}$$

The errors between Lagrangian position update and average Lagrangian velocity are:

$$(\text{error}) = (\mathsf{I} - \mathsf{P})\boldsymbol{V}^{(n)} \tag{2.38}$$

or error proportional to null space terms. Note that this error is likely smaller then error when $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$. Note that for FLIP, $\mathsf{P} = \mathsf{I}$ and this approach reduces to approach in section 2.3.1 while for PIC, $\mathsf{P}\boldsymbol{V}^{(n)} = \mathsf{S}\boldsymbol{v}$ and this approach reduces to approach in section 2.3.2.

First Order

This method too might benefit from shifting to midpoint of the time step, which changes only the position update, or:

$$\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t \tag{2.39}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left[\mathsf{P}\boldsymbol{V}^{(n)} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t\right]\Delta t \tag{2.40}$$

2.3.4 Energy Dissipation

The energy change on one particle between state n and $n+1$ is effective force times distance or:

$$\Delta E_p = \frac{M_p \Delta \boldsymbol{V}_p}{\Delta t} \cdot \Delta \boldsymbol{X}_p = M_p \mathbb{A}_p^{(n)} \Delta t \cdot \left(\mathbb{V}_p^{(n)} + \frac{1}{2}\mathbb{A}_p^{(n)}\Delta t\right) \tag{2.41}$$

For a non-dissipative, FLIP update, the energy change is

$$\Delta E_{p,FLIP} = M_p \boldsymbol{A}_{g\to p}\Delta t \cdot \left(\boldsymbol{V}_{g\to p} + \frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t\right) \tag{2.42}$$

where $\boldsymbol{A}_{g\to p}$ and $\boldsymbol{V}_{g\to p}$ are the p^{th} elements of $\mathsf{S}\boldsymbol{a}$ and $\mathsf{S}\boldsymbol{v}$, respectively. The difference $\Delta E_{p,FLIP} - \Delta E_p$ is energy dissipated (positive for loss of energy) by use of an effective velocity or

$$\Delta E_{p,disp} = M_p \Delta t \left(\boldsymbol{A}_{g\to p}\cdot\boldsymbol{V}_{g\to p} - \mathbb{A}_p^{(n)}\cdot\mathbb{V}_p^{(n)} + \frac{\Delta t}{2}\left(|\boldsymbol{A}_{g\to p}|^2 - |\mathbb{A}_p^{(n)}|^2\right)\right) \tag{2.43}$$

For the usual case of $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$, a simpler (and more general) derivation is

$$\frac{\Delta E_{p,disp}}{M_p} = \boldsymbol{A}_{g\to p}\Delta t \cdot \left(\boldsymbol{V}_{g\to p} + \frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t\right) - \Delta \boldsymbol{V}_p \cdot \left(\boldsymbol{V}_{g\to p} + \frac{\Delta \boldsymbol{V}_p}{2}\right) \tag{2.44}$$

$$= (\boldsymbol{A}_{g\to p}\Delta t - \Delta \boldsymbol{V}_p)\cdot\boldsymbol{V}_{g\to p} + \frac{1}{2}(\boldsymbol{A}_{g\to p}\Delta t - \Delta \boldsymbol{V}_p)\cdot(\boldsymbol{A}_{g\to p}\Delta t + \Delta \boldsymbol{V}_p) \tag{2.45}$$

$$= (\boldsymbol{A}_{g\to p}\Delta t - \Delta \boldsymbol{V}_p)\cdot\left(\boldsymbol{V}_{g\to p} + \frac{1}{2}(\boldsymbol{A}_{g\to p}\Delta t + \Delta \boldsymbol{V}_p)\right) \tag{2.46}$$

$$= (\boldsymbol{A}_{g\to p}\Delta t - \Delta \boldsymbol{V}_p)\cdot\left(\frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t + \frac{\Delta \boldsymbol{X}_p}{\Delta t}\right) \tag{2.47}$$

2.4 Current MPM Methods

2.4.1 FLIP Particle Updates

FLIP is recovered by setting $\mathsf{P} = \mathsf{I}$. The particle velocity update is always

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}_{damp}^{(0)})\Delta t \tag{2.48}$$

The position update can take two forms

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(\frac{\boldsymbol{V}^{(n)} + \boldsymbol{V}^{(n+1)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \boldsymbol{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{2.49}$$

The first version works well. The second one differs from standard FLIP by $\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}$ in the position update. This scheme does not work well.

For the usually case of $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$ and FLIP result of $\boldsymbol{A}_{g\to p} - \mathbb{A}_p^{(n)} = \boldsymbol{A}_{p,damp}^{(0)}$, the energy dissipation is

$$\Delta E_{p,disp} = M_p \boldsymbol{A}_{p,damp}^{(0)} \Delta t \cdot \left(\tilde{\boldsymbol{V}}_{g\to p} - \frac{1}{2} \boldsymbol{A}_{p,damp}^{(0)} \Delta t \right) \tag{2.50}$$

A. Modified FLIP Particle Updates #1

Many codes use an apparently first order position update based on updated grid velocity extrapolated to the particle and combine that with a standard FLIP velocity update, which for undamped case is:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t & (2.51)\\ \boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \mathsf{S}\tilde{\boldsymbol{v}}\Delta t = \boldsymbol{X}^{(n)} + (\mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}\Delta t)\Delta t & (2.52) \end{aligned}$$

We can reverse engineer the effective particle terms as

$$\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t, \qquad \mathbb{V}^{(n+1)} = \mathsf{S}\boldsymbol{v} + \frac{3}{2}\mathsf{S}\boldsymbol{a}\Delta t, \qquad \text{and} \qquad \mathbb{A}^{(n)} = \mathsf{S}\boldsymbol{a} \tag{2.53}$$

This approach is like section 2.4.1 except velocity is taken from end of time step rather then middle (position update adds full acceleration rather than half the acceleration). This scheme works reasonably well in many codes and is same as first order option suggested in section 2.3.2. It is ignoring second order terms, but some problems (*e.g.*, radial compression of disk) appear more stable with second order terms ignored. This shifting of $\mathbb{V}^{(n)}$ by half the acceleration might be a option in other methods too.

B. Modified FLIP Particle Updates #2

Some MPM papers (at least one by Berzins) adopt the strategy of using the updated particle velocity in a first order update of the particle position. The (undamped) updates in this approach are:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t & (2.54)\\ \boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \boldsymbol{V}^{(n+1)}\Delta t = \boldsymbol{X}^{(n)} + \left[\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t\right]\Delta t & (2.55) \end{aligned}$$

This approach looks like an end-point approximation to second order position update while FLIP with $\mathbb{V}^{(n)} = \boldsymbol{V}^{(n)}$ is a midpoint, second-order position update in this same scheme. We can reverse engineer the effective particle terms as

$$\mathbb{V}^{(n)} = \boldsymbol{V}^{(n)} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t, \quad \mathbb{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \frac{3}{2}\mathsf{S}\boldsymbol{a}\Delta t, \quad \text{and} \quad \mathbb{A}^{(n)} = \mathsf{S}\boldsymbol{a} \text{ (or } \mathsf{P} = \mathsf{I}) \tag{2.56}$$

This approach is like section 2.3.1 except velocity is taken from end of time step rather then middle (for the position update). Neither of these approaches work well.

2.4.2 PIC Particle Updates

PIC is recovered by setting $\mathsf{P} = \mathsf{S}\mathsf{S}^{+}$. The particle velocity update is always

$$\boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v} + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}_{damp}^{(0)})\Delta t = \mathsf{S}\tilde{\boldsymbol{v}} - \boldsymbol{A}_{damp}^{(0)}\Delta t \tag{2.57}$$

The position update looks similar to above, but combines options for $\mathbb{V}^{(n)}$ differently:

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} = \mathsf{P}\boldsymbol{V}^{(n)} \\ \left(\frac{\boldsymbol{V}^{(n)} + \boldsymbol{V}^{(n+1)}}{2}\right) & \mathbb{V}^{(n)} = \boldsymbol{V}^{(n)} \end{cases} \tag{2.58}$$

Also, the position update differ from FLIP because the update velocity used in position update differs.

For the usually case of $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$ and PIC result of:

$$(\boldsymbol{A}_{g\to p} - \mathbb{A}_p^{(n)})\Delta t = \boldsymbol{V}^{(n)} - \boldsymbol{V}_{g\to p} + \boldsymbol{A}_{p,damp}^{(0)}\Delta t \tag{2.59}$$

the energy dissipation is

$$\Delta E_{p,disp} = M_p(\boldsymbol{V}^{(n)} - \boldsymbol{V}_{g\to p} + \boldsymbol{A}_{p,damp}^{(0)}\Delta t)\cdot\left(\tilde{\boldsymbol{V}}_{g\to p} - \frac{\boldsymbol{V}^{(n)} - \boldsymbol{V}_{g\to p}}{2} - \frac{1}{2}\boldsymbol{A}_{p,damp}^{(0)}\Delta t)\right) \tag{2.60}$$

In OSParticulas, crack planes and surfaces update using PIC methods (because velocity is PIC too) and it does not have damping (maybe it should). Using $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$, the crack velocity and particle updates are:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}\Delta t = \mathsf{S}\tilde{\boldsymbol{v}} && (2.61)\\ \boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \left(\frac{3}{2}\mathsf{S}\boldsymbol{v} + \frac{\mathsf{S}\boldsymbol{a}\Delta t - \boldsymbol{V}^{(n)}}{2}\right)\Delta t && (2.62)\\ &= \boldsymbol{X}^{(n)} + \left(\frac{3}{2}\mathsf{S}\tilde{\boldsymbol{v}} - \mathsf{S}\boldsymbol{a}\Delta t - \frac{\boldsymbol{V}^{(n)}}{2}\right)\Delta t && (2.63) \end{aligned}$$

A. Modified PIC Particle Updates #1

Some papers (at least first from Disney), modified the velocity update, but not the common, first-order position update based on extrapolated grid velocities. This style particle update for PIC (with no damping) becomes

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \mathsf{S}\tilde{v} && (2.64)\\ \boldsymbol{X}^{(n+1)} &= \boldsymbol{X}^{(n)} + \mathsf{S}\tilde{\boldsymbol{v}}\Delta t = \boldsymbol{X}^{(n)} + (\mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}\Delta t)\Delta t && (2.65) \end{aligned}$$

We can reverse engineer

$$\mathbb{V}^{(n)} = \frac{1}{2}\big(\mathsf{S}\tilde{\boldsymbol{v}} + \boldsymbol{V}^{(n)}\big), \quad \mathbb{V}^{(n+1)} = \frac{1}{2}\big(3\mathsf{S}\tilde{\boldsymbol{v}} - \boldsymbol{V}^{(n)}\big), \quad \text{and} \quad \mathbb{A}^{(n)} = \frac{1}{\Delta t}\big(\mathsf{S}\tilde{\boldsymbol{v}} - \boldsymbol{V}^{(n)}\big) \tag{2.66}$$

In terms of initial grid velocity and grid acceleration, this approach gives

$$\begin{aligned} \mathbb{V}^{(n)} &= \boldsymbol{V}^{(n)} + \frac{1}{2}\left(\mathsf{S}\boldsymbol{a} - \frac{\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}}{\Delta t}\right)\Delta t = \mathsf{S}\boldsymbol{v} + \frac{1}{2}\left(\mathsf{S}\boldsymbol{a} + \frac{\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}}{\Delta t}\right)\Delta t && (2.67)\\ \mathbb{V}^{(n+1)} &= \boldsymbol{V}^{(n)} + \frac{3}{2}\left(\mathsf{S}\boldsymbol{a} - \frac{\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}}{\Delta t}\right)\Delta t = \mathsf{S}\boldsymbol{v} + \frac{1}{2}\left(3\mathsf{S}\boldsymbol{a} - \frac{\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}}{\Delta t}\right)\Delta t && (2.68)\\ \mathbb{A}^{(n)} &= \mathsf{S}\boldsymbol{a} - \frac{\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}}{\Delta t} && (2.69) \end{aligned}$$

These results can be compared to PIC method used in `OSParticulas` in the previous section. They do not work well.

