

# Implementation of Crack and Material Velocity Fields with Contact and Nodal Velocity Boundary Conditions

John Nairn

September 16, 2013

## 1 Velocity Fields

Each node has one to four crack velocity fields denoted [0], [1], [2], and [3]. If the problem has no cracks, then each node has just the single crack velocity field [0]. Each crack velocity field has any number of material velocity fields or just one velocity field if using single material mode.

### 1.1 Crack Velocity Fields

The crack velocity field is determined by drawing a line from the particle to the node. If the line crosses no cracks, that material point uses field [0]. If the line crosses one crack, that material point uses field [1] (for the first crack found) or field [2] (if a second crack is found for the same node). If the line crosses two cracks, the material point uses field [3]. This scheme can handle one or two cracks at each node. A third crack (*i.e.*, a line that crosses a third crack or one of the cracks in field [3] being different than the cracks for fields [1] and [2]) is handled by picking one of the four fields, but is likely to not get the physics correct. A warning is issued if a third crack is found.

Figure 1 shows schematic view of crack velocity fields for the indicated nodal point. Because field [2] is never allocated before field [1] (*i.e.*, the first single crack crossing is put into field [1] and the second into field [2]), the following crack field situations are all that are possible: [0], [1], [3], [0]&[1], [0]&[3], [1]&[2], [1]&[3], [0]&[1]&[2], [0]&[1]&[3], [1]&[2]&[3], and [0]&[1]&[2]&[3]. The combinations that never occur are the remaining ones or [2], [0]&[2], [2]&[3], and [0]&[2]&[3].

When doing crack contact, the possible field contacts are between [0]&[1] and [2]&[3] on the crack #1 surface and between [0]&[2] and [1]&[3] on the crack #2 surface. Fields [0]&[3] and [1]&[2] do not contact each other.

### 1.2 Material Velocity Fields

Each crack velocity field may have 1 to  $n$  material velocity fields where  $n$  is the number of materials assigned to material points. For a single velocity field,  $n$  is set to 1 and all materials use the same velocity field.

## 2 MPM Time Step Tasks

This section lists each task in NairnMPM outlining only those parts relevant to crack and material velocity fields, to contact conditions, to imperfect interface and traction laws on cracks, and to nodal velocity boundary conditions.

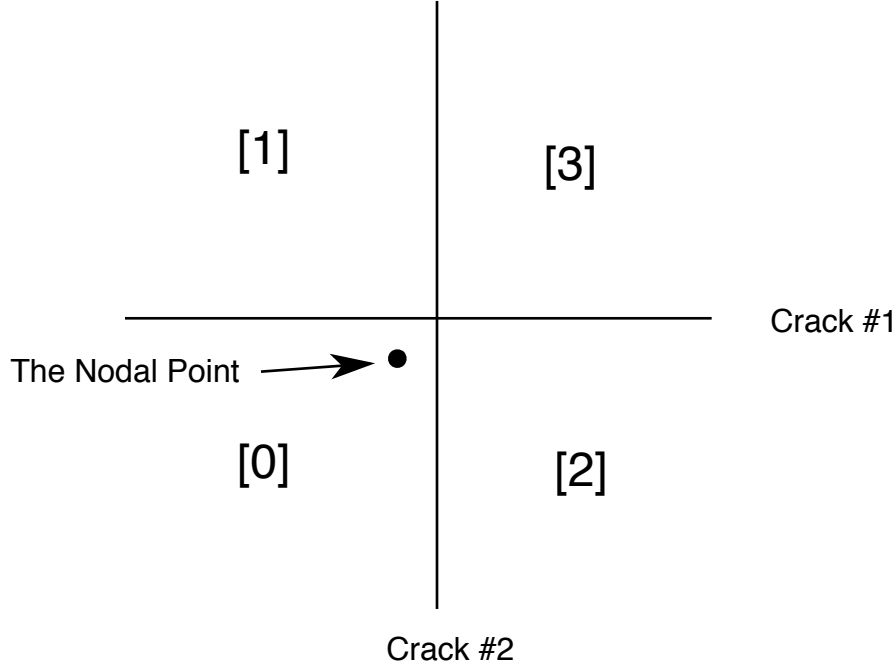


Figure 1: Schematic view of crack velocity fields around a single node. The possibility of two interacting cracks leads to at most four velocity fields.

