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

John Nairn

April 8, 2017

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. The crack velocity field number n is denoted by square brackets — $[n]$. 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 geting 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. These can be visualized by looking at Figure 1.

1.2 Material Velocity Fields

When running in multimaterial mode (which is needed to model material contact), each crack velocity field may have 1 to n material velocity fields where n is the number of materials active in the simulations. When not using multimaterial mode, n is set to 1 and all materials use the same velocity field in each crack velocity field. All this fields are on the node. In these notes, we denote a quantity, q, in a material velocity field (which is where momentum, forces, *etc.* are stored) on the grid as

$$q^{(k)}_{i,c:m} \tag{1}$$

where i is node number, c is crack velocity field, m is material velocity field, and k indicates time step number. When no cracks are present, c can be omitted; when in single material model, m can

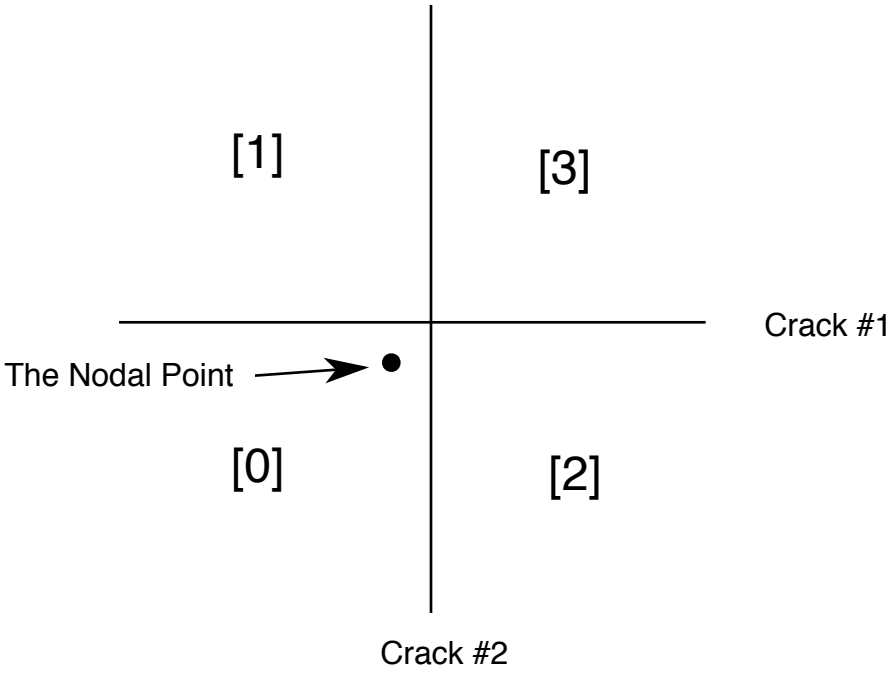

Figure 1: Schematic view of crack velocity fields around a single node. The possibility of two interacting cracks leads to at most four velocity fields.

be omitted, when no cracks and single material mode. both c and m can be ommitted (*i.e.*, each node has only on material velocity field). When analyzing all values within a single time step, (k) can be omitted for simplicity.

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 interfaces, to traction laws on cracks, and to nodal velocity boundary conditions.

2.1 Simulation Initialization

Each nodal point is permanently assigned crack field [0]. Other crack fields are dynamically allocated as needed. When in single material velocity field, field [0] is given a single material velocity field (field [0:0] for [crack field number:material field number]). When using multimaterial mode, crack velocity field [0] is assigned pointers to material velocity fields 0 to $m-1$ where m is the number of active materials in the simulations (each index maps to material number defined in the input file). The pointers are initialized to `NULL`. Any material velocity fields needed in a simulation are dynamically allocated as needed.

2.2 Time Step Initialization

All currently allocated crack and material velocity fields are zeroed. Then each material point is examined and compared to all nodes it sees with non-zero shape functions. When cracks a present,

a line is traced from the particle to the node and compared to crack surfaces. The end result is determination of the particles crack velocity field number `vfld` from [0] to [3]. If the node does not have that crack velocity field, it is allocated on that node. When no cracks are present `vfld` is set to 0. The crack field number, `vfld`, is stored on the particle. This crack field number determination is likely the most time-consuming part of explicit crack calculations.

A table look up converts material number (from input file) to index `matfld` (or field ID) into the `mvf[]` array of pointers on each crack velocity field. If that pointer `mvf[matfld]` is `NULL`, a new material velocity field is created.

2.3 Momentum and Mass Extrapolation to the Grid

For each material point, 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` (where `vfld` is stored on the particle). To support crack and material contact, each node needs to extrapolate particle particle position or displacement, particle size, and particle gradient (the last two are for normal detection and gradient only when in multimaterial mode). The relevant quantities for material velocity field j =`matfld` in crack velocity field c =`vfld` for particle p are:

$$m_{i,c:j} = \sum_{p\in j} S_{ip} m_p \tag{2}$$

$$\boldsymbol{p}_{i,c:j} = m_{i,c:j}\boldsymbol{v}_{i,c:j} = \sum_{p\in j} S_{ip}\boldsymbol{p}_p \tag{3}$$

$$m_{i,c:j}\boldsymbol{x}_{i,c:j} = \begin{cases} \sum_{p\in j} S_{ip} m_p \boldsymbol{x}_p & \text{position extrapolation} \\ \sum_{p\in j} S_{ip} m_p \Delta\boldsymbol{x}_p & \text{displacement extrapolation} \end{cases} \tag{4}$$

$$\Omega_{i,c:j} = \begin{cases} \sum_{p\in j} S_{ip} V_p & \text{3D} \\ \sum_{p\in j} S_{ip} A_p & \text{2D and axisymmetric} \end{cases} \tag{5}$$

$$\boldsymbol{g}_{i,c:j} = \begin{cases} \sum_{p\in j} \boldsymbol{G}_{ip} V_p & \text{3D} \\ \sum_{p\in j} \boldsymbol{G}_{ip} A_p & \text{2D and axisymmetric} \end{cases} \tag{6}$$

The time step (k) has been omitted.

After the extrapolations are done, each node is revisited for more calculations and when needed to impose multimaterial and crack contact and imperfect interfaces. The steps are

1. Copy the initially extrapolated momenta $\boldsymbol{p}_{i,c:j}$ for all crack and material velocity fields on all nodes.