Extending the position update to second order gives

$$\boldsymbol{V}^{(n+1)} = \mathsf{S}\tilde{v} \tag{2.70}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \mathsf{S}\boldsymbol{v}\Delta t + \frac{1}{2}\mathsf{S}\boldsymbol{a}(\Delta t)^2 \tag{2.71}$$

We can reverse engineer

$$\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} + \frac{1}{2}\mathsf{S}\boldsymbol{a}\Delta t - \frac{1}{2}\mathsf{S}\tilde{v} + \frac{1}{2}\boldsymbol{V}^{(n+1)} = \frac{\mathsf{S}\boldsymbol{v} + \boldsymbol{V}^{(n)}}{2} \tag{2.72}$$

2.4.3 Combined FLIP/P Particle Updates

A combined fraction β of FLIP (using $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$) and $(1-\beta)$ of method with P (using either $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$ or $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)}$) can be expressed as follows

$$\boldsymbol{V}^{(n+1)} = \beta\boldsymbol{V}^{(n)} + (1-\beta)\mathsf{P}\boldsymbol{V}^{(n)} + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp})\Delta t \tag{2.73}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(\beta\mathsf{S}\boldsymbol{v} + (1-\beta)\mathsf{P}\boldsymbol{V}^{(n)} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{2.74}$$

Note that this velocity update can be re-expressed as FLIP method with damping term that depends on P by replacing $\boldsymbol{A}^{(0)}_{damp}$ with a new effective damping term called $\boldsymbol{A}_{damp}$:

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}_{damp})\Delta t \tag{2.75}$$

$$\boldsymbol{A}_{damp} = (1-\beta)\frac{(\mathsf{I}-\mathsf{P})\boldsymbol{V}^{(n)}}{\Delta t} + \boldsymbol{A}^{(0)}_{damp} \tag{2.76}$$

$$\boldsymbol{A}^{(0)}_{damp} = \alpha_g\mathsf{S}\boldsymbol{v} + \alpha_p\boldsymbol{V}^{(n)} \tag{2.77}$$

$$\tag{2.78}$$

Note the for mixed FLIP/PIC and for $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$, the above general mixing simplifies down to methods once used in OSParticulas:

$$\boldsymbol{A}_{damp} = (1-\beta)\frac{\boldsymbol{V}^{(n)} - \mathsf{S}v}{\Delta t} + \alpha_g\mathsf{S}\boldsymbol{v} + \alpha_p\boldsymbol{V}^{(n)} \tag{2.79}$$

$$= \alpha^{(tot)}_g\mathsf{S}\boldsymbol{v} + \alpha^{(tot)}_p\boldsymbol{V}^{(n)} \tag{2.80}$$

$$\alpha^{(tot)}_g = \alpha_g - \frac{(1-\beta)}{\Delta t} \quad \text{and} \quad \alpha^{(tot)}_p = \alpha_p + \frac{(1-\beta)}{\Delta t} \tag{2.81}$$

Using the "tot" form and updated grid velocity gives:

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \left(\left(1 + \alpha^{(tot)}_g\Delta t\right)\mathsf{S}\boldsymbol{a} - \alpha^{(tot)}_g\mathsf{S}\tilde{\boldsymbol{v}} - \alpha^{(tot)}_p\boldsymbol{V}^{(n)}\right)\Delta t \tag{2.82}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \mathsf{S}\tilde{\boldsymbol{v}}\Delta t - \left(\left(1 - \alpha^{(tot)}_g\Delta t\right)\mathsf{S}\boldsymbol{a} + \alpha^{(tot)}_g\mathsf{S}\tilde{\boldsymbol{v}} + \alpha^{(tot)}_p\boldsymbol{V}^{(n)}\right)\frac{(\Delta t)^2}{2} \tag{2.83}$$

Reverting to direct terms gives

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} - (1-\beta)\left(\boldsymbol{V}^{(n)} - \mathsf{S}v\right) + \left(\mathsf{S}\boldsymbol{a} - \alpha_g\mathsf{S}\boldsymbol{v} - \alpha_p\boldsymbol{V}^{(n)}\right)\Delta t \tag{2.84}$$

$$= \beta\boldsymbol{V}^{(n)} + (1-\beta)\mathsf{S}v + \left(\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp}\right)\Delta t \tag{2.85}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left(\mathsf{S}\boldsymbol{v} - \frac{(1-\beta)\left(\boldsymbol{V}^{(n)} - \mathsf{S}v\right)}{2}\right)\Delta t + \left(\mathsf{S}\boldsymbol{a} - \alpha_g\mathsf{S}\boldsymbol{v} - \alpha_p\boldsymbol{V}^{(n)}\right)\frac{(\Delta t^2)}{2} \tag{2.86}$$

Chapter 3

XPIC(m) Extrapolations

3.1 XPIC or an Extended PIC Method

A standard FLIP update (without damping) can be written:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t = \mathsf{S}\tilde{\boldsymbol{v}} + \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t - \mathsf{S}\tilde{\boldsymbol{v}} && (3.1)\\ &= \mathsf{S}\tilde{\boldsymbol{v}} + \boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v} = \mathsf{S}\tilde{\boldsymbol{v}} + \boldsymbol{V}^{(n)} - \mathsf{S}\mathsf{S}^{+}\boldsymbol{V}^{(n)} && (3.2)\\ &= \mathsf{S}\tilde{\boldsymbol{v}} + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} && (3.3) \end{aligned}$$

Chad's initial idea was to replace $\boldsymbol{V}^{(n)}$ by the initial PIC velocity of $\mathsf{S}\boldsymbol{v} = \mathsf{S}\mathsf{S}^{+}\boldsymbol{V}^{(n)}$, leading to $\mathsf{P} = (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\mathsf{S}\mathsf{S}^{+}$. The acceleration on the particle becomes

$$\begin{aligned} \mathbb{A}^{(n)}\Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)} &= \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} && (3.4)\\ &= \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{2}\boldsymbol{V}^{(n)} && (3.5) \end{aligned}$$

Now substitute the first $\mathsf{P}\boldsymbol{V}^{(n)}$ for $\boldsymbol{V}^{(n)}$ and the new projector is $\mathsf{P} = (3\mathsf{I} - 3\mathsf{S}\mathsf{S}^{+} + (\mathsf{S}\mathsf{S}^{+})^{2})\mathsf{S}\mathsf{S}^{+}$. The acceleration on the particle becomes:

$$\begin{aligned} \mathbb{A}^{(n)}\Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)} &= \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - (3\mathsf{I} - 3\mathsf{S}\mathsf{S}^{+} + (\mathsf{S}\mathsf{S}^{+})^{2})\mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} && (3.6)\\ &= \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{3}\boldsymbol{V}^{(n)} && (3.7) \end{aligned}$$

Repeating back substitution will continue to get higher powers in the $(\mathsf{I} - \mathsf{S}\mathsf{S}^{+})$ term where traditional PIC had power of 1. We are lead to propose generalized, extended PIC update using

$$\mathbb{A}^{(n)}\Delta t = \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m}\boldsymbol{V}^{(n)} \tag{3.8}$$

where $m = 1$ is PIC, $m = 2$ is Chad's idea and perhaps $m \to \infty$ converges optimized result. Later work showed that high m converges to a projector that removes null-space modes from the velocity. Even further work showed that this projector converges to using a full mass matrix.

3.1.1 Null-space Analogy

Current FLIP and PIC updates have two versions. FLIP can be written as:

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \qquad \text{and} \qquad \boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v} + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \tag{3.9}$$

while the two corresponding forms for PIC are

$$\boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}\Delta t \qquad \text{and} \qquad \boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \tag{3.10}$$

The second form for XPIC is

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m}\boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \tag{3.11}$$

The first are usual forms. The second have a null space term. If S^{+} was the actual Moore-Penrose pseudoinverse, then the term $(\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\boldsymbol{V}^{(n)}$ would be the orthogonal projection onto null space of S. The justification of XPIC(m) is that $(\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m}\boldsymbol{V}^{(n)}$ is approximate projection of the current particle velocities onto the null space of S and approaches exact answer as m increases. In terms of projection operator, we write

$$\mathsf{P} = \mathsf{I} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m} \tag{3.12}$$

3.2 XPIC Implementation

From the first chapter, all we need is to evaulate $\mathsf{P}\boldsymbol{V}^{(n)}$ where $\mathsf{P} = \mathsf{I} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m}$:

The particle acceleration for XPIC with optional damping is

$$\begin{aligned}
\mathsf{P}\boldsymbol{V}^{(n)} &= \boldsymbol{V}^{(n)} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^{m}\boldsymbol{V}^{(n)} && (3.13)\\
&= m\mathsf{S}\boldsymbol{v} - \sum_{k=2}^{m}\binom{m}{k}(-\mathsf{S}\mathsf{S}^{+})^{k}\boldsymbol{V}^{(n)} && (3.14)\\
&= m\mathsf{S}(\boldsymbol{v} - \boldsymbol{v}^{*}) && (3.15)
\end{aligned}$$

where

$$\boldsymbol{v}^{*} = \frac{1}{m}\sum_{k=2}^{m}\binom{m}{k}(-1)^{k}(\mathsf{S}^{+}\mathsf{S})^{k-1}\boldsymbol{v} = \sum_{k=2}^{m}(-1)^{k}\boldsymbol{v}_{k}^{*} \tag{3.16}$$

where

$$\boldsymbol{v}_{k}^{*} = \frac{1}{m}\binom{m}{k}(\mathsf{S}^{+}\mathsf{S})^{k-1}\boldsymbol{v} = \frac{1}{m}\binom{m}{k}\mathsf{S}^{+}\mathsf{S}(\mathsf{S}^{+}\mathsf{S})^{k-2}\boldsymbol{v} = \frac{\frac{1}{m}\binom{m}{k}}{\frac{1}{m}\binom{m}{k-1}}\mathsf{S}^{+}\mathsf{S}\boldsymbol{v}_{k-1}^{*} = \frac{m-k+1}{k}\mathsf{S}^{+}\mathsf{S}\boldsymbol{v}_{k-1}^{*} \tag{3.17}$$

starting with

$$\boldsymbol{v}_{1}^{*} = \boldsymbol{v} \tag{3.18}$$

As explicit sums

$$\begin{aligned}
(\boldsymbol{v}_{k}^{*})_{i} &= \frac{m-k+1}{k}\sum_{p}\frac{M_{p}S_{pi}}{\boldsymbol{m}_{i}^{(n)}}(\mathsf{S}\boldsymbol{v}_{k-1}^{*})_{p} \qquad \forall\ i \in \Omega_{p} && (3.19)\\
&= \frac{m-k+1}{k}\sum_{p}\sum_{j\in\Omega_{p}}\frac{M_{p}S_{pi}S_{pj}}{\boldsymbol{m}_{i}^{(n)}}(\boldsymbol{v}_{k-1}^{*})_{j} \qquad \forall\ i \in \Omega_{p} && (3.20)
\end{aligned}$$

It might make sense to to define an adjusted grid velocity as $\boldsymbol{v}(m) = m(\boldsymbol{v} - \boldsymbol{v}^{*})$ such that $\mathsf{P}\boldsymbol{V}^{(n)} = \mathsf{S}\boldsymbol{v}(m)$. This velocity is found like above using:

$$\boldsymbol{v}(m) = m(\boldsymbol{v} - \boldsymbol{v}^{*}) = m\left(\boldsymbol{v}_{1}^{*} - \sum_{k=2}^{m}(-1)^{k}\boldsymbol{v}_{k}^{*}\right) = m\sum_{k=1}^{m}(-1)^{k+1}\boldsymbol{v}_{k}^{*} \tag{3.21}$$

It might look better to bring m inside the sum

$$\boldsymbol{v}(m) = \sum_{k=1}^{m}(-1)^{k+1}m\boldsymbol{v}_{k}^{*} = \sum_{k=1}^{m}(-1)^{k+1}\boldsymbol{v}_{k} \tag{3.22}$$

where

$$\boldsymbol{v}_k = \binom{m}{k}(\mathsf{S}^+\mathsf{S})^{k-1}\boldsymbol{v} = \binom{m}{k}\mathsf{S}^+\mathsf{S}(\mathsf{S}^+\mathsf{S})^{k-2}\boldsymbol{v} = \frac{\binom{m}{k}}{\binom{m}{k-1}}\mathsf{S}^+\mathsf{S}\boldsymbol{v}_{k-1} = \frac{m-k+1}{k}\mathsf{S}^+\mathsf{S}\boldsymbol{v}_{k-1} \tag{3.23}$$

starting with

$$\boldsymbol{v}_1 = m\boldsymbol{v} \tag{3.24}$$

Or removing the recurrence relation:

$$\boldsymbol{v}(m) = \sum_{k=1}^{m}\binom{m}{k}(-1)^{k+1}(\mathsf{S}^+\mathsf{S})^{k-1}\boldsymbol{v} \tag{3.25}$$

The XPIC updates become

$$\boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v}(m) + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp})\Delta t \tag{3.26}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(\mathsf{S}\boldsymbol{v}(m) + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{3.27}$$

or mixed with fraction β of FLIP become:

$$\boldsymbol{V}^{(n+1)} = \beta\boldsymbol{V}^{(n)} + (1-\beta)\mathsf{S}\boldsymbol{v}(m) + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp})\Delta t \tag{3.28}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(\beta\mathsf{S}\boldsymbol{v} + (1-\beta)\mathsf{S}\boldsymbol{v}(m) + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{3.29}$$

Previous XPIC notes with expanded out displacement updates is give in the Appendix

3.3 An Alternate Approach to Implementing XPIC(m)

If the $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}$ (or its shifted version) was used, it suggests a different approach to XPIC(m) done elsewhere in the time step with the updates being simpler (and closer to FLIP). The potential approach is:

Extrapolate to the Grid

Immediately after extrapolating to the grid (and probably before doing contact calculations), replace grid $\boldsymbol{p}$ with $\mathsf{m}\boldsymbol{v}(m)$ on the grid. The calculation is:

$$\boldsymbol{v}(m) = m(\boldsymbol{v} - \boldsymbol{v}^*) = m\sum_{k=1}^{m}(-1)^{k+1}\boldsymbol{v}^*_k \tag{3.30}$$

starting with $\boldsymbol{v}^*_1 = \boldsymbol{v}$. Note that $\boldsymbol{v}(m)$ could be accumulated direction in $\boldsymbol{p}$ (and scaled by m_i at some point), to eliminate the need to store $\boldsymbol{v}^*$ on each node during XPIC(m) calculations. For FLIP and PIC, this calculation is skipped.

Various Tasks

Various calculations that depend on grid velocity would now be done using $\boldsymbol{v}(m)$. Examples include

boundary conditions, contact, and constitutive law. For example the velocity gradient on the particles would be found from:

$$\nabla \boldsymbol{v}_p = \sum_{i \in \Omega_p} G_{ip} \boldsymbol{v}(m) \tag{3.31}$$

This change might be an advantage of this approach to XPIC(m) giving it more influence on particle stresses than other approaches. It might be crucial for boundary condition and contact methods.

Particle Update

Because original grid velocity is discarded after extrapolating to the grid, the particle updates most easily would use $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)}$ (or its shifted version), but may need to use $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$ by storing $\boldsymbol{v}$ for later use. The updates are others the same as above (see Eqs. (3.26) and (3.27)). Note that because $\boldsymbol{v}$ might be discarded the grid damping term will need to be scaled by $\mathsf{S}\boldsymbol{v}(m)$, such as:

$$\boldsymbol{A}^{(0)}_{damp} = \alpha_g \mathsf{S}\boldsymbol{v}(m) + \alpha_p \boldsymbol{V}^{(n)} \tag{3.32}$$

Because grid damping is trial-and-error term, this change should not matter. It is tempting to replace $\boldsymbol{V}^{(n)}$ with $\mathsf{S}\boldsymbol{v}(m)$ after the extrapolations, but then $\boldsymbol{V}^{(n)}$ would not be available for damping or for position update in the particle updates.

Second Extrapolation

A drawback of this approach is that when using `USAVG+` or `USL+`, will have to redo XPIC(m) extrapolations to convert grid $\boldsymbol{v}$ to $\boldsymbol{v}(m)$ before second round of boundary conditions, contact, and stress updates.