## 2.1 Task 0: Initialization

Each nodal point is permanently assigned field [0] and that field is zeroed. Any material velocity fields in field [0] that were used in the last step are also zeroed. If the previous step allocated fields [1], [2], or [3], they are deleted.

For each material point, find the crack velocity field (`vfld` = 0, 1, 2, or 3) by checking the line crossings of all cracks (if there are no cracks `vfld` is always 0). This task was moved to initialization when code went parallel. It is one of the largest tasks that is not parallelized.

## 2.2 Task 1: Project Momentum and Mass to the Grid

For each material point, look for the crack velocity field and determine the appropriate material field (`matfld`) from the material assigned to that material point. Add the momentum and mass to material velocity field `matfld` of crack velocity field `vfld`. For crack contact and imperfect interface calculations, project the displacement (or position) and undeformed volume for crack velocity field `vfld` to the grid. This displacement and undeformed volume treats all materials in the crack velocity field as a single material velocity field.

## 2.3 Task 2: Contact and Optional Update Strains First

Adjust momenta at all overlapping material velocity fields in each crack velocity field using the material-material contact law. Next adjust momenta for crack contact conditions. The crack contact calculations use the center-of-mass material velocity field for that crack velocity field. Then, for each node with a fixed velocity on any component of velocity, copy all velocity fields

(these are denoted as  $p_{int}$  or the initially interpolated velocity and will be needed in task 3) and then set the corresponding component to the prescribed momentum.

Note: I used to prescribe boundary conditions before doing contact, but now do the contact calculations first. The logic being that the projected velocity fields before prescribing boundary conditions for multiple velocity fields with cracks in which contact is done by stick conditions should be identical to those for a single velocity field/no-crack calculation. Doing contact before boundary conditions achieves this goal.

When updating strains first (or first and last), calculate velocities in all velocity fields and update the strains and stresses.

## 2.4 Task 3: Project Forces to the Grid

For each material point, project  $f_{int}$  and  $f_{ext}$  to the grid. These are added to material velocity field `matfld` of crack velocity field `vfld`, where `vfld` was found in task and `matfld` is known by the material point's material type.

For all cracks with traction laws, add  $f_{ext}$  for the traction. For all cracks with imperfect interfaces, add  $f_{int}$  due to interfacial displacements (note: it might be possible to do material contact by an interface law as well).

Finally, revisit all nodes with velocity boundary condition and change to total force to the force required to convert  $p_{int}$  to the boundary condition momentum  $p_{BC}$  after the task 4 momentum update using the relation:

$$f_{tot} = \frac{p_{BC} - p_{int}}{\Delta t} \quad (1)$$

where  $\Delta t$  is the time step. This step is essential to get consistent forces and momenta, which are converted to accelerations and velocities during the particle update in task 5. Moreover, if force or momentum are changed again between this task and the particle update, they *must* be changed together to remain consistent.

## 2.5 Task 4: Update Nodal Momenta

This task updates the nodal momenta using

$$\vec{p}_{k+1} = \vec{p}_k + f_{tot}\Delta t \quad (2)$$

Now, depending on the problem, this update may cause more contact. Thus the update must be followed by adjusting momenta at all nodes with overlapping materials and at all cracks now in contact. The logic again is that the final momenta in this step should be identical when doing multiple velocity fields with stick contact to the momenta when using a single velocity field.

Two details are important. First, because of the need to keep momenta and forces consistent until the particle update in task 5, any time the momentum is changed by  $\Delta\vec{p}$ , the corresponding force must be changed by  $\Delta\vec{p}/\Delta t$ .

Second, the momentum component for any node that fixes that component with a velocity boundary condition must not be changed. If boundary conditions are applied to all velocity fields, this situation may never arise because the boundary conditions should insure the momentum update does not induce new contact (*i.e.*, the velocity fields will be moving at the same velocity).

