Transferring Momentum as an Element Field

Dan Ibanez

September 17, 2016

The integral of momentum in a mesh comprised of d-dimensional simplices, assuming piecewise linear velocity and piecewise constant densities, is:

$$\mathbf{p}_\Omega = \sum_{e \in E} m_e \frac{1}{d+1} \sum_{a \in \mathcal{N}(e)} v_a \tag{1}$$

Ω	mesh
$\mathbf{p}_\Omega$	integral of momentum over the mesh
E	set of elements in the mesh
m_e	mass of element e
d	dimension of simplex elements
$\mathcal{N}(e)$	nodes of element e
v_a	velocity of node a

We can just as easily rearrange the sums to be over nodes:

$$\mathbf{p}_\Omega = \frac{1}{d+1} \sum_{a \in \mathcal{N}} v_a \sum_{e \in E(a)} m_e \tag{2}$$

$E(a)$	set of elements sharing node a

In order to transfer velocity such that the integral of momentum is conserved, we can start by first assigning to each element the integral of momentum over it, following Equation 1:

$$\forall e \in E_D : \mathbf{p}_e = m_e \frac{1}{d+1} \sum_{a \in \mathcal{N}_D(e)} v_a \tag{3}$$

Ω_D	"donor" mesh
E_D	elements in donor mesh
$\mathbf{p}_e$	momentum assigned to element e

We then fully adapt the mesh while transferring this momentum field $\mathbf{p}_e$ as an element-wise field in a conservative manner, as we do with element mass and energy:

$$\sum_{e \in E_T} \mathbf{p}_e = \sum_{e \in E_D} \mathbf{p}_e \tag{4}$$

E_T	elements in “target” mesh

Afterwards, we can project momentum back to the nodes such that the sum of the nodal momentum values equals the sum of the element momentum values:

$$\forall a \in \mathcal{N}_T : \mathbf{p}_a = \frac{1}{d+1} \sum_{e \in E_T(a)} \mathbf{p}_e \tag{5}$$

$\mathcal{N}_T$	nodes in target mesh
$E_T(a)$	target elements sharing node a

$$\sum_{a \in \mathcal{N}_T} \mathbf{p}_a = \sum_{e \in E_T} \mathbf{p}_e \tag{6}$$

E_T	elements in target mesh
$\mathbf{p}_a$	momentum assigned to node a

After that we can derive nodal velocities based on the single terms of Equation 2:

$$\forall a \in \mathcal{N}_T : v_a = \frac{(d+1)\mathbf{p}_a}{\sum_{e \in E_T(a)} m_e} \tag{7}$$

Putting Equations 5 and 7 together gives:

$$\forall a \in \mathcal{N}_T : v_a = \frac{\sum_{e \in E_T(a)} \mathbf{p}_e}{\sum_{e \in E_T(a)} m_e} \tag{8}$$

If we subsitute the new nodal velocity from Equation 8 back into the integral Equation 2, we see that the momentum of the target mesh is:

$$\mathbf{p}_{\Omega_T} = \frac{1}{d+1} \sum_{a \in \mathcal{N}_T} \frac{\sum_{e \in E_T(a)} \mathbf{p}_e}{\sum_{e \in E_T(a)} m_e} \sum_{e \in E_T(a)} m_e = \frac{1}{d+1} \sum_{a \in \mathcal{N}_T} \sum_{e \in E_T(a)} \mathbf{p}_e \tag{9}$$

Which is exactly the sum of the nodal momentum values from Equation 5, so by Equations 6, 4, 3 and 1, the target mesh momentum is exactly the donor mesh momentum.

Some of the reasons to recommend this method include:

1. It exactly conserves the integral of momentum over the mesh
2. It reuses existing code to transfer piecewise constant fields, so no new logic is needed inside the mesh adaption code.
3. The only new code required is in Equations 3 and Equations 8, which are fairly trivial and fast.
4. It doesn't require 2-layer cavities, the use of which slows down our adaptive code by a factor 3X overall in MPI parallel mode.
5. It doesn't require solving a linear or nonlinear system at each cavity, which would also slow things down.
6. The element-wise transfer that satisfies Equation 4 can be minimally diffusive, especially if the element-intersection algorithm is used. Equation 3 is not diffusive at all, and Equation 8 should not diffuse momentum that much, hence there is a good chance that diffusion will not be a serious issue.