This approach was implement in OSParticulas, but appears not the be a good idea. First, it did not seem to help much and and update with $\mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}$ is less stable (perhaps because it applies less smoothing to position update). I tried using this revised approach and storing initial velocity to all position update to still use $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$. It may have been slightly better, but not complicated on maintaining meaning of stored $\boldsymbol{v}$ when other things happen (like contact). Second, the revised XPIC(m) for contact (see below) works bettern in standard XPIC(m) than in this alternate approach. Last, tracking dissipated energy might be harder. The option for a revised update was removed in revision 2115.

3.4 Energy Dissipation

For the usually case of $\mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v}$, the energy dissipation on a particle from above is:

$$\Delta E_{p,disp} = M_p(\boldsymbol{A}_{g\to p} - \mathbb{A}^{(n)}_p)\Delta t \cdot \left(\tilde{\boldsymbol{V}}_{g\to p} - \frac{\Delta t}{2}(\boldsymbol{A}_{g\to p} - \mathbb{A}^{(n)}_p)\right) \tag{3.33}$$

where $\tilde{\boldsymbol{V}}_{g\to p}$ is the p^{th} element of $\mathsf{S}\tilde{\boldsymbol{v}}$. For XPIC($m$) with damping:

$$(\boldsymbol{A}_{g\to p} - \mathbb{A}^{(n)}_p)\Delta t = \boldsymbol{V}^{(n)}_p - m(\boldsymbol{V}_{g\to p} - \boldsymbol{V}^{*}_{g\to p}) + \boldsymbol{A}^{(0)}_{p,damp}\Delta t \tag{3.34}$$

$\boldsymbol{V}^{*}_{g\to p}$ is the p^{th} element $\mathsf{S}\boldsymbol{v}^{*}$.

3.5 Energy Dissipation - Previous

We can rewrite the pure XPIC updates as

$$\boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v}(m) + (\mathsf{S}\boldsymbol{a} - \boldsymbol{A}^{(0)}_{damp})\Delta t \tag{3.35}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \begin{cases} \left(2\mathsf{S}\boldsymbol{v} + \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}\right)\frac{\Delta t}{2} & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(2\mathsf{S}\boldsymbol{v}(m) + \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}\right)\frac{\Delta t}{2} & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{3.36}$$

The energy change on one particle between state n and $n+1$ is

$$\Delta E_{p,XPIC} = \frac{M_p \Delta \boldsymbol{V}_p}{\Delta t} \cdot \Delta \boldsymbol{X}_p \tag{3.37}$$

$$= \frac{1}{2} M_p \left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) \cdot \begin{cases} \left(2\boldsymbol{V}^*_{g\to p} + \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(2\boldsymbol{V}_{g\to p} + \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{3.38}$$

where $\boldsymbol{V}_p$ and $\boldsymbol{X}_p$ are p^{th} elements for the corresponding vectors, $\boldsymbol{V}^*_{g\to p}$ is the p^{th} element of $\mathsf{S}\boldsymbol{v}$, and $\boldsymbol{V}_{g\to p}$ is the p^{th} element of $\mathsf{S}\boldsymbol{v}(m)$. For undamped FLIP, the updates are:

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \tag{3.39}$$

$$\boldsymbol{X}^{(n+1)} = \boldsymbol{X}^{(n)} + \left(\mathsf{S}\boldsymbol{v} + \frac{\boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}}{2}\right)\Delta t \tag{3.40}$$

with energy change on one particle of

$$\Delta E_{p,FLIP} = M_p \boldsymbol{A}_{g\to p}\Delta t \cdot \left(\boldsymbol{V}^*_{g\to p} + \frac{1}{2}\left(\boldsymbol{A}_{g\to p} - \boldsymbol{A}^{(0)}_{damp,p}\right)\Delta t\right) \tag{3.41}$$

$$= \frac{1}{2} M_p \left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) \cdot \left(2\boldsymbol{V}^*_{g\to p} + \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}\right) \tag{3.42}$$

If there is no external damping ($\boldsymbol{A}^{(0)}_{damp} \to 0$) the energy change by FLIP and XPIC on one particle are

$$\Delta E^{(0)}_{p,XPIC} = M_p \left(\boldsymbol{A}_{g\to p}\Delta t + \boldsymbol{V}_{g\to p} - \boldsymbol{V}_p^{(n)}\right) \cdot \begin{cases} \left(\boldsymbol{V}^*_{g\to p} + \frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t + \frac{1}{2}\left(\boldsymbol{V}_{g\to p} - \boldsymbol{V}_p^{(n)}\right)\right) & \mathbb{V}^{(n)} = \mathsf{S}\boldsymbol{v} \\ \left(\boldsymbol{V}_{g\to p} + \frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t + \frac{1}{2}\left(\boldsymbol{V}_{g\to p} - \boldsymbol{V}_p^{(n)}\right)\right) & \mathbb{V}^{(n)} = \mathsf{P}\boldsymbol{V}^{(n)} \end{cases} \tag{3.43}$$

$$\Delta E^{(0)}_{p,FLIP} = M_p \boldsymbol{A}_{g\to p}\Delta t \cdot \left(\boldsymbol{V}^*_{g\to p} + \frac{1}{2}\boldsymbol{A}_{g\to p}\Delta t\right) \tag{3.44}$$

The energy change using FLIP is physical term of force time distance. The XPIC energy change as same force times distance plus terms that involve difference in two velocities. When these grid and particle velocities are close, the energy changes matches expected force times distance result. Otherwise the velocity difference times Δt is "slippage" leading to energy dissipation.

Writing out FLIP energy change and summing over particles, total energy change on all particles is:

$$\sum_p \Delta E^{(0)}_{p,FLIP} = \sum_p \left[\left(\sum_i M_p S_{pi} \boldsymbol{a}_i \Delta t\right) \cdot \sum_i S_{pi}\left(\boldsymbol{v}_i + \frac{1}{2}\boldsymbol{a}_i \Delta t\right)\right] \tag{3.45}$$

$$= \sum_p \sum_i \sum_j M_p S_{pi} S_{pj} \boldsymbol{a}_i \Delta t \cdot \left(\boldsymbol{v}_j + \frac{1}{2}\boldsymbol{a}_j \Delta t\right) \tag{3.46}$$

$$= \sum_i \sum_j m^{(n)}_{ij} \boldsymbol{a}_i \Delta t \cdot \left(\boldsymbol{v}_j + \frac{1}{2}\boldsymbol{a}_j \Delta t\right) \tag{3.47}$$

where $m_{ij}^{(n)}$ is element of the full mass matrix. The energy change on the grid in the time step would be

$$\sum_i \Delta E_{i,FLIP}^{(0)} = \sum_i \frac{\Delta \boldsymbol{p}_i}{\Delta t} \cdot \Delta \boldsymbol{x}_i = \sum_i m_i \boldsymbol{a}_i \cdot \left(\boldsymbol{v}_i + \frac{1}{2} \boldsymbol{a}_i \Delta t \right) \tag{3.48}$$

$$= \sum_p \sum_i M_p S_{pi} \boldsymbol{a}_i \cdot \left(\boldsymbol{v}_i + \frac{1}{2} \boldsymbol{a}_i \Delta t \right) \tag{3.49}$$

$$= \sum_i m_i^{(n)} \boldsymbol{a}_i \cdot \left(\boldsymbol{v}_i + \frac{1}{2} \boldsymbol{a}_i \Delta t \right) \tag{3.50}$$

where $m_i^{(n)}$ is from diagonal of the lumped mass matrix. To conserve energy, these two results should be the same. Or lack of conservation could be due to lumping of the mass matrix.

3.6 XPIC Implementation For Transport Tasks

The XPIC method can be applied to updates in transport tasks only. The acceleration becomes a transport value velocity and velocity is replaced by transport value (T which is temperature or concentration). The particle transport for XPIC with optional linear addition of FLIP

$$\mathbb{A}^{(n)} \Delta t = \mathsf{S} \boldsymbol{v}_T \Delta t - (1-\beta)(\mathsf{I} - \mathsf{S}\mathsf{S}^+)^m \boldsymbol{T}^{(n)} \tag{3.51}$$

$$= \mathsf{S} \boldsymbol{v}_T \Delta t - (1-\beta)\left(\boldsymbol{T}^{(n)} - m\mathsf{S}\boldsymbol{t} + m\mathsf{S}\boldsymbol{t}^* \right) \tag{3.52}$$

where $\boldsymbol{t}$ are nodal transport values and S^+ is now

$$S_{ip}^+ = \frac{V_p C_{T,p} S_{pi}}{c_i^{(n)}} \qquad \text{and} \qquad c_i^{(n)} = \sum_p V_p C_{T,p} S_{pi} \tag{3.53}$$

where V_p is particle volume, $C_{T,p}$ is particle transport capacity. The XPIC(m) term is

$$\boldsymbol{t}^* = \frac{1}{m} \sum_{k=2}^{m} \binom{m}{k} (-1)^k (\mathsf{S}^+\mathsf{S})^{k-1} \boldsymbol{t} = \sum_{k=2}^{m} (-1)^k \boldsymbol{t}_k^* \tag{3.54}$$

where

$$\boldsymbol{t}_k^* = \frac{m-k+1}{k} \mathsf{S}^+\mathsf{S} \boldsymbol{t}_{k-1}^* \tag{3.55}$$

starting with

$$\boldsymbol{t}_1^* = \boldsymbol{t} = \tilde{\boldsymbol{t}} - \boldsymbol{v}_T \Delta t \tag{3.56}$$

The explicit sums are analgous to the sums for velocity updates:

$$(\boldsymbol{t}_k^*)_i = \frac{m-k+1}{k} \sum_p \sum_{j \in \Omega_p} \frac{V_p C_{T,p} S_{pi} S_{pj}}{c_i^{(n)}} (\boldsymbol{t}_{k-1}^*)_j \qquad \forall\ i \in \Omega_p \tag{3.57}$$

When particle update is done, nodal transport values will have been updated to $\tilde{\boldsymbol{t}} = \boldsymbol{t} + \boldsymbol{v}_T \Delta t$. The transport particle update becomes:

$$\boldsymbol{T}^{(n+1)} = \beta \boldsymbol{T}^{(n)} + m(1-\beta)\mathsf{S}(\tilde{\boldsymbol{t}} - \boldsymbol{t}^*) + \big(1 - m(1-\beta)\big)\mathsf{S}\boldsymbol{v}_T \Delta t \tag{3.58}$$

or $\boldsymbol{T}^{(n+1)} = \boldsymbol{T}^{(n)} + d\boldsymbol{T}$ where

$$d\boldsymbol{T} = (1-\beta)\big(m(\mathsf{S}(\tilde{\boldsymbol{t}} - \boldsymbol{t}^*) - \mathsf{S}\boldsymbol{v}_T \Delta t) - \boldsymbol{T}^{(n)}\big) + \mathsf{S}\boldsymbol{v}_T \Delta t \tag{3.59}$$

Standard mixed FLIP/PIC is recovered by setting $m = 1$ and $\boldsymbol{t}^* = 0$:

$$\boldsymbol{T}^{(n+1)} = \beta\left(\boldsymbol{T}^{(n)} + \mathsf{S}\boldsymbol{v}_T\Delta t\right) + (1-\beta)\mathsf{S}\tilde{\boldsymbol{t}} \tag{3.60}$$

PIC methods for transport tasks are generally not used because of too much numerical diffusion. For XPIC alone ($\beta = 0$):

$$\boldsymbol{T}^{(n+1)} = m\mathsf{S}(\tilde{\boldsymbol{t}} - \boldsymbol{t}^*) + (1-m)\mathsf{S}\boldsymbol{v}_T\Delta t \tag{3.61}$$

which hopefully will have less numerical diffusion. As a stand-along XPIC(m) task the update is recast as

$$\begin{aligned} \boldsymbol{T}^{(n+1)} &= \tilde{\boldsymbol{T}}^{(n)} + \mathsf{S}\boldsymbol{v}_T\Delta t & (3.62)\\ \tilde{\boldsymbol{T}}^{(n)} &= \beta\boldsymbol{T}^{(n)} + m(1-\beta)\mathsf{S}(\boldsymbol{t} - \boldsymbol{t}^*) & (3.63) \end{aligned}$$

These changes could be done in the XPIC tasks by adding a loop over particles after finding $\boldsymbol{t}^*$. This approach might be better separation of XPIC(m) from rest of code (because the particle update can be normal), but slightly less efficient because the XPIC(m) task is immediately followed by particle update any (and that task loops over particle again).