2. When in multimaterial mode, check all nodes with more than one material. The material possibilities are:

   - If materials are in contact, calculate the change in momenta needed to reflect the contact physics (more details below) as $\Delta p_{i,c:a}$. Add this momentum to material a and opposite of this momentum to material b. Thus momentum is conserved. Some hacks are needed if the node has more than two materials.
   - If the two (or more) materials in contact are connected by an imperfect interface, the calculations are different. Here calculate the interface force $\boldsymbol{f}^{(INT)}_{i,c:a}$ on material a which will depend on calculated material separation. Add this force to nodal force and add total interfacial energy to global interface energy variable.

- Note that the different treatments for contact and interface were found needed. If imperfect interface were modeled as a momentum change at this stage, the results are wrong. Interfaces can be done my finding momentum change and force after the momentum update, but adding interface force here appears more stable.

3. When cracks are present, check all nodes the see more than one crack velocity field and undertake contact calculations for all crack contact situations that are present (*i.e.*, between [0]&[1] and [2]&[3] on the crack #1 surface and between [0]&[2] and [1]&[3] on the crack #2 surface).

   - For each crack pair, if faces are in contact, calculate the change in momenta needed to reflect the contact physics (more details below) as $\Delta p_{i,c}$. Add equal an opposite momenta changes to the crack velocity fields on opposites of the crack. The added momentum is spread out over all materials (weighted by their mass fraction) if the crack velocity field has more than one material.
   - If the crack surface are connected by an imperfect interface, find interfacial force, $\boldsymbol{f}_{i,c}^{(INT)}$, and add to each crack velocity field (spreading out of material by mass fraction if needed). Also add interfacial energy. Interfaces are handled different than contact as explained above.

4. For all nodes with grid velocity fields, change the nodal momenta to be:

$$\boldsymbol{p}'_{i,c:m} = m_{i,c:m} v_i^{(BC)} \hat{n} + \left(\boldsymbol{p}_{i,c:m} \cdot \hat{t}\right) \hat{t} \tag{7}$$

$$= \boldsymbol{p}_{i,c:m} + \left(m_{i,c:m} v_i^{(BC)} - \boldsymbol{p}_{i,c:m} \cdot \hat{n}\right) \hat{n} \tag{8}$$

   where $\hat{t}$ is unit vector in tangential direction (but it is not needed in the second form). where $v_i^{(BC)}$ is impose grid velocity applied in the $\hat{n}$ direction. Note that crack and material velocity field indices have been omitted on mass and momentum, but this change is done for all crack and material velocity fields on the node.

2.4 Update Stresses and Strains on Particle

When updating stress and strains before finding forces and updating momenta (needed for `USAVG`$\pm$ and for `USF` update methods), extrapolate velocity gradient ($\nabla \boldsymbol{v}$) to the particles and implement the chosen constitutive laws.

2.5 Project Forces to the Grid

For each material point, extrapolate $f_{i,c:m}$ due to particle stresses and $f_{i,T,c:m}$ due to particle traction loads. These are added to interface forces (if any are present). The values from particle p are added to material velocity field `matfld` of crack velocity field `vfld`, where `vfld` was found during time step initialization and `matfld` is known by the material point's material type. The other force calculations done are:

1. Simple particle force boundary conditions are added for those particles (although these force are actually set in the initialization task).

2. If any cracks have traction laws, forces current stored on crack particles are interpolated to nodes. Note that this extrapolation uses ordinaary grid shape functions because crack particle have no volume and GIMP shape function imply a particle domain.

3. Add any body forces (such as gravity).

4. Restore nodal momenta to the initially extrapolated values that were copied above. This task removes and momentum changes that were done prior to updating stresses and strains. The changes were caused by contact (not interfaces) and by grid velocity conditions.

Once all above force steps are done, the force on nodes with grid velocity boundary conditions are changed to be consistent with that condition. The updated momentum on a node with prescribed velocity in the $\hat{n}$ direction needs to be:

$$\begin{aligned} \boldsymbol{p}_{i,c:m}^{(k+1)} &= m_{i,c:m} v_i^{(BC)} \hat{n} + \left(\left(\boldsymbol{p}_{i,c:m}^{(k)} + \boldsymbol{f}_{i,c:m}^{(tot)} \Delta t \right) \cdot \hat{t} \right) \hat{t} \qquad (9) \\ &= \boldsymbol{p}_{i,c:m}^{(k)} + \left(\boldsymbol{f}_{i,c:m}^{(tot)} + f_{i,c:m}^{(BC)} \hat{n} \right) \Delta t \qquad (10) \end{aligned}$$

where $f_{i,c:m}^{(BC)}$ is magnitude of reaction force applied in the $\hat{n}$ direction to satisfy the boundary condition on that node. The normal component of the updated momentum must give the boundary condition or:

$$\boldsymbol{p}_{i,c:m}^{(k+1)} \cdot \hat{n} = m_{i,c:m} v_i^{(BC)} = \boldsymbol{p}_{i,c:m}^{(k)} \cdot \hat{n} + (\boldsymbol{f}_{i,c:m}^{(tot)} \cdot \hat{n}) \Delta t + f_{i,c:m}^{(BC)} \Delta t \qquad (11)$$

This equation can be solved to find magnitude of the reaction force:

$$f_{i,c:m}^{(BC)} = \frac{m_{i,c:m} v_i^{(BC)} - \boldsymbol{p}_{i,c:m}^{(k)} \cdot \hat{n}}{\Delta t} - \boldsymbol{f}_{i,c:m}^{(tot)} \cdot \hat{n} \qquad (12)$$

To impose this condition after all other forces are found, revisit all nodes with boundary conditions and add $f_i^{(BC)} \hat{n}$ to nodal force, which results in

$$\boldsymbol{f}_{i,c:m}^{(tot)'} = \boldsymbol{f}_{i,c:m}^{(tot)} - (\boldsymbol{f}_{i,c:m}^{(tot)} \cdot \hat{n}) \hat{n} + \frac{m_{i,c:m} v_i^{(BC)} - \boldsymbol{p}_{i,c:m}^{(k)} \cdot \hat{n}}{\Delta t} \hat{n} \qquad (13)$$

To support superposition of velocity conditions on the same node (but in the same direction $\hat{n}$), the boundary condition value is replaced by

$$m_i^{(n)} v_i^{(BC)} = \sum_k m_i^{(n)} v_i^{(BC,k)} \qquad (14)$$

where $v_i^{(BC,k)}$ is velocity for boundary condition k. A single node can superpose boundary conditions in different directions only if the directions are orthogonal. Note that this approach to velocity boundary conditions (rather then just imposing them in the subsequent momentum update) is required to have the accelerations determined from nodal forces be the correct accelerations needed to move the particle from its initial position in the time step to the proper position the reflects the boundary conditions.