## 2.6 Task 5: Update the Particles

Update particle position and velocity using nodal velocity (from momentum) and acceleration (from  $f_{tot}$ ) found in material velocity field `matfld` of crack velocity field `vfld`.

## 2.7 Task 6: Optional Update Strain Last

My reading of the Sulsky, Zhou, Scherer paper that claimed to improve the original Sulsky paper by dealing with modal momentum rather than nodal velocities, is really that the algorithm projects momenta to the grid a second time prior to updating strains. Thus, this task rezero's all velocity field momenta, loops over particles, and projects to the grid. The projection uses the pre-calculated material velocity field `matfld` of crack velocity field `vfld`. For crack contact calculations, also project the displacements for each crack velocity field. Because this projection might induce new contact, adjust momenta again at nodes with multiple materials and at crack surfaces. Again, the logic is make sure stick conditions are identical to single velocity field and a single velocity field would not permit overlap at this stage.

Finally, impose boundary conditions, calculate nodal velocities (from the final momenta), and update stresses and strains.

## 2.8 Task 7: Custom Tasks

Nothing here is related to contact or to velocity boundary conditions.

## 2.9 Task 8: Move Cracks

The crack surfaces and crack plane are updated using the center-of-mass velocity field of each crack velocity field (for those that have multiple materials).

# 3 Contact Forces and Changes in Momentum

Once contact is detected, the key task is to change the momenta for the various fields to implement stick or frictional sliding. Consider two materials,  $a$  and  $b$ , with nodal velocities on the same node of  $\vec{v}_a$  and  $\vec{v}_b$  and define a normal vector,  $\hat{n}$ , that is positive when directed from material  $a$  to  $b$ . The center of mass velocity is

$$\vec{v}_c = \frac{m_a \vec{v}_a + m_b \vec{v}_b}{m_a + m_b} \quad (3)$$

where  $m_a$  and  $m_b$  are the masses for the two materials. Next, define  $\Delta \vec{p}_a$  as the momentum change required on material  $a$  for its velocity to change from  $\vec{v}_a$  to  $\vec{v}_c$  or

$$\vec{v}_a + \frac{\Delta \vec{p}_a}{m_a} = \vec{v}_c \quad \text{which means} \quad \Delta \vec{p}_a = -m_a(\vec{v}_a - \vec{v}_c) \quad (4)$$

Inserting the definition of  $\vec{v}_c$  leads to

$$\Delta \vec{p}_a = -m_a(\vec{v}_a - \vec{v}_c) = m_b(\vec{v}_b - \vec{v}_c) = \frac{m_a \vec{p}_b - m_b \vec{p}_a}{m_a + m_b} \quad (5)$$

where  $\vec{p}_a$  and  $\vec{p}_b$  are the nodal momenta for the two materials. Notice that

$$\Delta \vec{v} = \vec{v}_b - \vec{v}_a = \frac{\Delta \vec{p}_a}{m_b} + \frac{\Delta \vec{p}_a}{m_a} = \frac{\Delta \vec{p}_a}{m_{red}} \quad (6)$$

where  $m_{red} = m_a m_b / (m_a + m_b)$  is the reduced nodal mass. Thus the relative velocity of approach of the two surfaces is proportional to, and in the same direction as,  $\Delta \vec{p}_a$ . However contact is detected, one of the criteria should be that the two surfaces are approaching each other. With the above

definition for the one normal, this criteria implies  $\Delta\vec{v} \cdot \hat{n} < 0$  or identically that  $\Delta\vec{p}_a \cdot \hat{n} < 0$ . We thus assume that under all contact situations that  $\Delta\vec{p}_a \cdot \hat{n} < 0$ .