3.7 XPIC Implementation For Stress

XPIC methods can be applied to particle stress, but rather then incorporate into particle stress update in constitutive law methods (which effectively is based on velocity gradient extrapolated from the grid), it can be applied in a custom task and just apply projection operator to remove null space noise from the stress. The stress null space is

$$\boldsymbol{\Omega}^{(n)} = (\mathsf{I} - \mathsf{S}\mathsf{S}^+)^m\boldsymbol{\Sigma}^{(n)} \tag{3.64}$$

where $\beta = 0$ (because this tasks will not blend FLIP with XPIC) and $\boldsymbol{\Sigma}^{(n)}$ is vector of particle stress tensors (6 component vectors for symmetric stress tensors). The null space is

$$\begin{aligned} (\mathsf{I} - \mathsf{S}\mathsf{S}^+)^m\boldsymbol{\Sigma}^{(n)} &= \boldsymbol{\Sigma}^{(n)} - m\mathsf{S}\boldsymbol{\sigma} + \sum_{k=2}^{m}\binom{m}{k}(-\mathsf{S}\mathsf{S}^+)^k\boldsymbol{\Sigma}^{(n)} & (3.65)\\ &= \boldsymbol{\Sigma}^{(n)} - m\mathsf{S}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) & (3.66) \end{aligned}$$

Subtracting this null space from particle stresses, results in smoothed stress equal to

$$\tilde{\boldsymbol{\Sigma}}^{(n)} = m\mathsf{S}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \tag{3.67}$$

where the grid terms are

$$\begin{aligned} \boldsymbol{\sigma}^* &= \frac{1}{m}\sum_{k=2}^{m}\binom{m}{k}(-1)^k(\mathsf{S}^+\mathsf{S})^{k-1}\boldsymbol{\sigma} = \sum_{k=2}^{m}(-1)^k\boldsymbol{\sigma}_k^* & (3.68)\\ \boldsymbol{\sigma}_k^* &= \frac{m-k+1}{k}\mathsf{S}^+\mathsf{S}\boldsymbol{\sigma}_{k-1}^* & (3.69)\\ \boldsymbol{\sigma}_1^* &= \boldsymbol{\sigma} = \mathsf{S}^+\boldsymbol{\Sigma}^{(n)} & (3.70) \end{aligned}$$

The main difference here is that standard MPM does not extrapolate stress to the grid. Thus, this calculation needs to extrapolate stress to grid, find $\boldsymbol{\sigma}^*$, and then extrapolate back to the particles. The final grid term that is extrapolated back to particles is

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \boldsymbol{\sigma}^* = \boldsymbol{\sigma}_1^* - \sum_{k=2}^{m}(-1)^k\boldsymbol{\sigma}_k^* = \sum_{k=1}^{m}(-1)^{k+1}\boldsymbol{\sigma}_k^* \tag{3.71}$$

The stress XPIC algorithm as stand alone custom tasks is:

1. Initialize: Each node will need three stress tensors to support XPIC extrapolations and they are initialized to
$$\boldsymbol{\sigma}_{prev} = \mathsf{S}^{+}\boldsymbol{\Sigma}^{(n)}, \quad \tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{prev}, \quad \text{and} \quad \boldsymbol{\sigma}_{next} = 0 \tag{3.72}$$
Note that this initialization step needs an extrapolation from particles to the grid.

2. XPIC Extrapolations: PIC is done ($m = 1$), but for higher order XPIC(m), loop k from 2 to m:

   (a) Sum terms needed for $\boldsymbol{\sigma}_{next} = \boldsymbol{\sigma}_k$ that depend only on $\boldsymbol{\sigma}_{prev} = \boldsymbol{\sigma}_{k-1}$. The stress tensor on node i is
$$(\boldsymbol{\sigma}_{next})_i = \frac{m-k+1}{k}\sum_p \sum_{j\in\Omega_p} \frac{M_p S_{pi} S_{pj}}{\boldsymbol{m}_i^{(n)}} (\boldsymbol{\sigma}_{prev})_j \qquad \forall\ i \in \Omega_p \tag{3.73}$$

   (b) At end of each k iteration, add to $\tilde{\boldsymbol{\sigma}}$ and prepare for next k:
$$\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}} + (-1)^{k+1}\boldsymbol{\sigma}_{next}, \quad \boldsymbol{\sigma}_{prev} = \boldsymbol{\sigma}_{next}, \quad \text{and} \quad \boldsymbol{\sigma}_{next} = 0 \tag{3.74}$$

3. Project to Particles: extrapolate $\tilde{\boldsymbol{\sigma}}$ to the particles:
$$\tilde{\boldsymbol{\Sigma}}^{(n)} = m\mathsf{S}\tilde{\boldsymbol{\sigma}} \tag{3.75}$$

Another possibility is to apply XPIC to forces instead of stresses. Maybe it would need less memory and be more general for materials that do not use incremental stress updates (where non-incremental will erase all the filtering). Imagine doing stress XPIC just before the force extrapolations. Those extrapolations using smoothed or initial stresses would be

$$\tilde{\boldsymbol{f}}^{(n)} = -\mathsf{G}^{+}\tilde{\boldsymbol{\Sigma}}^{(n)} \quad \text{and} \quad \boldsymbol{f}^{(n)} = -\mathsf{G}^{+}\boldsymbol{\Sigma}^{(n)} \tag{3.76}$$

where G^{+} is matrix of shape function gradient vectors with elements:

$$G^{+}_{ip} = \frac{M_p G_{pi}}{\rho_{p,0}} \tag{3.77}$$

and $\tilde{\boldsymbol{\Sigma}}^{(n)}$ and $\boldsymbol{\Sigma}^{(n)}$ are vectors with Kirchoff stress of particle as elements (note that code uses specific Kirchoff stress or $\rho_{p,0}$ are inside the stress vectors). The XPIC-smoothed force in terms of unsmoothed force is:

$$\tilde{\boldsymbol{f}}^{(n)} = \boldsymbol{f}^{(n)} - \mathsf{G}^{+}\left(\tilde{\boldsymbol{\Sigma}}_{\mathsf{p}}{}^{(n)} - \boldsymbol{\Sigma}_{\mathsf{p}}{}^{(n)}\right) \tag{3.78}$$

The final term is effectively the null space of forces found by extrapolating stress that have had null space remove or:

$$\boldsymbol{\Omega}_{\boldsymbol{f}}{}^{(n)} = -\mathsf{G}^{+}\boldsymbol{\Omega}^{(n)} \tag{3.79}$$

The question remains — can XPIC be done directly on forces or should a stress XPIC task be inserted just before extrapolation of forces to the grid?

Chapter 4

Multimaterial XPIC(m) Extrapolations

4.1 Introduction

This chapter proposes a formal definition of multimaterial mode MPM with goal to decide on the proper particle update. One criterion is that multimaterial mode MPM should reduce to single material mode MPM if interfaces between all materials are perfect. It turns XPIC(m) cannot reduce to perfect interface, but a new term is introduced that should minimize the error.

4.2 Extrapolation to the Grid and Contact conditions

We treat extrapolation to the grid plus imposition of contact changes as the net extrapolation and momentum and mass to the grid. The grid velocity for each separately tracked material (α, β, ...) becomes

$$\boldsymbol{v}^{\alpha} = \mathsf{S}^{+\alpha}\boldsymbol{V} + \Delta\boldsymbol{v}^{\alpha} \tag{4.1}$$

where $\mathsf{S}^{+\alpha}$ extrapolates only particle with $p \in \alpha$

$$S_{ip}^{+\alpha} = \begin{cases} \frac{M_p S_{pi}}{m_i^{\alpha}} & p \in \alpha \\ 0 & p \notin \alpha \end{cases} \tag{4.2}$$

and $\Delta\boldsymbol{v}^{\alpha}$ is the velocity change caused by contact calculations for material α.

The extrapolated momenta and mass for material α ignoring contact are

$$\boldsymbol{p}_{i,0}^{\alpha} = \sum_p S_{pi}^{\alpha}\boldsymbol{p}_p \qquad \text{and} \qquad m_i^{\alpha} = \sum_p M_p S_{pi}^{\alpha} \tag{4.3}$$

where

$$S_{pi}^{\alpha} = \begin{cases} S_{pi} & p \in \alpha \\ 0 & p \notin \alpha \end{cases} \tag{4.4}$$

In matrix form, we can get initial grid momenta ($\boldsymbol{p}_0$) from particle momenta ($\boldsymbol{P}$) using

$$\boldsymbol{p}_0^{\alpha} = \mathsf{S}^{\alpha T}\boldsymbol{P} = \mathsf{m}^{\alpha}\mathsf{S}^{+\alpha}\boldsymbol{V} = \mathsf{m}^{\alpha}\boldsymbol{v}_0^{\alpha} \tag{4.5}$$

where m^{α} is a diagonal matrix (size equal to number of nodes) with m_i^{α} on corresponding diagonal for each node that sees material α and $\boldsymbol{v}_0^{\alpha}$ is velocity extrapolation that ignores contact.

The contact changes can be written in terms of changes in momenta:

$$\Delta\boldsymbol{p}^{\alpha} = \mathsf{m}^{\alpha}\Delta\boldsymbol{v}^{\alpha} \tag{4.6}$$

The grid momenta after imposing contact laws becomes

$$\boldsymbol{p}^\alpha = \boldsymbol{p}_0^\alpha + \Delta\boldsymbol{p}^\alpha = \mathsf{m}^\alpha \boldsymbol{v}^\alpha \quad \text{and} \quad \boldsymbol{v}^\alpha = \mathsf{m}^{-\alpha}\boldsymbol{p}^\alpha \tag{4.7}$$

where $\mathsf{m}^{-\alpha}$ has $1/m_i^\alpha$ on the diagonal for nodes that see material α.

4.2.1 Single Material Mode

The corresponding single material mode will have

$$\boldsymbol{v} = \mathsf{S}^+ \boldsymbol{V} \quad \text{and} \quad \boldsymbol{p} = \mathsf{S}^T \boldsymbol{P} \tag{4.8}$$

Note that S^+ normalizes row i by m_i while $\mathsf{S}^{+\alpha}$ normalizes it by m_i^α. If useful, we could write

$$\mathsf{S}^+ = \sum_\alpha \mathsf{S}_\alpha^+ = \mathsf{m}^{-1} \sum_\alpha \mathsf{m}^\alpha \mathsf{S}^{+\alpha} \tag{4.9}$$

or

$$S_{\alpha,ip}^+ = \begin{cases} \frac{M_p S_{pi}}{m_i} & p \in \alpha \\ 0 & p \notin \alpha \end{cases} \tag{4.10}$$

There will be no changes due to contact, but single and multimaterial modes are related by:

$$\boldsymbol{p} = \sum_\alpha \boldsymbol{p}_0^\alpha = \sum_\alpha \boldsymbol{p}^\alpha \tag{4.11}$$

The latter holds provided contact changes conserve momentum or

$$\sum_\alpha \Delta\boldsymbol{p}^\alpha = 0 \tag{4.12}$$

This condition is easily satisfied for two materials in contact, but may only be approximate for three or more materials in contact on a single node.

4.2.2 Perfect Interfaces

If the interfaces are perfect than all materials will move in the center of mass velocity fields by invoking

$$\Delta\boldsymbol{v}^\alpha = \boldsymbol{v} - \boldsymbol{v}_0^\alpha \quad \text{and} \quad \Delta\boldsymbol{p}^\alpha = \mathsf{m}^\alpha \mathsf{m}^{-1}\boldsymbol{p} - \boldsymbol{p}_0^\alpha \tag{4.13}$$

4.3 Update Stresses and Strains on Particle

When updating stresses and strains before finding forces and updating momenta (needed for `USAVG±` and for `USF` update methods), extrapolate velocity gradient ($\nabla\boldsymbol{v}$) to the particles and implement the chosen constitutive laws. The multimaterial mode extrapolations are:

$$\nabla\boldsymbol{v}_{p\in\alpha} = \sum_i \boldsymbol{v}_i^\alpha \otimes \boldsymbol{G}_{ip} \tag{4.14}$$

In other words, extrapolate the velocity field associated with particle p

Other additions are traction BC and body forces and imposition for forces for velocity BCs (visit later).

4.3.1 Single Material Mode

Single material mode extrapolates the center of mass velocity:

$$\nabla \boldsymbol{v}_p = \sum_i \boldsymbol{v}_i \otimes \boldsymbol{G}_{ip} \tag{4.15}$$

This differ from multimaterial mode because they are implementing the effects of contact.

4.3.2 Perfect Interfaces

Because $\boldsymbol{v}_i^\alpha = \boldsymbol{v}_i$ for a perfect interface, the velocity gradient on the particles will be same in multimaterial mode with perfect interfaces as the velocity gradient in single material mode. The result particle stresses and strain will also be the same.

4.4 Project Forces to the Grid

For each material point, extrapolate $\boldsymbol{f}_{i,0}^\alpha$ due to particle stresses and particle forces (later not used often):

$$\boldsymbol{f}_{i,0}^\alpha = \sum_{p \in j} \left(-M_p \frac{\boldsymbol{\tau}_p \cdot \boldsymbol{G}_{ip}}{\rho_0} + \boldsymbol{F}_p S_{ip} \right) \tag{4.16}$$

In other words, extrapolate each material type differently. A vector of material forces is $\boldsymbol{f}_0^\alpha$. These forces do not include any force implied by contact laws or by boundary condition.

4.4.1 Single Material Mode

Single material mode extrapolates all materials to each node. As a result, the resulting forces $\boldsymbol{f}$ are related to material forces by:

$$\boldsymbol{f} = \sum_\alpha \boldsymbol{f}_0^\alpha \tag{4.17}$$

In general, the nodal accelerations will differ and reflect the contact mechanics. It is important to handle accelerations correctly.

4.4.2 Perfect Interfaces

Because of mass-weighted stress in force extrapolation, the relation between modes for a perfect interfaces is not simple. An important criterion, however, is that in the particle updates, the acceleration extrapolated to particles must be the same for all materials or:

$$\boldsymbol{a}_i = \frac{\boldsymbol{f}_i}{m_i} = \frac{\boldsymbol{f}_i^\alpha}{m_i^\alpha} = \boldsymbol{a}_i^\alpha \tag{4.18}$$

This criterion will help determine the calculation for material force account for contact.

4.5 Update Nodal Momenta

This task updates the nodal momenta for each material. Because forces are found from stresses that account for initial contact conditions, we write

$$\boldsymbol{p}_0^{\alpha+} = \boldsymbol{p}^{\alpha} + \boldsymbol{f}_0^{\alpha}\Delta t = \boldsymbol{p}_0^{\alpha} + \Delta\boldsymbol{p}^{\alpha} + \boldsymbol{f}_0^{\alpha}\Delta t \tag{4.19}$$

This update is a provisional update. Because particles may have come into contact, a second contact calculation is done to find the final momenta for each material:

$$\begin{aligned} \boldsymbol{p}^{\alpha+} = \boldsymbol{p}_0^{\alpha+} + \Delta\boldsymbol{p}^{\alpha+} &= \boldsymbol{p}_0^{\alpha} + \Delta\boldsymbol{p}^{\alpha} + \Delta\boldsymbol{p}^{\alpha+} + \boldsymbol{f}_0^{\alpha}\Delta t \\ &= \boldsymbol{p}^{\alpha} + \Delta\boldsymbol{p}^{\alpha+} + \boldsymbol{f}_0^{\alpha}\Delta t \end{aligned} \tag{4.20}$$

4.5.1 Single Material Mode

The update in single material mode is

$$\boldsymbol{p}^{+} = \boldsymbol{p} + \boldsymbol{f}\Delta t \tag{4.21}$$

Because contact corrections (usually) conserve momenta, this update is related to multimaterial mode by

$$\boldsymbol{p}^{+} = \sum_{\alpha} \boldsymbol{p}^{\alpha+} = \sum_{\alpha} \boldsymbol{p}_0^{\alpha+} \tag{4.22}$$

4.5.2 Perfect Interfaces

For perfect interface, the change in momenta after the update will be:

$$\Delta\boldsymbol{p}^{\alpha+} = \mathsf{m}^{\alpha}\mathsf{m}^{-1}\boldsymbol{p}^{+} - \boldsymbol{p}_0^{\alpha+} \tag{4.23}$$

Substitution into momentum update gives:

$$\boldsymbol{p}^{\alpha+} = \mathsf{m}^{\alpha}\mathsf{m}^{-1}\boldsymbol{p}^{+} \quad \text{and} \quad \boldsymbol{v}^{\alpha+} = \mathsf{m}^{-1}\boldsymbol{p}^{+} = \boldsymbol{v}^{+} \tag{4.24}$$

or each material matches the center of mass velocity.