2.6 Update Nodal Momenta

This task updates the nodal momenta using

$$\boldsymbol{p}_{i,c:m}^{(k+1)} = \boldsymbol{p}_{i,c:m}^{(k+1)} + f_{i,c:m} \Delta t \qquad (15)$$

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 task, any time the momentum is changed by $\Delta \boldsymbol{p}_{i,c:m}$, the corresponding force must be changed by $\Delta \boldsymbol{p}_{i,c:m}/\Delta t$. Second, the momentum component for any node that fixes that component with a velocity boundary condition must not be changed.

2.7 Update the Particles

Update particle position and velocity using nodal velocity (from $\boldsymbol{v}_{i,c:m} = \boldsymbol{p}_{i,c:m}^{(k+1)}/m_{i,c:m}$) and acceleration (from $\boldsymbol{a}_{i,c:m} = \boldsymbol{f}_{i,c:m}/m_{i,c:m}$) found in material velocity field `matfld` (for type of particle) of crack velocity field `vfld` (determined on current time step for the particle.

2.8 Optional Update Strain Last

Optionally the calculations can update stain after the momentum update. This calculation is done for methods `USAVG`$\pm$ and for `USL`$\pm$. The - (minus) version updates using the update velocity and their extrapolated velocity gradient (exactly like the first stress and strain update). The + (plus) method does new extrapolation of momentum to the grid. 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. The momenta change are only for contact and not for interfaces with imperfect interfaces or for traction loaded cracks.

2.9 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). In addition, the new crack opensing displacements are used to calculation traction loads to be input to the next time step during its force calculations.

3 Finding Normal Vector for Contact and Interfaces

Get the normal correct is vital to accurate contact in MPM. Most problems involving contact can be traced to inaccurate. Although dynamic contact detection is advantage of MPM, finding the normal is cost of using that advantage. Consider material a and b in one crack velocity field on node i. All except material ID in property subscripts will by omitted for clarity. The goal is to find normal $\hat{n}$ directed from a to b. Ignoring b, a unit normal can be estimated from volume gradient:

$$\hat{n} = \frac{\boldsymbol{g}_a}{||\boldsymbol{g}_a||} \tag{16}$$

Note that code should use volume gradient instead of mass gradient to handle situations involving contact between materials with different densities. This normal may or may not be accurate and may or may not be same as corresponding normal found from material b. Some MPM codes found normal for each material and looped over materials. But when normals are not equal and opposite, the momentum would not be conserved. A better approach is to use a single normal for the material pair. No on method works best. The options in `OSParticulas` are:

- `AVGGRAD`: Find normal from volume average volume gradients of the two materials:

$$\hat{n}||\hat{n}|| = \Omega_a \boldsymbol{g}_a + \Omega_b \boldsymbol{g}_b \tag{17}$$

  The justification is that averaging is often best and volume weighting counts the one with more information (*i.e.* closest to the contact point) the most.

- `MAXGRAD`: Find normal from materials whose volume gradient has the largest magnitude:

$$\hat{n}||\hat{n}|| = \begin{cases} \frac{\boldsymbol{g}_a}{||\boldsymbol{g}_a||} & \text{if } ||\boldsymbol{g}_\text{a}|| > ||\boldsymbol{g}_\text{b}|| \\ -\frac{\boldsymbol{g}_b}{||\boldsymbol{g}_b||} & \text{otherwise} \end{cases} \tag{18}$$

  The justification is that one with more information, represented by higher absolute magnitude, might have more accurate information.

- `MAXVOL`: Find normal from materials whose volume has the largest magnitude:

$$\hat{n}||\hat{n}|| = \begin{cases} \frac{\boldsymbol{g}_a}{||\boldsymbol{g}_a||} & \text{if } \Omega_\text{a} > \Omega_\text{b} \\ -\frac{\boldsymbol{g}_b}{||\boldsymbol{g}_b||} & \text{otherwise} \end{cases} \tag{19}$$

  The justification is that one with more information, represented by more volume, might have more accurate information.

- `OWN`: Find normal from each material and consider contact effects separately for all materials:

$$\hat{n} = \frac{\boldsymbol{g}_a}{||\boldsymbol{g}_a||} \tag{20}$$

  This method is never recommended and only included in `OSParticulas` to allow comparison to old MPM methods for handling contact.

- `SPECIFY`: Instead of calculating the normals, prescribe a normal that applies to all contact situations. The normal is directed from the lower number material (in the list of defined materials) to the higher one. This approach is only viable when all contact surfaces have the same normal and the normal does not change during the simulation. This situation is uncommon in real problems. Note that if the contact is with a rigid material (see details below), a specified normal mans from non-rigid into the rigid material, regardless of order of the material numbering.

4 Detecting Contact

The next task is to decide if materials are in contact or if they are separated. When in contact, the materials are often left to move in the separate velocity fields. When they are in contact, however, the momenta have to change to prevent interpenetration and to implement contact mechanics. If the contact law includes adhesion, it may be necessary to examine separated material as well to decided if they have overcome some input adhesion value. For imperfect interface, force generally need to be added for both contacted and separated interfaces. All cases need to know which has occurred. Below are the following steps used in `OSParticulas`. The steps can be used together (*i.e.*, must pass all test to be in contact) or some can be skipped (with one exception). Two prescreening test are run:

- **Volume Prescreening**: Two checks are allowed to skip contact and interface calculations on a node. The first is a volume screen where total volume on a node (sum of material volumes) is compared to total volume associated with that node (*i.e.*, cell volume — for grid with unequal element sizes, it is net volume of cells around node i). Let V_r be ratio of material volumes to node i cell volume. This prescreen will ignore contact on this node if $V_r < V_{min}$ where V_{min} is simulations parameter. The logic is not that contact ad interface are not present, but rahter that this node is likely on the edge of a material and the calculations will be inaccurate. It might better to skip them to to attempt them. Normally $V_{min} = 0$, which skips this prescreen. If problems are noted near an edge, it can be increased (such as to $V_{min} = 0.5$) to skip such nodes. Attempts to use V_{mmin} around 1 (on idea that contact nodes on in compression with increased material volume) as primary contact detection have not worked well.