In simulations with more than two materials, a single node may have velocity fields from three or more materials. This situation may mean contact is not handled well (*i.e.*, that resolution needs to be increased until no nodes see three or more materials, if possible) or may mean an unavoidable situation (*e.g.*, meeting at corners of materials). The code needs to proceed as well as it can. The approach is to take the center of mass results

$$\vec{p}_c = \sum_i \vec{p}_i \quad m_c = \sum_i m_i \quad \vec{v}_c = \frac{\vec{p}_c}{m_c} \quad (7)$$

and define a *virtual* material  $b$  derived from all mass and momentum not associated with material  $a$ :

$$\vec{p}_b = \vec{p}_c - \vec{p}_a \quad m_b = m_c - m_a \quad \vec{v}_b = \frac{\vec{p}_b}{m_b} \quad (8)$$

The equations for  $\Delta\vec{p}_a$  and  $\Delta\vec{v}$  are unchanged, but in terms of just  $a$  and  $c$ , they become

$$\Delta\vec{p}_a = \frac{m_a}{m_c} \vec{p}_c - \vec{p}_a \quad \Delta\vec{v} = \frac{m_c}{m_a(m_c - m_a)} \Delta\vec{p}_a \quad (9)$$

To implement contact conditions, with inclusion of nodes with three or more materials, each material is considered relative to the center of mass velocity and  $\Delta\vec{p}_a$  and  $\Delta\vec{v}$  are found. Stick conditions are trivial. The momentum change of  $\Delta\vec{p}_a$  is applied to material  $a$ . These momenta changes correspond to normal and tangential forces of

$$\vec{f}_n = \frac{(\Delta\vec{p}_a \cdot \hat{n})}{\Delta t} \hat{n} \quad \text{and} \quad \vec{f}_t = \frac{(\Delta\vec{p}_a \cdot \hat{t})}{\Delta t} \hat{t} \quad (10)$$

where  $\Delta t$  is time step and  $\hat{t}$  is unit vector in the tangential direction of motion. We choose  $\hat{t}$  to be in the same direction as the relative motion in the tangential direction such that  $(\vec{v}_b - \vec{v}_a) \cdot \hat{t} > 0$  and identically such that  $\Delta\vec{p}_a \cdot \hat{t} > 0$  or that the angle between  $\hat{t}$  and  $\vec{v}_b - \vec{v}_a$  is between -90 and 90 degrees. From the known  $\hat{n}$ , the tangential vector is found from

$$\Delta\vec{p}_a - (\Delta\vec{p}_a \cdot \hat{n})\hat{n} = m_{red}(\Delta\vec{v} \cdot \hat{t})\hat{t} \quad (11)$$

The vector on the left can be calculated from prior results and is seen to be parallel to  $\hat{t}$ . There are two solutions for  $\hat{t}$  for the two directions along the parallel. We choose the solution such that  $\Delta\vec{v} \cdot \hat{t} > 0$  or equivalently that  $\Delta\vec{p}_a \cdot \hat{t} > 0$ .

The positive normal contact force is thus  $f_n = -(\Delta\vec{p}_a \cdot \hat{n})/\Delta t$  and the positive sliding force is  $f_t = (\Delta\vec{p}_a \cdot \hat{t})/\Delta t$ . If  $f_t < \mu f_n + s_a$ , the objects will stick using the above stick conditions (here  $s_a$  is an optional adhesion strength). Otherwise, the frictional force is changed to  $\mu f_n + s_a$ , but it still sticks in the normal direction. The complete frictional sliding algorithm in an optimized form is

1. Determine if contact is occurring. Those calculations will find  $\Delta\vec{p}_a$  and  $\hat{n}$  and will verify that  $d_n = \Delta\vec{p}_a \cdot \hat{n} < 0$ . Finding an accurate normal is crucial. The recommended method is to consider the gradient of the volume gradient for the contacting pair of materials (when three or more material are present, the algorithm looks at the volume gradient from material  $a$  and the other material with the most volume). The normal is taken as being in the direction of the volume gradient for the material that has the largest magnitude of its volume gradient.
2. Find an unnormalized tangent vector from  $\vec{t} = \Delta\vec{p}_a - d_n\hat{n}$ . If  $\vec{t} \cdot \vec{t} = 0$ , then there is no tangential motion and the momentum change  $\Delta\vec{p}_a$  can be used and the algorithm is done (*i.e.*, skip to step 5 with  $\Delta\vec{p}_a = \Delta\vec{p}_a$ ). If it is not zero, proceed to next step.