4.6 Particle Updates

4.6.1 FLIP Updates

For a standard FLIP update in material α, we write

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t \tag{4.25}$$

where $\boldsymbol{a}^{\alpha}$ is acceleration of material α. But we have to find this acceleration correctly. Looking at particle updates (see Eq. (4.20)), the acceleration for material α should be determined from a net force, but we can pick net force due to transition from $\boldsymbol{p}_0^{\alpha}$ or from $\boldsymbol{p}^{\alpha}$ to $\boldsymbol{p}^{\alpha+}$ leading to two options:

$$\boldsymbol{a}_0^{\alpha} = \mathsf{m}^{-\alpha}\left(\frac{\Delta\boldsymbol{p}^{\alpha} + \Delta\boldsymbol{p}^{\alpha+}}{\Delta t} + \boldsymbol{f}_0^{\alpha}\right) \tag{4.26}$$

$$\boldsymbol{a}^{\alpha} = \mathsf{m}^{-\alpha}\left(\frac{\Delta\boldsymbol{p}^{\alpha+}}{\Delta t} + \boldsymbol{f}_0^{\alpha}\right) \tag{4.27}$$

The next two section show that $\boldsymbol{a}^\alpha$ is the correct choice.

In other words, we define a force corrected for contact of

$$\boldsymbol{f}^\alpha = \boldsymbol{f}_0^\alpha + \frac{\Delta \boldsymbol{p}^{\alpha+}}{\Delta t} \tag{4.28}$$

The particle update for material α becomes:

$$\boldsymbol{p}^{\alpha+} = \boldsymbol{p}^\alpha + \boldsymbol{f}^\alpha \Delta t \quad \text{with} \quad \boldsymbol{a}^\alpha = \mathsf{m}^{-\alpha} \boldsymbol{f}^\alpha \tag{4.29}$$

Also, when finding contact forces for output, they should be based only on $\Delta \boldsymbol{p}^{\alpha+}$.

Note that this update and contact force forces are from $\boldsymbol{p}^\alpha$ which itself was corrected for contact, but that correction was not included in total force. In other words, $\Delta \boldsymbol{p}^\alpha$ should be treated as part of the initial extrapolation to the grid and not as a contact force. One can often get reasonable results (when using USL update methods) by skipping the first contact calculations. The contact calculations done after the momentum update will get total momentum change for the time step ($\Delta \boldsymbol{p}^\alpha + \Delta \boldsymbol{p}^{\alpha+}$), but this approach does not partition contact calculations into momentum change due to extrapolation mismatches and changes that physically corresponding for force. It will not be correct. The contact law calculations will be based on wrong contact stress and particle update will be based on wrong acceleration. The errors are hard to see because it appears that $\Delta \boldsymbol{p}^\alpha << \Delta \boldsymbol{p}^{\alpha+}$ in well modeled or quasi-static problems. For harder problems and shock physics, however, the problems grow.

Single Material Mode

In single material mode

$$\boldsymbol{V}^{(n+1)} = \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}\Delta t \quad \text{where} \quad \boldsymbol{a} = \mathsf{m}^{-1}\boldsymbol{f} \tag{4.30}$$

Perfect Interfaces

For perfect interface, the change in momenta after the update will be:

$$\Delta \boldsymbol{p}^{\alpha+} = \mathsf{m}^\alpha \mathsf{m}^{-1} \boldsymbol{p}^+ - \boldsymbol{p}_0^{\alpha+} = \mathsf{m}^\alpha \mathsf{m}^{-1} \boldsymbol{p}^+ - \boldsymbol{p}^\alpha - \boldsymbol{f}_0^\alpha \Delta t \tag{4.31}$$

For perfect interfaces $\boldsymbol{p}^\alpha = \mathsf{m}^\alpha \mathsf{m}^{-1} \boldsymbol{p}$ leading to

$$\Delta \boldsymbol{p}^{\alpha+} = \mathsf{m}^\alpha \mathsf{m}^{-1} \left(\boldsymbol{p}^+ - \boldsymbol{p} \right) - \boldsymbol{f}_0^\alpha \Delta t \tag{4.32}$$

Finally:

$$\boldsymbol{a}^\alpha = \mathsf{m}^{-\alpha} \boldsymbol{f}^\alpha = \mathsf{m}^{-\alpha} \left(\boldsymbol{f}_0^\alpha + \mathsf{m}^\alpha \mathsf{m}^{-1} \frac{\boldsymbol{p}^+ - \boldsymbol{p}}{\Delta t} - \boldsymbol{f}_0^\alpha \right) = \mathsf{m}^{-1} \boldsymbol{f} = \boldsymbol{a} \tag{4.33}$$

which matches single material mode.

4.6.2 PIC Updates

For a PIC update, we expect to extrapolate the updated (and contact-law corrected) velocity to the particles:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v}^{\alpha+} &= \mathsf{S}\boldsymbol{v}^\alpha + \mathsf{S}\boldsymbol{a}^\alpha \Delta t & (4.34) \\ &= \mathsf{S}\mathsf{S}^{+\alpha} \boldsymbol{V}^{(n)} + \mathsf{S}\Delta \boldsymbol{v}^\alpha + \mathsf{S}\boldsymbol{a}^\alpha \Delta t & (4.35) \end{aligned}$$

The effective acceleration becomes:

$$\begin{aligned} \mathbb{A}^{(n)} \Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)} &= \mathsf{S}\Delta \boldsymbol{v}^\alpha + \mathsf{S}\boldsymbol{a}^\alpha \Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha}) \boldsymbol{V}^{(n)} & (4.36) \\ &= \mathsf{S}\boldsymbol{a}^\alpha \Delta t - \left(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^\alpha \right) & (4.37) \end{aligned}$$

Single Material Mode

In single material mode

$$\begin{aligned} \boldsymbol{V}^{(n+1)} = \mathsf{S}\boldsymbol{v}^{+} &= \mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}\Delta t && (4.38)\\ \mathbb{A}^{(n)}\Delta t &= \mathsf{S}\boldsymbol{a}\Delta t - \left(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}\right) && (4.39)\end{aligned}$$

Perfect Interfaces

For perfect interfaces $\boldsymbol{v}^{\alpha} = \boldsymbol{v}$ and $\boldsymbol{a}^{\alpha} = \boldsymbol{a}$, which makes PIC same in multimaterial and single material mode.

4.6.3 XPIC(2) Updates

A standard FLIP update for particle of material α (without damping) can alternatively be written:

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t = \boldsymbol{V}^{(n)} + \mathsf{S}\boldsymbol{v}^{\alpha+} - \mathsf{S}\boldsymbol{v}^{\alpha} && (4.40)\\ &= \mathsf{S}\boldsymbol{v}^{\alpha+} + \boldsymbol{V}^{(n)} - \mathsf{S}\mathsf{S}^{+\alpha}\boldsymbol{V} - \mathsf{S}\Delta\boldsymbol{v}^{\alpha} && (4.41)\\ &= \mathsf{S}\boldsymbol{v}^{\alpha+} + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\boldsymbol{V}^{(n)} - \mathsf{S}\Delta\boldsymbol{v}^{\alpha} && (4.42)\end{aligned}$$

Following XPIC(2) methods for single material mode, replace $\boldsymbol{V}^{(n)}$ by the initial PIC velocity (now including contact corrections) of $\mathsf{S}\boldsymbol{v}^{\alpha} = \mathsf{S}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)} + \mathsf{S}\Delta\boldsymbol{v}^{\alpha}$. The acceleration on the particle ($\mathbb{A}^{(n)}\Delta t = \boldsymbol{V}^{(n+1)} - \boldsymbol{V}^{(n)}$) becomes

$$\begin{aligned} \mathbb{A}^{(n)}\Delta t &= \mathsf{S}\boldsymbol{v}^{\alpha} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})(\mathsf{S}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)} + \mathsf{S}\Delta\boldsymbol{v}^{\alpha}) - \mathsf{S}\Delta\boldsymbol{v}^{\alpha} - \boldsymbol{V}^{(n)} && (4.43)\\ &= \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\Delta\boldsymbol{v}^{\alpha} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})^2\boldsymbol{V}^{(n)} && (4.44)\\ &= \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^{\alpha}) && (4.45)\\ &= \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\Delta\boldsymbol{v}^{\alpha} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})^2\boldsymbol{V}^{(n)} && (4.46)\end{aligned}$$

The update becomes

$$\begin{aligned} \boldsymbol{V}^{(n+1)} &= \boldsymbol{V}^{(n)} - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^{\alpha}) + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t && (4.47)\\ &= \mathsf{S}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)} + (\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v}^{\alpha} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t && (4.48)\\ &= (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v}^{\alpha} - \mathsf{S}\Delta\boldsymbol{v}^{\alpha} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t && (4.49)\end{aligned}$$

Single Material Mode

In single material mode

$$\begin{aligned} \mathbb{A}^{(n)}\Delta t &= \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})^2\boldsymbol{V}^{(n)} = \mathsf{S}\boldsymbol{a}\Delta t - (\mathsf{I} - \mathsf{S}\mathsf{S}^{+})(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}) && (4.50)\\ \boldsymbol{V}^{(n+1)} &= (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+})\mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t && (4.51)\end{aligned}$$

Perfect Interfaces

Because of the $\mathsf{S}\mathsf{S}^{+}$ compared to the $\mathsf{S}\mathsf{S}^{+\alpha}$ term, this method will not revert to single material mode when perfect interfaces. Single material mode is filtering global material velociities while multimaterial mode filters each material separately. Filtering globally is correct for perfect interfaces, but not for imperfect and likely very bad for separated interfaces.

More explicitly, the single material mode is

$$\mathsf{P}\boldsymbol{V}^{(n)} = (2\mathsf{I}-\mathsf{S}\mathsf{S}^{+})\mathsf{S}\boldsymbol{v} = \mathsf{S}\left[\boldsymbol{v} + (\mathsf{I}-\mathsf{S}^{+}\mathsf{S})\boldsymbol{v}\right] \tag{4.52}$$

In multimaterial mode $\boldsymbol{v}^{\alpha} = \boldsymbol{v}$ and $\Delta\boldsymbol{v}^{\alpha} = \boldsymbol{v} - \boldsymbol{v}_0^{\alpha}$ leading to projection part of update as

$$\begin{aligned}\mathsf{P}\boldsymbol{V}^{(n)} &= (2\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v} - \mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{v}_0^{\alpha} = (\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v} + \mathsf{S}\boldsymbol{v}_0^{\alpha} &(4.53)\\ &= \mathsf{S}\left[\boldsymbol{v}_0^{\alpha} + (\mathsf{I}-\mathsf{S}^{+\alpha}\mathsf{S})\boldsymbol{v}\right] &(4.54)\end{aligned}$$

Substracting multimaterial from single material modes, the difference is:

$$\begin{aligned}\Delta\mathsf{P}\boldsymbol{V}^{(n)} &= \mathsf{S}\left[\boldsymbol{v} - \boldsymbol{v}_0^{\alpha} - (\mathsf{S}^{+}-\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v}\right] &(4.55)\\ &= \mathsf{S}(\mathsf{S}^{+}-\mathsf{S}^{+\alpha})(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}) &(4.56)\end{aligned}$$

Because the difference it not zero, the results in multimaterial mode will no longer match single material mode. But, in the absence of the new $\Delta\boldsymbol{v}^{\alpha}$ term (*i.e*, each material's XPIC same as if a single material), the difference would be:

$$\Delta\mathsf{P}\boldsymbol{V}^{(n)} = -\mathsf{S}(\mathsf{S}^{+}-\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{v} \tag{4.57}$$

This error is proportional to a velocity while the new result is proportional to a velocity difference. The new results should have (in theory) a lot smaller error.

4.6.4 XPIC(m) Updates

Notice that compared to traditional PIC, the $\mathsf{S}\Delta\boldsymbol{v}^{\alpha}$ term in XPIC(2) is scaled by $(\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})$ and the $\boldsymbol{V}^{(n)}$ term is squared. Perhaps we could propose generalized, XPIC(m) update using

$$\begin{aligned}\mathbb{A}^{(n)}\Delta t &= \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t + (\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^{m-1}\mathsf{S}\Delta\boldsymbol{v}^{\alpha} - (\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^{m}\boldsymbol{V}^{(n)} &(4.58)\\ &= \mathsf{S}\boldsymbol{a}^{\alpha}\Delta t - (\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^{m-1}(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^{\alpha}) &(4.59)\end{aligned}$$

where $m = 1$ is PIC, $m = 2$ is result of using smoothed velocity in one term of FLIP update, and $m \to \infty$ converges to the optimal result (hopefully).