- **Mass Prescreening**: The final change in momentum is related to reduced mass on the node, which means the change will be neglible if the mass ratio is higher or small. This prescreen skips contact calculations if:

$$\frac{m_a}{m_{tot}} < \epsilon \qquad \text{or} \qquad \frac{m_a}{m_{tot}} > 1 - \epsilon \tag{21}$$

  where ϵ is a small number. The current version of `OSParticulas` uses $\epsilon = 1.e^{-6}$.

The above screening options can be done without calculations of the normal. There are thus done before finding the normal. If the calculations are nor screened out, the normal is calculated (as explained above) and then contact is determined by one or both of the following checks. The first is mandatory, but the second is options (although always recommended). They are described in following two sections

4.1 Approach Velocity

For two materials with velocities $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$, the center of mass velocity is

$$\boldsymbol{v}_c = \frac{m_a \boldsymbol{v}_a + m_b \boldsymbol{v}_b}{m_a + m_b} \tag{22}$$

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

$$\boldsymbol{v}_a + \frac{\Delta \boldsymbol{p}_a}{m_a} = \boldsymbol{v}_c \quad \text{which means} \quad \Delta \boldsymbol{p}_a = -m_a(\boldsymbol{v}_a - \boldsymbol{v}_c) \tag{23}$$

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

$$\Delta \boldsymbol{p}_a = -m_a(\boldsymbol{v}_a - \boldsymbol{v}_c) = m_b(\boldsymbol{v}_b - \boldsymbol{v}_c) = \frac{m_a \boldsymbol{p}_b - m_b \boldsymbol{p}_a}{m_a + m_b} \tag{24}$$

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

$$\Delta \boldsymbol{v} = \boldsymbol{v}_b - \boldsymbol{v}_a = \frac{\Delta \boldsymbol{p}_a}{m_b} + \frac{\Delta \boldsymbol{p}_a}{m_a} = \frac{\Delta \boldsymbol{p}_a}{m_{red}} \tag{25}$$

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, $\Delta \boldsymbol{p}_a$. For detecting contact based

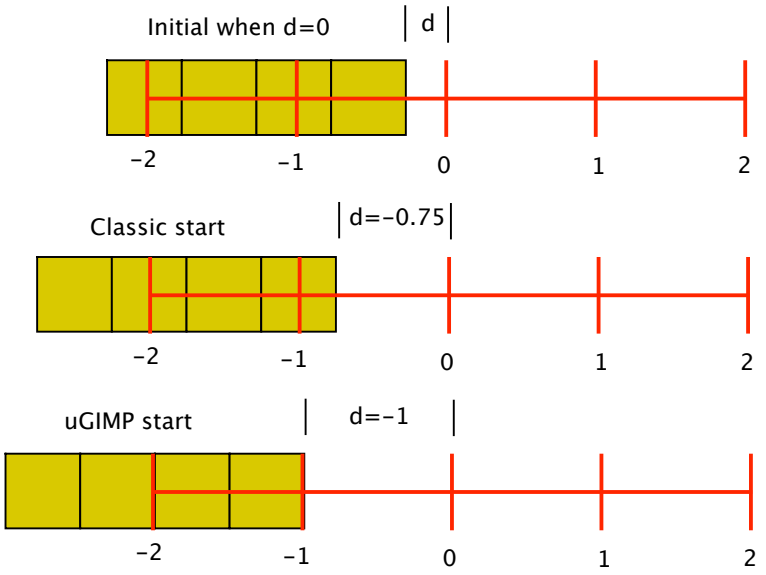

Figure 2: One dimensional extrapolation of particle position to grid node. Node i is at $x = 0$ (x in units of cells size). d is distance from edge of material points to node i. The "Classic" and "uGIMP" start are the position the extrapolation first becomes nonzero.

on approach velocity, the two surfaces must be approaching each other. With a normal directed from a to b, this criteria implies $\Delta \boldsymbol{v} \cdot \hat{n} < 0$ or identically that $\Delta \boldsymbol{p}_a \cdot \hat{n} < 0$.

We thus assume that under all contact situations that $\Delta \boldsymbol{p}_a \cdot \hat{n} < 0$. In other words, this criterion must always be included. As shown below, this criterion is equivalent to a finding the contact surface have negative traction (*i.e.*, compression stress) in the normal direction. Many contact laws (such as friction) are based on compression state. With this check, those contact laws can assume that $\Delta \boldsymbol{p}_a \cdot \hat{n} < 0$ in their calculations.

4.2 Determination of Material Separation

The use of approach velocity alone is not enough. It often implies contact even when edges are clearly separated. The detection of contact is improved by looking a separation between the materials in the normal direction. Two approaches are possible. One is to extrapolate displacement to the grid. While this works, it requires that the two materials start out in contact such that zero displacement is equal to contact. When the materials come into contact only after motion (such as closing a gap), displacements are not enough. An alternative approach is to extrapolate particle position. The two possible extrapolations are in Eq. (4).

By either method, the interfacial displacement discontinuity on node i (subscript i omitted for clarity) is:

$$\boldsymbol{\delta} = \boldsymbol{x}_b - \boldsymbol{x}_a \tag{26}$$

where a and b are two materials (for material contact) or above and below a crack. The separation is then given by

$$\delta_n = \delta \cdot \hat{n} \tag{27}$$

When extrapolating displacements, $\delta_n \leq 0$ can be used to detect contact, but when extrapolating positions, more calculations are needed.

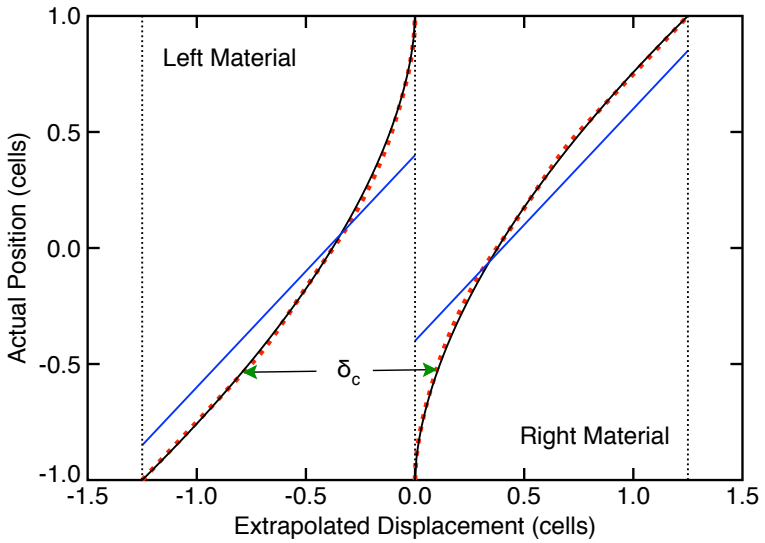

Figure 3: Plot of actual distance from edge to node i at $x = 0$ (d) as a function of extrapolated position (d_{ext} on node i. The "Left Material" if for edge approaching from the left. The "Right Material" is for an edge departing to the right.