3. Normalize  $\hat{t}$  and find  $d_t = \Delta\vec{p}_a \cdot \hat{t}$ . Since  $\hat{t}$  is defined from the tangential component of  $\Delta\vec{p}_a$ ,  $d_t \geq 0$ . The zero option is handled in previous step, so now it must be non-zero and positive.
4. If  $f_t > \mu f_n + s_a$  (or  $d_t > -\mu d_n + s_a \Delta t$ ), both sides of which are always positive, then frictional sliding is occurring. The force to make material  $a$  “stick” in normal direction and feel “frictional” force in tangential direction is  $\vec{f}_a = (-f_n)\hat{n} + (\mu f_n + s_a)\hat{t}$ . Thus, the momentum change for material  $a$  becomes  $\Delta\vec{p}'_a = \vec{f}_a \Delta t = d_n(\hat{n} - \mu\hat{t}) + s_a\hat{t}\Delta t$ . But, if  $d_t < -\mu d_n + s_a \Delta t$ , use  $\Delta\vec{p}'_a = \Delta\vec{p}_a$ ; *i.e.*, use the stick-condition momentum change.
5. Change momenta of material  $a$  by  $\Delta\vec{p}'_a$
6. If the node has only two materials and the method to find normals is the same for both materials, then change momenta of material  $b$  by  $-\Delta\vec{p}'_a$  and this node is done. If three or more materials are present, or if normals may differ, proceed to next material at this node.

### 3.1 Low Mass Situations

The above algorithm never divides by a possibly low mass, but that does not necessarily mean it is not affected by one of the masses being very low. In GIMP calculations it will be common for a node one element away from the contact region to have nearly all mass in one material and a very low mass in the other material. When that occurs:

$$\text{if } m_a \ll m_b, \text{ then } \Delta\vec{p}_a \approx m_a \vec{v}_b - \vec{p}_a \quad \text{or} \quad \text{if } m_b \ll m_a, \text{ then } \Delta\vec{p}_a \approx \vec{p}_b - m_b \vec{v}_a \quad (12)$$

In other words,  $\Delta\vec{p}_a$  will be based on the unreliable nodal momentum (since it is likely to be from one particle on the edge of a material). Since  $\Delta\vec{p}_a$  is used to determine contact and to adjust the momenta, the entire algorithm will be unreliable. Now, since  $\Delta\vec{p}_a$  will also be very small, it might not make a difference, but in one calculation sliding down an incline plane, the calculation stopped when accepting all contact situations and was fixed by screening out those with  $m_a/(m_a + m_n)$  or  $m_b/(m_a + m_n)$  less than  $10^{-6}$ . Thus in step one above, if the mass ratio meets these criteria, ignore contact for that node.

### 3.2 Rigid Material Contact

If material  $b$  is rigid, contact can be handled by a special case of the two-material contact, A rigid material corresponds to  $m_b \rightarrow \infty$ . The basic contact equations change to

$$\vec{v}_c = \vec{v}_b \quad (13)$$

$$\Delta\vec{p}_a = -m_a(\vec{v}_a - \vec{v}_b) \quad (14)$$

$$\Delta\vec{v} = \vec{v}_b - \vec{v}_a = \frac{\Delta\vec{p}_a}{m_a} \quad (15)$$

$$m_{red} = m_a \quad (16)$$

After these changes, the contact methods above can be applied the same way except the momentum of the rigid particle is not changed. In `NairnMPM`, the density of rigid materials is set to 1000 g/cm<sup>3</sup> such that material point mass found in `NairnMPM::PreliminaryCalcs()` is equal to the volume for the rigid material point in mm<sup>3</sup>. Thus the mass gradient needed in the contact algorithm is equal to the volume gradient. The nodal mass, however, for material velocity field of rigid particles is set to zero as a way to determine a rigid particle velocity field. The nodal  $\vec{v}_k$  is always the correct velocity and is calculated when the first rigid particle is extrapolated to that node.