To implement, need to evaluate the equivalent of $\mathsf{P}\boldsymbol{V}^{(n)}$ or

$$\begin{aligned}\mathsf{P}\boldsymbol{V}^{(n)} &= \boldsymbol{V}^{(n)} - (\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^{m-1}(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^{\alpha}) &(4.60)\\ &= \mathsf{S}\boldsymbol{v}^{\alpha} - \sum_{k=1}^{m-1}\binom{m-1}{k}(-\mathsf{S}\mathsf{S}^{+\alpha})^{k}(\boldsymbol{V}^{(n)} - \mathsf{S}\boldsymbol{v}^{\alpha}) &(4.61)\\ &= \mathsf{S}\boldsymbol{v}^{\alpha} - \sum_{k=1}^{m-1}\binom{m-1}{k}(-1)^{k}\mathsf{S}(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\left(\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)} - \mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{v}^{\alpha}\right) &(4.62)\end{aligned}$$

Using $\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)} = \boldsymbol{v}^{\alpha} - \Delta\boldsymbol{v}^{\alpha}$ leads to

$$\mathsf{P}\boldsymbol{V}^{(n)} = S\left[\boldsymbol{v}^{\alpha} - \sum_{k=1}^{m-1}\binom{m-1}{k}(-1)^{k}(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\left((\mathsf{I}-\mathsf{S}^{+\alpha}\mathsf{S})\boldsymbol{v}^{\alpha} - \Delta\boldsymbol{v}^{\alpha}\right)\right] \tag{4.63}$$

Evaluating the $k = 1$ term gives

$$\begin{aligned}\mathsf{P}\boldsymbol{V}^{(n)} = \mathsf{S}\Bigg[&m\boldsymbol{v}^{\alpha} - (m-1)\Delta\boldsymbol{v}^{\alpha} - (m-1)\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{v}^{\alpha}\\ &- \sum_{k=2}^{m-1}\binom{m-1}{k}(-1)^{k}(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\left((\mathsf{I}-\mathsf{S}^{+\alpha}\mathsf{S})\boldsymbol{v}^{\alpha} - \Delta\boldsymbol{v}^{\alpha}\right)\Bigg] &(4.64)\end{aligned}$$

The terms involving $\Delta\boldsymbol{v}^\alpha$ reduce to

$$-(m-1)\mathsf{S}\left[\Delta\boldsymbol{v}^\alpha-\frac{1}{m-1}\sum_{k=2}^{m-1}\binom{m-1}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\Delta\boldsymbol{v}^\alpha\right] \tag{4.65}$$

The terms involving $\boldsymbol{v}^\alpha$ reduce to

$$\mathsf{S}\left[m\boldsymbol{v}^\alpha-(m-1)\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{v}^\alpha-\sum_{k=2}^{m-1}\binom{m-1}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}(\mathsf{I}-\mathsf{S}^{+\alpha}\mathsf{S})\boldsymbol{v}^\alpha\right] \tag{4.66}$$

$$= \mathsf{S}\left[m\boldsymbol{v}^\alpha-(m-1)\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{v}^\alpha-\sum_{k=2}^{m-1}\binom{m-1}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\mathsf{S})\boldsymbol{v}^\alpha-\sum_{k=3}^{m}\binom{m-1}{k-1}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\boldsymbol{v}^\alpha\right]$$

Pulling out the $k=2$ term of first sum and make use of:

$$\binom{m}{k}=\binom{m-1}{k}+\binom{m-1}{k-1}\quad\text{and}\quad\binom{m-1}{m}=0 \tag{4.67}$$

where last is used to extend first sum to m, this results simplifies to:

$$m\mathsf{S}\left[\boldsymbol{v}^\alpha-\frac{1}{m}\sum_{k=2}^{m}\binom{m}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\boldsymbol{v}^\alpha\right] \tag{4.68}$$

Alternatively, look at the two terms separately:

$$\mathsf{P}\boldsymbol{V}^{(n)} = \boldsymbol{V}^{(n)}-(\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^m\boldsymbol{V}^{(n)}+(\mathsf{I}-\mathsf{S}\mathsf{S}^{+\alpha})^{m-1}\mathsf{S}\Delta\boldsymbol{v}^\alpha \tag{4.69}$$

$$= m\mathsf{S}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)}+\mathsf{S}\Delta\boldsymbol{v}^\alpha-\sum_{k=2}^{m}\binom{m}{k}(-\mathsf{S}\mathsf{S}^{+\alpha})^k\boldsymbol{V}^{(n)}+\sum_{k=1}^{m-1}\binom{m-1}{k}(-\mathsf{S}\mathsf{S}^{+\alpha})^k\mathsf{S}\Delta\boldsymbol{v}^\alpha \tag{4.70}$$

Rearranging gives:

$$= \mathsf{S}\left(m\boldsymbol{v}^\alpha-(m-1)\Delta\boldsymbol{v}^\alpha-\sum_{k=2}^{m}\binom{m}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)}+\sum_{k=1}^{m-1}\binom{m-1}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^k\Delta\boldsymbol{v}^\alpha\right)$$

$$= \mathsf{S}\left(m\boldsymbol{v}^\alpha-(m-1)\Delta\boldsymbol{v}^\alpha-\sum_{k=2}^{m}\binom{m}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\mathsf{S}^{+\alpha}\boldsymbol{V}^{(n)}-\sum_{k=2}^{m}\binom{m-1}{k-1}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\Delta\boldsymbol{v}^\alpha\right)$$

$$= \mathsf{S}\left(m\boldsymbol{v}^\alpha-(m-1)\Delta\boldsymbol{v}^\alpha-\sum_{k=2}^{m}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\left(\binom{m}{k}(\boldsymbol{v}^\alpha-\Delta\boldsymbol{v}^\alpha)+\binom{m-1}{k-1}\Delta\boldsymbol{v}^\alpha\right)\right)$$

$$= \mathsf{S}\left(m\boldsymbol{v}^\alpha-(m-1)\Delta\boldsymbol{v}^\alpha-\sum_{k=2}^{m}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\left(\binom{m}{k}\boldsymbol{v}^\alpha-\binom{m-1}{k}\Delta\boldsymbol{v}^\alpha)\right)\right) \tag{4.71}$$

This result agrees with first derivation (and was slightly easier). Defining new terms gives

$$\mathsf{P}\boldsymbol{V}^{(n)}=m\mathsf{S}(\boldsymbol{v}^\alpha-\boldsymbol{v}^{*\alpha})-(m-1)\mathsf{S}(\Delta\boldsymbol{v}^\alpha-\Delta\boldsymbol{v}^{*\alpha}) \tag{4.72}$$

where

$$\boldsymbol{v}^{*\alpha} = \frac{1}{m}\sum_{k=2}^{m}\binom{m}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\boldsymbol{v}^\alpha=\sum_{k=2}^{m}(-1)^k\boldsymbol{v}_k^{*\alpha} \tag{4.73}$$

$$\Delta\boldsymbol{v}^{*\alpha} = \frac{1}{m-1}\sum_{k=2}^{m}\binom{m-1}{k}(-1)^k(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\Delta\boldsymbol{v}^\alpha=\sum_{k=2}^{m}(-1)^k\Delta\boldsymbol{v}_k^{*\alpha} \tag{4.74}$$

From prior XPIC(m)

$$\boldsymbol{\nu}_k^{*\alpha} = \frac{m-k+1}{k}\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{\nu}_{k-1}^{*} \quad \text{with} \quad \boldsymbol{\nu}_1^{*\alpha} = \boldsymbol{\nu}^{\alpha} \tag{4.75}$$

The new term is:

$$\begin{aligned}\Delta\boldsymbol{\nu}_k^{*\alpha} &= \frac{1}{m-1}\binom{m-1}{k}(\mathsf{S}^{+\alpha}\mathsf{S})^{k-1}\Delta\boldsymbol{\nu}^{\alpha} = \frac{1}{m-1}\binom{m-1}{k}\mathsf{S}^{+\alpha}\mathsf{S}(\mathsf{S}^{+\alpha}\mathsf{S})^{k-2}\Delta\boldsymbol{\nu}^{\alpha} &(4.76)\\ &= \frac{\frac{1}{m-1}\binom{m-1}{k}}{\frac{1}{m-1}\binom{m-1}{k-1}}\mathsf{S}^{+\alpha}\mathsf{S}\Delta\boldsymbol{\nu}_{k-1}^{*\alpha} = \frac{m-k}{k}\mathsf{S}^{+\alpha}\mathsf{S}\Delta\boldsymbol{\nu}_{k-1}^{*\alpha} &(4.77)\end{aligned}$$

starting with

$$\Delta\boldsymbol{\nu}_1^{*\alpha} = \Delta\boldsymbol{\nu}^{\alpha} \tag{4.78}$$

Notice that $\Delta\boldsymbol{\nu}_m^{*\alpha} = 0$ or the sum can be stopped at $m-1$.

For XPIC(2):

$$\boldsymbol{\nu}^{*\alpha} = \frac{1}{2}\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{\nu}^{\alpha} \quad \text{and} \quad \Delta\boldsymbol{\nu}^{*\alpha} = 0 \tag{4.79}$$

leading to

$$\mathsf{P}\boldsymbol{V}^{(n)} = 2\mathsf{S}\left(\boldsymbol{\nu}^{\alpha} - \frac{1}{2}\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{\nu}^{\alpha}\right) - \mathsf{S}\Delta\boldsymbol{\nu}^{\alpha} = (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{\nu}^{\alpha} - \mathsf{S}\Delta\boldsymbol{\nu}^{\alpha} \tag{4.80}$$

which agrees with direct derivation.

To fit better with prior code, we incorporate $\Delta\boldsymbol{\nu}_1^{*\alpha} = \Delta\boldsymbol{\nu}^{\alpha}$ into the sum, truncate sum at $m-1$, and finally shift the sum by 1 in k variable:

$$\begin{aligned}(m-1)(\Delta\boldsymbol{\nu}^{\alpha} - \Delta\boldsymbol{\nu}^{*\alpha}) &= (m-1)\sum_{k=1}^{m-1}(-1)^{k+1}\Delta\boldsymbol{\nu}_k^{*\alpha} &(4.81)\\ &= (m-1)\sum_{k=2}^{m}(-1)^{k}\Delta\boldsymbol{\nu}_{k-1}^{*\alpha} &(4.82)\end{aligned}$$

We then redefine the XPIC(m) term as

$$\mathsf{P}\boldsymbol{V}^{(n)} = m\mathsf{S}(\boldsymbol{\nu}^{\alpha} - \boldsymbol{\nu}^{*\alpha}) \tag{4.83}$$

where

$$\boldsymbol{\nu}^{*\alpha} = \sum_{k=2}^{m}(-1)^{k}\left(\boldsymbol{\nu}_k^{*\alpha} + \frac{m-1}{m}\Delta\boldsymbol{\nu}_{k-1}^{*\alpha}\right) \tag{4.84}$$

To implement in code, need to store $\Delta\boldsymbol{\nu}^{\alpha}$ due to constant and then expand to two more XPIC(m) vectors to be able to calculate the new term. The two vectors will hold $\Delta\boldsymbol{\nu}_{k-1}^{*\alpha}$ and $\Delta\boldsymbol{\nu}_k^{*\alpha}$. The total $\boldsymbol{\nu}^{*\alpha}$ can be combined on the fly.

A parity check for XPIC(2) gives

$$\boldsymbol{\nu}^{*\alpha} = \frac{1}{2}\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{\nu}^{\alpha} + \frac{1}{2}\Delta\boldsymbol{\nu}^{\alpha} \tag{4.85}$$

Again:

$$\mathsf{P}\boldsymbol{V}^{(n)} = 2\mathsf{S}\left(\boldsymbol{\nu}^{\alpha} - \frac{1}{2}\mathsf{S}^{+\alpha}\mathsf{S}\boldsymbol{\nu}^{\alpha} - \frac{1}{2}\Delta\boldsymbol{\nu}^{\alpha}\right) = (2\mathsf{I} - \mathsf{S}\mathsf{S}^{+\alpha})\mathsf{S}\boldsymbol{\nu}^{\alpha} - \mathsf{S}\Delta\boldsymbol{\nu}^{\alpha} \tag{4.86}$$

Chapter 5

Extrapolation of Linear and Angular Momentum

5.1 Introduction

This chapter looks at extrapolation of linear and angular momentum to the grid and back and back after an update on the grid. It can consider issues of conservation so momentum. This chapter is for standard GIMP MPM methods. See subsequent chapters to add spin (or maybe more) to the extrapolations.

5.2 Linear Momentum

5.2.1 Initial Extrapolation

At step n, the total momentum on the particles is

$$\boldsymbol{P}_{part}^{(n)} = \sum_p M_p \boldsymbol{V}_p^{(n)} \tag{5.1}$$

After extrapolating to the grid, the total momentum on the grid is

$$\boldsymbol{P}_{grid}^{(n)} = \sum_i \boldsymbol{m}_i^{(n)} \boldsymbol{v}_i^{(n)} = \sum_p \sum_i M_p \boldsymbol{V}_p^{(n)} S_{pi} = \sum_p M_p \boldsymbol{V}_p^{(n)} \tag{5.2}$$

which conserves linear momentum. The grid result can be recast as:

$$\sum_p \sum_i \boldsymbol{m}_i^{(n)} S_{ip}^+ \boldsymbol{V}_p^{(n)} = \sum_p \sum_q \sum_i S_{qi} S_{ip}^+ M_q \boldsymbol{V}_p^{(n)} = \sum_p \sum_q (\mathsf{S}\mathsf{S}^+)_{qp} \boldsymbol{V}_p^{(n)} M_q \tag{5.3}$$

which, by the above results, is equal $\sum_p M_p \boldsymbol{V}_p^{(n)}$. In matrix form, we have

$$\mathsf{S}\mathsf{S}^+ \boldsymbol{V} \cdot \boldsymbol{M} = \mathsf{I} \boldsymbol{V} \cdot \boldsymbol{M} \quad \text{or} \quad (\mathsf{I} - \mathsf{S}\mathsf{S}^+) \boldsymbol{V} \cdot \boldsymbol{M} = 0 \tag{5.4}$$

where $\boldsymbol{M}$ and $\boldsymbol{V}$ are vectors of particle masses and velocities.