To correct extrapolated positions for material separation calculations we consider 1D extrapolation near the edge of an object. Figure 2 shows group of undeformed particles approaching node i at $x = 0$ with d as the actual distance from the edge of the material points to that node. The "Classic" start shows that position on node i will be zero until $d = -0.75$ or until the center of the first particle enters the element to the left of node i. Both `GIMP` and `CPDI` will start extrapolating to node i the instant the edge enters that element.

For each d, or actual distance from edge of the material points to node i, we can calculate d_{ext}, or the position extrapolated to node i by given shape function method. Once these results are found, we invert and the results as shown in Fig. 3. The "Left Material" is for edge approaching from the left. When the particle edge first enters the element to left of node i (assuming `GIMP` or `CPDI`), the extrapolated position (from particle center) will be -1.25 from the node. As the edge approaches, the extrapolation will decrease, but never becomes negative. When the particles completely fill the two elements on opposite sides of the node, the extrapolated position will be zero. Thus extrapolation values from -1.25 to 0 are mapping to actual edge distances from $d = -1$ to 1. The dotted red lines shows this mapping for `GIMP` (Classic would be piecewise linear appromation starting at (-1,-0.75) and `CPDI` should be similar to `CPDI`, but neither are shown). The dotted red curve for "Right Material" is the same calculation for edge of particles departing to the right.

Now imagine two materials (or crack surfaces) near node i. The the normal from material a to b is taken as positive direction, then material a and the "left" material and b is the "Right" material. The separation is written as:

$$\delta_n = d(d_{b,ext}) - d(d_{a,ext}) \quad \text{where} \quad d(d_{m,ext}) = (\boldsymbol{x}_m - \boldsymbol{x}_i) \cdot \hat{n} \tag{28}$$

In general, we expect $d_{b,ext} > 0$ and $d_{a,ext} < 0$ and δ_n is actual separation or $\delta_n \leq 0$ is contact.

`OSParticulas` has two methods to find δ_c. The first method (as published in my papers) is to

assume the $d()$ functions are linear and given by:

$$d(d_{ext}) = \begin{cases} d_{ext} + 0.4\Delta x & \text{left} \\ d_{ext} - 0.4\Delta x & \text{right} \end{cases} \tag{29}$$

where Δx is an effective cell size (and equal to cell size with square (2D) or cubic (3D) cells, but altered for rectangular cells). With this equation, the material separation becomes

$$\delta_n = (\boldsymbol{x}_b - \boldsymbol{x}_i) \cdot \hat{n} - 0.4\Delta x - \big((\boldsymbol{x}_a - \boldsymbol{x}_i) \cdot \hat{n} - 0.4\Delta x\big) = \boldsymbol{\delta} x - 0.8\Delta x \tag{30}$$

My paper descrived as subtracting a contact, but the net effect if assuming the above linear conversion form d_{ext} to d. This net subtraction of 0.8 was found by a weighted average of $d(d_{b,ext}) - d(d_{a,ext})$ where the weighting counted the result near $d = 0$ more. The weighting logic was that momentum change is greatest when d is close to zero or contact is near a node. Although the subtraction value can be varied to any value, it appears results far from 0.8 cells give poor results or that 0.8 cells appears to be universal value when using this linear method.

An alternate approach is to fit the inverted curves derived from shape function calculations. The solid black lines show the fits:

$$\frac{d(d_{ext})}{\Delta x} = \begin{cases} 1 - 2\left(\frac{-d_{ext}}{1.25}\right)^{0.58} & \text{left} \\ 2\left(\frac{d_{ext}}{1.25}\right)^{0.58} - 1 & \text{right} \end{cases} \tag{31}$$

Thus in contact calculations, one finds $d(d_{ext})$ for left (a) and right (b) materials using this equation, and then subtracts for the actual δ_n or surface separation. Contact occurs when $\delta_n \leq 0$. This option is selected in `OSParticulas` by setting the `ContactPosition` to a negative number and the absolute value of that number becomes the exponent in the above fits. Like 0.8 method, the exponent can be varied, but it appears that -0.58 (in the command or use exponent of 0.58) is universal value that should not be adjusted.

In summary, the three options in `OSParticulas` are:

1. No `ContactPosition` command — if contact position is never set, contact is determined by extrapolated displacements and no correction is applied to the extrapolated values. (Note: this is **not** the same as setting contact position to 0, which is to extrapolate position instead of displacement but not correct the values; these results would be very bad).

2. Positive `ContactPosition` — contact is determined by extrapolated positions which are adjusted by constant value provided (should probably always be 0.8).

3. Negative `ContactPosition` — contact is determined by extrapolated positions which are adjusted Eq. (31) where exponent is absolute value of value provided (should probably always be -0.58).

5 Changes in Momentum for Contact

This section considers changes in momenta (or applied forces) needed to enforce a contact law. The basic analysis if for material a and contact with material b, but to handle general problems where a node may see more that two materials (although it may not be handled rigorously), the following algorithm is uses. At a node with any number of materials, the center of mass results are:

$$\boldsymbol{p}_c = \sum_i \boldsymbol{p}_i \qquad m_c = \sum_i m_i \qquad \boldsymbol{v}_c = \frac{\boldsymbol{p}_c}{m_c} \tag{32}$$

We define a *virtual* material b derived from all mass and momenta not associated with material a:

$$\boldsymbol{p}_b = \boldsymbol{p}_c - \boldsymbol{p}_a \qquad m_b = m_c - m_a \qquad \boldsymbol{v}_b = \frac{\boldsymbol{p}_b}{m_b} \tag{33}$$

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

$$\Delta\boldsymbol{p}_a = \frac{m_a}{m_c}\boldsymbol{p}_c - \boldsymbol{p}_a, \qquad \Delta\boldsymbol{v} = \frac{\Delta\boldsymbol{p}_a}{m_{red}}, \qquad \text{and} \qquad m_{red} = \frac{m_a(m_c - m_a)}{m_c} \tag{34}$$

All these results reduces exactly to two-material case for nodes with two materials.