5.2.2 Post Update Linear Momentum

After updating velocity on the grid and extrapolating to the particles, the momenta on the grid and particles are

$$\boldsymbol{P}_{grid}^{(n+1)} = \sum_i \boldsymbol{m}_i^{(n)} \tilde{\boldsymbol{v}}_i^{(n+1)} = \boldsymbol{P}_{grid}^{(n)} + \sum_i \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t = \boldsymbol{P}_{grid}^{(n)} + \sum_p M_p \sum_i S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.5}$$

$$\boldsymbol{P}_{part}^{(n+1)} = \sum_p M_p (\boldsymbol{V}_p^{(n)} + \mathbb{A}_p^{(n)} \Delta t) = \boldsymbol{P}_{part}^{(n)} + \sum_p M_p \mathbb{A}_p^{(n)} \Delta t = \boldsymbol{P}_{part}^{(n)} + \sum_p \boldsymbol{F}_p^{(n)} \Delta t \tag{5.6}$$

where assigning particle force to be $\boldsymbol{F}_p^{(n)} = M_p \mathbb{A}_p^{(n)}$ gives expected momentum update on each particle (*i.e.*, $d\boldsymbol{P}_p^{(n)}/dt = \boldsymbol{F}_p^{(n)}$). The change in total momentum between particles and grid after updating is:

$$\Delta \boldsymbol{P} = \sum_p M_p \left(\mathbb{A}_p^{(n)} - \sum_i S_{pi} \boldsymbol{a}_i^{(n)} \right) \Delta t \tag{5.7}$$

An undamped FLIP update (where $\mathbb{A}_p^{(n)} = \sum_i S_{pi} \boldsymbol{a}_i^{(n)}$) conserves linear momentum on each particle. But note, that some might be tempted to update velocity by extrapolating force to the particles and dividing that by particle mass or

$$\mathbb{A}_p^{(n)} = \frac{1}{M_p} \sum_i S_{pi} \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} = \frac{1}{M_p} \sum_i S_{pi} \boldsymbol{f}_i^{(n)} \tag{5.8}$$

This ill-conceived approach would not conserve linear momentum and should therefore not be used. The same is true in transport properties where one must extratrapolate transport value to the particle and *not* extrapolate transport flux to be divided by particle transport capacity.

A FLIP/XPIC(m) update leads to

$$\Delta \boldsymbol{P} = \sum_p M_p (1-\beta) \left(m \left(\boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_{g\to p}^{*(n)} \right) - \boldsymbol{V}_p^{(n)} \right) \tag{5.9}$$

For FLIP ($\beta = 1$) momentum is obviously conserved. Furthermore, it conserves momentum on each particle (*i.e.*, no diffusion of momentum to other particles). After addition of PIC, it is still conserved:

$$\Delta \boldsymbol{P} = \left(m \sum_p M_p \boldsymbol{V}_{g\to p}^{(n)} - m \sum_p M_p \boldsymbol{V}_{g\to p}^{*(n)} - \sum_i \boldsymbol{m}_i^{(n)} \boldsymbol{v}_i^{(n)} \right) \tag{5.10}$$

$$= \sum_i \left[\left(m \sum_p M_p S_{pi} - \boldsymbol{m}_i^{(n)} \right) \boldsymbol{v}_i^{(n)} - m \sum_p M_p S_{pi} \boldsymbol{v}_i^{*} \right] \tag{5.11}$$

$$= \sum_i \sum_p M_p S_{pi} \left((m-1) \boldsymbol{v}_i^{(n)} - m \boldsymbol{v}_i^{*} \right) \tag{5.12}$$

For PIC (or XPIC(1) with $m = 1$ and $\boldsymbol{v}^* = 0$), $\Delta P = 0$ and total momentum is conserved. In other words, adding PIC update conserves total momentum as well, but does not conserve on each particle (only in aggregate), which will result is diffusion of kinetic energy. Each particle will change momentum by $M_p(1-\beta)\left(\boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_p^{(n)}\right)$, but the sum of changes over all particles is zero. For XPIC(m), we can expand the sums to

$$\Delta \boldsymbol{P} = \sum_i \sum_p M_p S_{pi} \left((m-1) \boldsymbol{v}_i^{(n)} - m \sum_{k=2}^{m} (-1)^k \boldsymbol{v}_k^{*} \right) \tag{5.13}$$

$$= \sum_i \sum_p M_p S_{pi} \left((m-1) \boldsymbol{v}_i^{(n)} - \sum_{k=2}^{m} (-1)^k \sum_r \binom{m}{k} (\mathsf{S}^{+}\mathsf{S})_{ir}^{k-1} \boldsymbol{v}_r^{(n)} \right) \tag{5.14}$$

Does this evaluate to zero or not? It may make sense to not be zero and then corresponding to XPIC(m) removing noise and the momentum associated with that noise.

In matrix form using XPIC(m) effective acceleration, the change in momentum is

$$\Delta \boldsymbol{P} = -(1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\boldsymbol{V}\cdot\boldsymbol{M} \tag{5.15}$$

For PIC or XPIC(1), it is zero by results derived above. This may be zero by induction:

1. It works for $m = 1$ (PIC result from above), which further allows

$$0 = \left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)\boldsymbol{V}\cdot\boldsymbol{M} \qquad \text{or} \qquad \mathsf{I}\boldsymbol{V}\cdot\boldsymbol{M} = \mathsf{S}\mathsf{S}^{+}\boldsymbol{V}\cdot\boldsymbol{M} \tag{5.16}$$

2. Assume it holds for m

$$0 = -(1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\boldsymbol{V}\cdot\boldsymbol{M} \tag{5.17}$$

3. Then for $m+1$

$$\begin{aligned}
\Delta \boldsymbol{P} &= -(1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m+1}\boldsymbol{V}\cdot\boldsymbol{M} && (5.18)\\
&= -(1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)\boldsymbol{V}\cdot\boldsymbol{M} && (5.19)\\
&= -(1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\boldsymbol{V}\cdot\boldsymbol{M} + (1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\mathsf{S}\mathsf{S}^{+}\boldsymbol{V}\cdot\boldsymbol{M} && (5.20)\\
&= 0 + (1-\beta)\left(\mathsf{I}-\mathsf{S}\mathsf{S}^{+}\right)^{m}\mathsf{I}\boldsymbol{V}\cdot\boldsymbol{M} && (5.21)\\
&= 0 && (5.22)
\end{aligned}$$

5.2.3 Damping in FLIP

With simple grid damping $\mathbb{A}_p^{(n)} = \boldsymbol{A}_{g\to p}^{(n)} - \alpha_g \boldsymbol{V}_{g\to p}^{(n)}$. The change in momentum after extrapolating is

$$\Delta \boldsymbol{P} = -\sum_p \alpha_g M_p \boldsymbol{V}_{g\to p}^{(n)} \Delta t = -\sum_i \sum_p \alpha_g M_p S_{pi} \boldsymbol{v}_i^{(n)} \Delta t = -\alpha_g \boldsymbol{P}_{grid}^{(n)} \Delta t \tag{5.23}$$

For given damping coefficient, this update causes and exponential decay of momentum with time constant $1/\alpha_g$.

With simple particle damping $\mathbb{A}_p^{(n)} = \boldsymbol{A}_{g\to p}^{(n)} - \alpha_p \boldsymbol{V}_p^{(n)}$. The change in momentum after extrapolating is

$$\Delta \boldsymbol{P} = -\sum_p \alpha_p M_p \boldsymbol{V}_p^{(n)} \Delta t = -\alpha_p \boldsymbol{P}_{part}^{(n)} \Delta t \tag{5.24}$$

When held, this causes and exponential decay of momentum with time constant $1/\alpha_p$.

Note that PIC damping (my term) conserves total momentum because both FLIP and PIC conserve it.

5.3 Angular Momentum

5.3.1 Initial Extrapolation

At step n, the total angular momentum referenced to the origin on the particles is

$$\boldsymbol{L}_{part}^{(n)} = \sum_p \boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} \tag{5.25}$$

After extrapolating to the grid, the total angular momentum on the grid is

$$\begin{aligned} \boldsymbol{L}_{grid}^{(n)} &= \sum_i \boldsymbol{x}_i \times \boldsymbol{p}_i^{(n)} = \sum_p \sum_i \boldsymbol{x}_i \times M_p \boldsymbol{V}_p^{(n)} S_{pi} = \sum_p \left(\sum_i \boldsymbol{x}_i S_{pi} \right) \times M_p \boldsymbol{V}_p^{(n)} && (5.26) \\ &= \sum_p \boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} && (5.27) \end{aligned}$$

Thus angular momentum is conserved (and choice of origin is arbitrary).

5.3.2 Post Update Angular Momentum

After updating velocity on the grid and extrapolating to the particles, the angular momentum on the particles is

$$\begin{aligned} \boldsymbol{L}_{part}^{(n+1)} &= \sum_p \left(\boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n)} \Delta t + \frac{1}{2} \mathbb{A}_p^{(n)} (\Delta t)^2 \right) \times M_p \left(\boldsymbol{V}_p^{(n)} + \mathbb{A}_p^{(n)} \Delta t \right) && (5.28) \\ &= \boldsymbol{L}_{part}^{(n)} + \sum_p M_p \boldsymbol{X}_p^{(n)} \times \mathbb{A}_p^{(n)} \Delta t - \sum_p M_p \boldsymbol{V}_p^{(n)} \times \mathbb{V}_p^{(n)} \Delta t \\ &\qquad + \sum_p M_p \mathbb{V}_p^{(n)} \Delta t \times \mathbb{A}_p^{(n)} \Delta t - \frac{1}{2} \sum_p M_p \boldsymbol{V}_p^{(n)} \Delta t \times \mathbb{A}_p^{(n)} \Delta t && (5.29) \\ &= \boldsymbol{L}_{part}^{(n)} + \sum_p M_p \left[\left(\boldsymbol{X}_p^{(n)} + \left(\mathbb{V}_p^{(n)} - \frac{\boldsymbol{V}_p^{(n)}}{2} \right) \Delta t \right) \times \mathbb{A}_p^{(n)} \Delta t - \boldsymbol{V}_p^{(n)} \Delta t \times \mathbb{V}_p^{(n)} \right] && (5.30) \\ &= \boldsymbol{L}_{part}^{(n)} + \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \mathbb{A}_p^{(n)} + \left(\mathbb{V}_p^{(n)} - \frac{\boldsymbol{V}_p^{(n)}}{2} \right) \times \mathbb{A}_p^{(n)} \Delta t - \boldsymbol{V}_p^{(n)} \times \mathbb{V}_p^{(n)} \right] \Delta t && (5.31) \end{aligned}$$

The angular momentum on the grid after updating and accounting for movement of nodes and updated nodal velocities is:

$$\begin{aligned} \boldsymbol{L}_{grid}^{(n+1)} &= \sum_i \left(\boldsymbol{x}_i + \boldsymbol{v}_i^{(n)} \Delta t + \frac{1}{2} \boldsymbol{a}_i^{(n)} (\Delta t)^2 \right) \times \boldsymbol{m}_i^{(n)} \left(\boldsymbol{v}_i^{(n)} + \boldsymbol{a}_i^{(n)} \Delta t \right) && (5.32) \\ &= \boldsymbol{L}_{grid}^{(n)} + \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t + \sum_i \boldsymbol{v}_i^{(n)} \Delta t \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t \\ &\qquad + \sum_i \frac{1}{2} \boldsymbol{a}_i^{(n)} (\Delta t)^2 \times \boldsymbol{m}_i^{(n)} \boldsymbol{v}_i^{(n)} && (5.33) \\ &= \boldsymbol{L}_{grid}^{(n)} + \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t + \frac{1}{2} \sum_i \boldsymbol{v}_i^{(n)} \Delta t \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t && (5.34) \\ &= \boldsymbol{L}_{grid}^{(n)} + \sum_i \left(\boldsymbol{x}_i + \frac{1}{2} \boldsymbol{v}_i^{(n)} \Delta t \right) \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t && (5.35) \\ &= \boldsymbol{L}_{grid}^{(n)} + \sum_i \left(\boldsymbol{x}_i + \frac{1}{2} \Delta \boldsymbol{x}_i \right) \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t && (5.36) \\ &= \boldsymbol{L}_{grid}^{(n)} + \sum_p M_p \left[\sum_i \left(\boldsymbol{x}_i + \frac{1}{2} \Delta \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \right] \Delta t && (5.37) \end{aligned}$$

Equation (5.36) used

$$\frac{1}{2}\boldsymbol{v}_i^{(n)}\Delta t \times \boldsymbol{m}_i^{(n)}\boldsymbol{a}_i^{(n)}\Delta t = \left(\frac{1}{2}\boldsymbol{v}_i^{(n)}\Delta t + \frac{1}{4}\boldsymbol{a}_i^{(n)}(\Delta t)^2\right) \times \boldsymbol{m}_i^{(n)}\boldsymbol{a}_i^{(n)}\Delta t = \frac{1}{2}\Delta \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)}\boldsymbol{a}_i^{(n)}\Delta t \tag{5.38}$$

where

$$\Delta \boldsymbol{x}_i = \boldsymbol{v}_i^{(n)}\Delta t + \frac{1}{2}\boldsymbol{a}_i^{(n)}(\Delta t)^2 \tag{5.39}$$

It could alternatively use

$$\sum_i \frac{1}{2}\boldsymbol{v}_i^{(n)}\Delta t \times \boldsymbol{m}_i^{(n)}\boldsymbol{a}_i^{(n)}\Delta t = \sum_p \sum_i \frac{1}{2} M_p \boldsymbol{V}_p^{(n)}\Delta t \times S_{pi}\boldsymbol{a}_i^{(n)}\Delta t = \sum_p M_p \boldsymbol{V}_p^{(n)} \times \frac{1}{2}\boldsymbol{A}_{g\to p}^{(n)}(\Delta t)^2 \tag{5.40}$$

The change in angular momentum (particles minus grid) after updating (using Eqs. (5.31) and (5.37) and adding and subtracting acceleration in two forms or $\boldsymbol{X}_p^{(n)} \times \boldsymbol{A}_{g\to p}^{(n)} = \sum_i \boldsymbol{X}_p^{(n)} \times S_{pi}\boldsymbol{a}_i^{(n)}$) is:

$$\begin{aligned}
\frac{\Delta \boldsymbol{L}}{\Delta t} &= \sum_p M_p \Bigg[\boldsymbol{X}_p^{(n)} \times \left(\mathbb{A}_p^{(n)} - \boldsymbol{A}_{g\to p}^{(n)}\right) + \left(\mathbb{V}_p^{(n)} - \frac{\boldsymbol{V}_p^{(n)}}{2}\right) \times \mathbb{A}_p^{(n)}\Delta t - \boldsymbol{V}_p^{(n)} \times \mathbb{V}_p^{(n)} \\
&\qquad + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i - \frac{1}{2}\Delta \boldsymbol{x}_i\right) \times S_{pi}\boldsymbol{a}_i^{(n)} \Bigg] \qquad (5.41) \\
&= \sum_p M_p \Bigg[\boldsymbol{X}_p^{(n)} \times \left(\mathbb{A}_p^{(n)} - \boldsymbol{A}_{g\to p}^{(n)}\right) + \left(\mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)}\right) \times \mathbb{V}_p^{(n+1)} + \frac{\boldsymbol{V}_p^{(n)}}{2} \times \mathbb{A}_p^{(n)}\Delta t \\
&\qquad + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i - \frac{1}{2}\Delta \boldsymbol{x}_i\right) \times S_{pi}\boldsymbol{a}_i^{(n)} \Bigg] \qquad (5.42)
\end{aligned}$$

Using Eq. (2.10), we have

$$\begin{aligned}
&\sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\mathbb{A}_p^{(n)} - \boldsymbol{A}_{g\to p}^{(n)}\right)\Delta t + \left(\mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)}\right)\Delta t \times \mathbb{V}_p^{(n+1)} + \boldsymbol{V}_p^{(n)} \times \frac{1}{2}\mathbb{A}_p^{(n)}(\Delta t)^2 \right] \qquad (5.43) \\
= &\sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n+1)} + \boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_p^{(n)}\right) + \left(\mathbb{V}_p^{(n+1)} + \frac{\boldsymbol{V}_p^{(n)}}{2}\right)\Delta t \times \boldsymbol{V}_p^{(n+1)} \right] \qquad (5.44) \\
= &\sum_p M_p \left[\left(\boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n+1)}\Delta t + \frac{\boldsymbol{V}_p^{(n)}}{2}\Delta t\right) \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_{g\to p}^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n)} + \boldsymbol{V}_p^{(n)}\right) \right] \qquad (5.45) \\
= &\sum_p M_p \left[\left(\boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n+1)}\Delta t + \frac{\boldsymbol{V}_p^{(n)} - \boldsymbol{V}_p^{(n+1)}}{2}\Delta t\right) \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_{g\to p}^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n)} + \boldsymbol{V}_p^{(n)}\right) \right] \\
= &\sum_p M_p \left[\boldsymbol{X}_p^{(n+1)} \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_{g\to p}^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n)} + \boldsymbol{V}_p^{(n)}\right) \right] \qquad (5.46)
\end{aligned}$$

The angular momentum change becomes

$$\begin{aligned}\Delta \boldsymbol{L} &= \sum_p M_p \Bigg[\boldsymbol{X}_p^{(n+1)} \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_{g\to p}^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n)} + \boldsymbol{V}_p^{(n)} \right) \\ &\qquad + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i - \frac{1}{2}\Delta \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \qquad (5.47) \\ &= \sum_p M_p \Bigg[\boldsymbol{X}_p^{(n+1)} \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \boldsymbol{V}_p^{(n)} - \sum_i \boldsymbol{X}_p^{(n)} \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \\ &\qquad + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i - \frac{1}{2}\Delta \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \qquad (5.48) \\ &= \sum_p M_p \Bigg[\boldsymbol{X}_p^{(n+1)} \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \boldsymbol{V}_p^{(n)} - \sum_i \left(\boldsymbol{x}_i + \frac{1}{2}\Delta \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \qquad (5.49)\end{aligned}$$

Invoking conservation of angular momentum the total angular momentum on the particle after updating particle position and velocity becomes

$$\sum_p \boldsymbol{X}_p^{(n+1)} \times M_p \boldsymbol{V}_p^{(n+1)} = \sum_p \Bigg[\boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} + \sum_i \left(\boldsymbol{x}_i + \frac{1}{2}\Delta \boldsymbol{x}_i \right) \times M_p S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \qquad (5.50)$$

which appears physically correct — new angular momentum initial momentum plus torque around the origin applied to point masses projected from particle to the nodes. But, unlike linear momentum where we can pick acceleration to conserve momentum, we have no quantity to pick for angular momentum. The equation seems to say that MPM conserves angular momentum if we assume zero particle rotation on its own axis and we assume constant velocity of each particle.

Or perhaps it should be written as

$$\begin{aligned}\sum_p \boldsymbol{X}_p^{(n+1)} \times M_p \boldsymbol{V}_p^{(n+1)} &= \sum_p \Bigg[\boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} + \sum_i \boldsymbol{X}_p^{(n)} \times M_p S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \\ &\qquad + \sum_i \left(\boldsymbol{x}_i + \frac{1}{2}\Delta \boldsymbol{x}_i - \boldsymbol{X}_p^{(n)} \right) \times M_p S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \qquad (5.51)\end{aligned}$$

The first two terms on the right are the initial angular momentum of the particles plus torque on the particles caused by nodal accelerations. The last term is a torque about the center of mass that would change the particle angular momentum due to rotation on its axis. In this form, the rotational contribution to angular momentum is lost unless the simulation tracks particle angular momentum. One approach is to add particle spin (but maybe better described as adding particle angular momentum) as described in Chapter **??**.

Alternate Forms for Particle Angular Momentum

An alternate form (using $\mathbb{A}_p^{(n)} \Delta t = \mathbb{V}_p^{(n+1)} - \mathbb{V}_p^{(n)} = \boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_p^{(n)}$, see Eq. (2.10)) is

$$\boldsymbol{L}_{part}^{(n+1)} = \boldsymbol{L}_{part}^{(n)} + \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \mathbb{A}_p^{(n)} + \frac{1}{2} \left(\left(\mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)} \right) \times \mathbb{V}_p^{(n+1)} + \mathbb{V}_p^{(n)} \times \boldsymbol{V}_p^{(n+1)} \right) \right] \Delta t \qquad (5.52)$$

If we take force on each particle used in momentum update, $\boldsymbol{F}_p = M_p \mathbb{A}_p^{(n)}$, to now apply a torque on each particle over the time step, we get

$$\boldsymbol{\tau}_p^{(n)} \Delta t = \int_0^{\Delta t} \left(\boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n)} t + \frac{1}{2} \mathbb{A}_p^{(n)} (t)^2 \right) \times M_p \mathbb{A}_p^{(n)} \, dt = \left(\boldsymbol{X}_p^{(n)} + \mathbb{V}_p^{(n)} \frac{\Delta t}{2} \right) \times M_p \mathbb{A}_p^{(n)} \Delta t \tag{5.53}$$

Substituting into particle angular momentum (Eq. (5.31)) gives

$$\begin{aligned} \boldsymbol{L}_{part}^{(n+1)} &= \boldsymbol{L}_{part}^{(n)} + \sum_p \left[\boldsymbol{\tau}_p^{(n)} \Delta t + M_p \left(\mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)} \right) \times \left(\mathbb{V}_p^{(n)} \Delta t + \mathbb{A}_p^{(n)} \frac{(\Delta t)^2}{2} \right) \right] & (5.54) \\ &= \boldsymbol{L}_{part}^{(n)} + \sum_p \left[\boldsymbol{\tau}_p^{(n)} \Delta t - \left(\boldsymbol{X}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \right) \times M_p \left(\mathbb{V}_p^{(n)} - \boldsymbol{V}_p^{(n)} \right) \right] & (5.55) \end{aligned}$$

The first term is expected torque update for each particle (*i.e.*, $d\boldsymbol{L}_p^{(n)}/dt = \boldsymbol{\tau}_p^{(n)}$). The second term is some extra angular momentum, perhaps due to movement of the particles.

Alternate Form in PIC Style

An alternate form for some terms (derived from initial angular momentum) uses

$$\begin{aligned} \sum_p \boldsymbol{X}_p^{(n)} \times M_p \left(\boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_p^{(n)} \right) &= \sum_p \sum_i \boldsymbol{X}_p^{(n)} \times M_p S_{pi} \boldsymbol{v}_i^{(n)} - \sum_p \boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} \\ &= \sum_p \sum_i \boldsymbol{X}_p^{(n)} \times M_p S_{pi} \boldsymbol{v}_i^{(n)} - \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{v}_i^{(n)} \\ &= \sum_p \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times M_p S_{pi} \boldsymbol{v}_i^{(n)} \end{aligned} \tag{5.56}$$

This result that assumes particle has no spin (will need to revise when add spin (*i.e.*, add $(1-\beta)\mathsf{I}_p^{(n)} \boldsymbol{\omega}_p^{(n)}$)). The change in angular momentum can also be

$$\begin{aligned} \Delta \boldsymbol{L} = \sum_p M_p \Bigg[& \boldsymbol{X}_p^{(n+1)} \times \boldsymbol{V}_p^{(n+1)} - \boldsymbol{X}_p^{(n)} \times \boldsymbol{V}_{g\to p}^{(n+1)} \\ & + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} - \frac{1}{2} \sum_i \Delta \boldsymbol{x}_i \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \Bigg] \end{aligned} \tag{5.57}$$

For PIC case with $\boldsymbol{V}_p^{(n+1)} = \boldsymbol{V}_{g\to p}^{(n+1)}$ gives

$$\Delta \boldsymbol{L} = \sum_p M_p \left[\sum_i \left(\boldsymbol{X}_p^{(n+1)} - \boldsymbol{x}_i \right) \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} - \frac{1}{2} \sum_i \Delta \boldsymbol{x}_i \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \tag{5.58}$$

5.3.3 Low Order Post Update Angular Momentum

After updating velocity on the grid and extrapolating to the particles, the angular momentum on the particles that ignores position change (as seems to be done in RPIC paper) is

$$\boldsymbol{L}_{part}^{(n+1)} = \sum_p \boldsymbol{X}_p^{(n)} \times M_p \left(\boldsymbol{V}_p^{(n)} + \mathbb{A}_p^{(n)} \Delta t \right) = \boldsymbol{L}_{part}^{(n)} + \sum_p M_p \boldsymbol{X}_p^{(n)} \times \mathbb{A}_p^{(n)} \Delta t \tag{5.59}$$

The angular momentum on the grid after updating (and ignoring nodal position change is

$$\boldsymbol{L}_{grid}^{(n+1)} = \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \left(\boldsymbol{v}_i^{(n)} + \boldsymbol{a}_i^{(n)} \Delta t \right) = \boldsymbol{L}_{grid}^{(n)} + \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.60}$$

$$= \boldsymbol{L}_{grid}^{(n)} + \sum_i \sum_p M_p \boldsymbol{S_{pi}} \boldsymbol{x}_i \times \boldsymbol{a}_i^{(n)} \Delta t = \boldsymbol{L}_{grid}^{(n)} + \sum_p M_p \sum_i \boldsymbol{x}_i \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.61}$$

The change in angular momentum (particles minus grid) is

$$\Delta \boldsymbol{L} = \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \mathbb{A}_p^{(n)} \Delta t - \sum_i \boldsymbol{x}_i \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \tag{5.62}$$

$$= \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\mathbb{A}_p^{(n)} - \boldsymbol{A}_{g \to p}^{(n)} \right) \Delta t + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \tag{5.63}$$

For FLIP, the change is

$$\Delta \boldsymbol{L} = \sum_p M_p \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.64}$$

$$= \sum_p \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times M_p S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.65}$$

For PIC, it is

$$\Delta \boldsymbol{L} = \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_{g \to p}^{(n)} - \boldsymbol{V}_p^{(n)} \right) + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \tag{5.66}$$

$$= \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \sum_i S_{pi} (\boldsymbol{v}_i^{(n)} + \boldsymbol{a}_i^{(n)} \Delta t) - \boldsymbol{X}_p^{(n)} \times \boldsymbol{V}_p^{(n)} - \sum_i \boldsymbol{x}_i \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \tag{5.67}$$

$$= \sum_p \sum_i M_p \boldsymbol{X}_p^{(n)} \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} - \sum_p \boldsymbol{X}_p^{(n)} \times M_p \boldsymbol{V}_p^{(n)} - \sum_i \boldsymbol{x}_i \times \sum_p M_p S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.68}$$

$$= \sum_p \sum_i M_p \boldsymbol{X}_p^{(n)} \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} - \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{v}_i^{(n)} - \sum_i \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \boldsymbol{a}_i^{(n)} \Delta t \tag{5.69}$$

$$= \sum_i \left[\sum_p \boldsymbol{X}_p^{(n)} \times M_p S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} - \boldsymbol{x}_i \times \boldsymbol{m}_i^{(n)} \tilde{\boldsymbol{v}}_i^{(n)} \right] \tag{5.70}$$

$$= \sum_i \left(\sum_p M_p S_{pi} \boldsymbol{X}_p^{(n)} - \boldsymbol{m}_i^{(n)} \boldsymbol{x}_i \right) \times \tilde{\boldsymbol{v}}_i^{(n)} \tag{5.71}$$

$$= \sum_p \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times M_p S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} \tag{5.72}$$

Note that one step used a result that assumes particle has no spin (will need to revise when add spin, *i.e.*, add $(1-\beta)\mathsf{I}_p^{(n)} \boldsymbol{\omega}_p^{(n)}$). This result agrees with the Disney/UCLA result for RPIC; hence that method is low order approach.

The change in angular momentum using identities in previous section is:

$$\Delta \boldsymbol{L} = \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n+1)} + \boldsymbol{V}_{g\to p}^{(n)} - \boldsymbol{V}_p^{(n)} \right) + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \boldsymbol{a}_i^{(n)} \Delta t \right] \quad (5.73)$$

$$= \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \left(\boldsymbol{V}_p^{(n+1)} - \boldsymbol{V}_{g\to p}^{(n+1)} \right) + \sum_i \left(\boldsymbol{X}_p^{(n)} - \boldsymbol{x}_i \right) \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} \right] \quad (5.74)$$

$$= \sum_p M_p \left[\boldsymbol{X}_p^{(n)} \times \boldsymbol{V}_p^{(n+1)} - \sum_i \boldsymbol{x}_i \times S_{pi} \tilde{\boldsymbol{v}}_i^{(n)} \right] \quad (5.75)$$