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 $\Delta\boldsymbol{p}_a$ and $\Delta\boldsymbol{v}$ are found. Stick conditions are trival. The momentum change of $\Delta\boldsymbol{p}_a$ is applied to material a. These momenta changes correspond to normal and tangential forces of

$$\boldsymbol{f}_n = \frac{(\Delta\boldsymbol{p}_a \cdot \hat{n})}{\Delta t}\hat{n} \qquad \text{and} \qquad \boldsymbol{f}_t = \frac{(\Delta\boldsymbol{p}_a \cdot \hat{t})}{\Delta t}\hat{t} \tag{35}$$

where Δt 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 $(\boldsymbol{v}_b - \boldsymbol{v}_a) \cdot \hat{t} > 0$ and identically such that $\Delta\boldsymbol{p}_a \cdot \hat{t} > 0$ or that the angle between $\hat{t}$ and $\boldsymbol{v}_b - \boldsymbol{v}_a$ is between -90 and 90 degrees. Give $\hat{n}$, $\hat{t}$ can be found from

$$\hat{t} = \frac{\Delta\boldsymbol{p}_a - (\Delta\boldsymbol{p}_a \cdot \hat{n})\hat{n}}{||\Delta\boldsymbol{p}_a - (\Delta\boldsymbol{p}_a \cdot \hat{n})\hat{n}||} \tag{36}$$

The positive normal contact force is thus $f_n = -(\Delta\boldsymbol{p}_a \cdot \hat{n})/\Delta t$ and the positive sliding force is $f_t = (\Delta\boldsymbol{p}_a \cdot \hat{t})/\Delta t$. 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 Δp_a and $\hat{n}$ and will verify that $d_n = \Delta\boldsymbol{p}_a \cdot \hat{n} < 0$. 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 $\boldsymbol{t} = \Delta p_a - d_n\hat{n}$. If $\boldsymbol{t} \cdot \boldsymbol{t} = 0$, then there is no tangential motion and the momentum change Δp_a can be used and the algorithm is done (*i.e.*, skip to step 5 with $\Delta\boldsymbol{p'}_a = \Delta\boldsymbol{p}_a$). If it is not zero, proceed to next step.

3. Normalize $\hat{t}$ and find $d_t = \Delta\boldsymbol{p}_a \cdot \hat{t}$. Since $\hat{t}$ is defined from the tangential component of $\Delta\boldsymbol{p}_a$, $d_t \geq 0$. 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 $\boldsymbol{f}_a = (-f_n)\hat{n} + (\mu f_n + s_a)\hat{t}$. Thus, the momentum change for material a becomes $\Delta\boldsymbol{p'}_a = \boldsymbol{f}_a\Delta t = d_n(\hat{n} - \mu\hat{t}) + s_a\hat{t}\Delta t$. But, if $d_t < -\mu d_n + s_a\Delta t$, use $\Delta\boldsymbol{p'}_a = \Delta\boldsymbol{p}_a$; *i.e.*, use the stick-condition momentum change.

5. Change momenta of material a by $\Delta\boldsymbol{p'}_a$

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 $-\Delta\boldsymbol{p'}_a$ and this node is done. If three or more materials are present, or if normals may differ, proceed to next material at this node.

5.1 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:

$$\text{if } m_a << m_b, \text{ then } \Delta\boldsymbol{p}_a \approx m_a\boldsymbol{v}_b - \boldsymbol{p}_a \quad \text{or} \quad \text{if } m_b << m_a, \text{ then } \Delta\boldsymbol{p}_a \approx \boldsymbol{p}_b - m_b\boldsymbol{v}_a \tag{37}$$

In other words, $\Delta\boldsymbol{p}_a$ will be based on the unreliable nodal momentum (since it is likely to be from one particle on the edge of a material). Since $\Delta\boldsymbol{p}_a$ is used to determine contact and to adjust the momenta, the entire algorithm will be unreliable. Now, since $\Delta\boldsymbol{p}_a$ 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.

5.2 Rigid Material Contact

Specify normals changes and add rigid bias option.

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

$$\boldsymbol{v}_c = \boldsymbol{v}_b \tag{38}$$

$$\Delta\boldsymbol{p}_a = -m_a(\boldsymbol{v}_a - \boldsymbol{v}_b) \tag{39}$$

$$\Delta\boldsymbol{v} = \boldsymbol{v}_b - \boldsymbol{v}_a = \frac{\Delta\boldsymbol{p}_a}{m_a} \tag{40}$$

$$m_{red} = m_a \tag{41}$$

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^3 such that material point mass found in `NairnMPM::PreliminaryCalcs()` is equal to the volume for the rigid material point in mm^3. 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 `vk` is always the correct velocity and is calculated when the first rigid particle is extrapolated to that node.

6 Imperfect Interface Calculations

The displacement at node i for materials a and b in crack field c after current time step can be written as:

$$\boldsymbol{u}_a(\Delta t) = \boldsymbol{u}_a(0) + \frac{\boldsymbol{p}_a\Delta t}{m_a} + \frac{(\boldsymbol{f}_a + \boldsymbol{f}_a^{(INT)})(\Delta t)^2}{2m_a} \tag{42}$$

$$\boldsymbol{u}_b(\Delta t) = \boldsymbol{u}_b(0) + \frac{\boldsymbol{p}_b\Delta t}{m_b} + \frac{(\boldsymbol{f}_b - \boldsymbol{f}_a^{(INT)})(\Delta t)^2}{2m_a} \tag{43}$$

where node i and crack field c have been omitted (for clarity) and $\boldsymbol{f}_a^{(INT)}$ is force that must be added to a and substracted from b to impose interface conditions. Here $\boldsymbol{f}_a$ is force found in the

current time step. When interface calculations are done after the momentum update, the initial momentum needed in above expression is found from updated momentum ($\boldsymbol{p}'_a$) using:

$$\boldsymbol{p}_a = \boldsymbol{p}'_a - \boldsymbol{f}_a \Delta t \tag{44}$$

The same holds for material b.

For interface modeling, we need to follow the displacement discontinuity or $\boldsymbol{\delta}(\Delta t) = \boldsymbol{u}_b(\Delta t) - \boldsymbol{u}_a(\Delta t)$:

$$\boldsymbol{\delta}(\Delta t) = \boldsymbol{\delta}(0) + \frac{\Delta \boldsymbol{p}_a \Delta t}{m_{red}} + \frac{\Delta \boldsymbol{f}_a (\Delta t)^2}{2m_{red}} - \frac{\boldsymbol{f}_a^{(INT)} (\Delta t)^2}{2m_{red}} \tag{45}$$

where

$$\Delta \boldsymbol{p}_a = \frac{m_a \boldsymbol{p}_b - m_b \boldsymbol{p}_a}{m_a + m_b}, \qquad \Delta \boldsymbol{f}_a = \frac{m_a \boldsymbol{f}_b - m_b \boldsymbol{f}_a}{m_a + m_b}, \quad \text{and} \quad m_{red} = \frac{m_a m_b}{m_a + m_b} \tag{46}$$

But the displacement discontinuity condition must also follow an equation of motion determined by interfacial traction law that superposes the forces found in MPM with interfacial forces, or:

$$\boldsymbol{\delta}(t) = \boldsymbol{\delta}(0) + \frac{\Delta \boldsymbol{p}_a t}{m_{red}} + \frac{\Delta \boldsymbol{f}_a t^2}{2m_{red}} - \frac{1}{m_{red}} \int_0^t dt_2 \int_0^{t_2} dt_1 \, \boldsymbol{F}_{int}(\boldsymbol{\delta}(t_1)) \tag{47}$$

where the interfacial force is

$$\boldsymbol{F}_{int}(\boldsymbol{\delta}(t)) = F_n(\delta_n(t))\hat{n} + F_t(\delta_t(t_1))\hat{t} \tag{48}$$

Here $\delta_n(t) = \boldsymbol{\delta}(t) \cdot \hat{n}$ and $\delta_t(t) = \boldsymbol{\delta}(t) \cdot \hat{t}$ are the normal and tangential displacement discontinuities and $F_n(\delta_n(t))$ and $F_t(\delta_t(t))$ and normal and tangential imperfect interface laws. Equating the two interfacial discontinuity equations at $t = \Delta t$, the interfacial force we must add (and subtract) becomes

$$\boldsymbol{f}_a^{(INT)} = \frac{2}{(\Delta t)^2} \int_0^{\Delta t} dt_2 \int_0^{t_2} dt_1 \, \boldsymbol{F}_{int}(\boldsymbol{\delta}(t_1)) \tag{49}$$

The integral term is found by solving the interfacial equations of motion. The equations for normal and tangential motion are:

$$\delta_n(t) = \delta_n(0) + \frac{d_n t}{m_{red}} + \frac{f_n t^2}{2m_{red}} - \frac{1}{m_{red}} \int_0^t dt_2 \int_0^{t_2} dt_1 \, F_n(\delta_n(t_1)) \tag{50}$$

$$\delta_t(t) = \delta_t(0) + \frac{d_t t}{m_{red}} + \frac{f_t t^2}{2m_{red}} - \frac{1}{m_{red}} \int_0^t dt_2 \int_0^{t_2} dt_1 \, F_t(\delta_t(t_1)) \tag{51}$$

where $d_n = \Delta \boldsymbol{p}_a \cdot \hat{n}$, $d_t = \Delta \boldsymbol{p}_a \cdot \hat{t}$, $f_n = \Delta \boldsymbol{f}_a \cdot \hat{n}$, and $f_t = \Delta \boldsymbol{f}_a \cdot \hat{t}$. These equations provide integral equations to be solved (for given interface law) to find $\delta_n(t)$ and $\delta_t(t)$. The analysis can be converted to differential equations:

$$\frac{d^2 \delta_n(t)}{dt^2} + \frac{F_n(\delta_n(t))}{m_{red}} = \frac{f_n}{m_{red}} \qquad \text{and} \qquad \frac{d^2 \delta_t(t)}{dt^2} + \frac{F_t(\delta_t(t))}{m_{red}} = \frac{f_t}{m_{red}} \tag{52}$$

Once these are solved, the solutions are used to find an *effective* interface force that must be added such that:

$$\delta_n(\Delta t) = \delta_n(0) + \frac{d_n \Delta t}{m_{red}} + \frac{f_n (\Delta t)^2}{2m_{red}} - \frac{f_{n,a}^{(INT)} (\Delta t)^2}{2m_{red}} \tag{53}$$

where the effective force is found from

$$f_{n,a}^{(INT)} = f_n + \frac{2}{\Delta t}\left(\frac{m_{red}\big(\delta_n(0) - \delta_n(\Delta t)\big)}{\Delta t} + d_n\right) \tag{54}$$

A similar equation is used for tangential force as well. The total force added to material a is then

$$\boldsymbol{f}_{i,c:a}^{(INT)} = f_{n,a}^{(INT)}\hat{n} + f_{t,a}^{(INT)}\hat{t} \tag{55}$$

And equal an opposite force is applied to material b.

6.1 Interfacial Energy

For calculations of composite properties, it is useful to track interfacial energy. Two approaches are to calculate total interfacial energy on each time step. This approach only works for elastic interfaces (need not be linear). The second approach is to find incremental interfacial energy.

The total interfacial energy is:

$$E_{int} = \int_0^{\delta_n} F_n(\delta)\, d\delta + \int_0^{\delta_t} F_t(\delta)\, d\delta \tag{56}$$

The incremental interfacial energy for one component

$$\Delta E_{int} = \int_0^{\Delta t} F_n(\delta_n(t))\frac{d\delta_n(t)}{dt}\, dt + \int_0^{\Delta t} F_t(\delta_t(t))\frac{d\delta_t(t)}{dt}\, dt \tag{57}$$

The derivative is found after solving for $\delta_n(t)$

6.2 Linear Imperfect Interface

For a linear imperfect interface, such as tangential component of linear interface material is `OSParticulas`, the differential equations can be solved in closed form. The tangential interface law is $F_t(\delta_t(t)) = D_t A \delta_t(t)$, where D_t is imperfect interface parameter and A is contact area. The differential equation and its solution are:

$$\frac{d^2\delta_t(t)}{dt^2} + \frac{D_t A \delta_t(t)}{m_{red}} = \frac{f_t}{m_{red}} \tag{58}$$

$$\delta_t(t) = \frac{f_t}{D_t A} + C_1 \sin\kappa_t t + C_2 \cos\kappa_t t \tag{59}$$

$$\text{where} \qquad \kappa_t = \sqrt{\frac{D_t A}{m_{red}}} \tag{60}$$

is frequency of the resulting harmonic oscillator. The initial conditions are

$$\delta_t(0) = C_2 + \frac{f_t}{D_t A} \qquad \text{and} \qquad \frac{\delta_t(0)}{dt} = C_1\kappa_t = \frac{d_t}{m_{red}} \tag{61}$$

The final solution becomes

$$\delta_t(t) = \frac{d_t}{m_{red}\kappa_t}\sin\kappa_t t + \left(\delta_t(0) - \frac{f_t}{D_t A}\right)\cos\kappa_t t + \frac{f_t}{D_t A} \tag{62}$$

$$= \frac{d_t}{m_{red}\kappa_t}\sin\kappa_t t + \left(\delta_t(0) - \frac{f_t}{m_{red}\kappa_t^2}\right)\cos\kappa_t t + \frac{f_t}{m_{red}\kappa_t^2} \tag{63}$$

The effective force is found using:

$$\delta_t(\Delta t) = \frac{d_t}{m_{red}\kappa_t}\sin\kappa_t\Delta t + \delta_t(0)\cos\kappa_t\Delta t + \frac{f_t}{m_{red}\kappa_t^2}(1-\cos\kappa_t\Delta t) \tag{64}$$

which leads to

$$\begin{aligned} f_{t,a}^{(INT)} &= f_t + \frac{2}{\Delta t}\left(\frac{m_{red}}{\Delta t}\left(\delta_t(0)-\delta_t(\Delta t)\right)+d_t\right) &(65)\\ &= f_t + \frac{2}{\Delta t}\left(d_t\left(1-\frac{\sin\kappa_t\Delta t}{\kappa_t\Delta t}\right)+\left(\frac{m_{red}\delta_t(0)}{\Delta t}-\frac{f_t}{\kappa_t^2\Delta t}\right)(1-\cos\kappa_t\Delta t)\right) &(66)\\ &= \frac{2d_t}{\Delta t}\left(1-\frac{\sin\kappa_t\Delta t}{\kappa_t\Delta t}\right)+\frac{2m_{red}\delta_t(0)\,(1-\cos\kappa_t\Delta t)}{(\Delta t)^2}+f_t\left(1-\frac{2\,(1-\cos\kappa_t\Delta t)}{\kappa_t^2(\Delta t)^2}\right) &(67)\end{aligned}$$

As $D_t \to 0$ or $\kappa_t \to 0$ and using L'Hopital's rule:

$$\begin{aligned} \lim_{\kappa_t\to 0}\frac{\sin\kappa_t\Delta t}{\kappa_t\Delta t} &= \lim_{\kappa_t\to 0}\frac{\Delta t\cos\kappa_t\Delta t}{\Delta t} = 1 &(68)\\ \lim_{\kappa_t\to 0}\frac{2(1-\cos\kappa_t\Delta t)}{\kappa_t^2(\Delta t)^2} &= \lim_{\kappa_t\to 0}\frac{2\Delta t\sin\kappa_t\Delta t}{2\kappa_t(\Delta t)^2} = 1 &(69)\end{aligned}$$

which leads to $f_{t,a}^{(INT)} \to 0$ as expected for a debonded interface. But as $D_t \to \infty$, $f_{t,a}^{(INT)}$ oscillates at increasing frequency. The issue is that MPM analysis cannot handle frequencies beyond its current resolution. Some calculations suggested stability requires $\kappa_t\Delta t < 1$ (or something on that order and certainly must have $\kappa_t\Delta t < \pi/2$).

This observed stability condition is

$$\Delta t < \frac{1}{\kappa_t} = \sqrt{\frac{m_{red}}{D_t A}} \tag{70}$$

These values vary in a simulation, but maximum $m_{red} \sim m_i/2 = \rho V_i/2 = \rho\Delta x^3/2$ and $A \sim \Delta x^2$. Thus stability requires

$$\Delta t < \sqrt{\frac{\rho\Delta x^3}{2D_t\Delta x^2}} = \Delta x\sqrt{\frac{\rho}{2D_t\Delta x}} = C_{int}\frac{\Delta x}{v_{int}} \tag{71}$$

where C_{int} is a CFL factor for interfaces and v_{int} is an effective wave speed for an interface and given by:

$$v_{int} = \sqrt{\frac{D_t\Delta x}{\rho}} \tag{72}$$

Here ρ would be minimum density of the materials in contact. This velocity depends on grid spacing. Imagine simulation with time step $\Delta t = C\Delta x/v$ (where C is CFL factor and v is maximum material wave speed), the maximum D_t that can be simulated would be

$$v_{int} < \frac{C_{int}v}{C}, \quad \sqrt{\frac{D_t\Delta x}{\rho}} < \frac{C_{int}v}{C}, \quad \text{or} \quad D_t < \frac{\rho C_{int}^2 v^2}{C^2\Delta x} \tag{73}$$

Thus, to simulate high D_t, need to have sufficiently low Δx and/or C. To allow any D_t in a simulation, each calculation can check $\kappa_t\Delta t$. If it is larger than 1 (or some number between 0 and $\pi/2$ likely near 1), the calculation can revert to stick conditions. In other words, the calculation is converted to infinite D_t. If the conversion to stick occurs when D_t has not reached the stick

limit, a better simulation would need to increase resolution or decrease time step. A simulation could pre-screen interface laws before starting a calculation, but if that screening was based on $m_{red} \sim m_i/2$, the simulation should still become unstable at node with smaller m_{red}. It is better to check stability on each interface calculation.

The total interfacial energy is

$$E_{int} = \int_0^{\delta_t(\Delta t)} D_t A x \, dx = \frac{1}{2} D_t A \big(\delta_t(\Delta t)\big)^2 \tag{74}$$

For incremental energy, the results are:

$$\begin{aligned} \Delta E_{int} &= D_t A \int_0^{\Delta t} \delta_t(t) \frac{d\delta_t(t)}{dt} \, dt = \frac{1}{2} D_t A \int_0^{\Delta t} \frac{d(\delta_t(t))^2}{dt} \, dt & (75) \\ &= \frac{1}{2} D_t A \, (\delta_t(t))^2 \big|_0^{\Delta t} = \frac{1}{2} D_t A \left((\delta_t(\Delta t))^2 - (\delta_t(0))^2 \right) & (76) \end{aligned}$$

Summarizing linear interfaces, set $\phi = \kappa_t \Delta t$ (dimensionless) and $m = m_{red}/\Delta t$ (mass/time), then:

$$\begin{aligned} f_{t,a}^{(INT)} &= \frac{2m\delta_t(0)\,(1-\cos\phi)}{\Delta t} + \frac{2d_t}{\Delta t}\left(1 - \frac{\sin\phi}{\phi}\right) + f_t\left(1 - \frac{2(1-\cos\phi)}{\phi^2}\right) & (77) \\ \delta_t(\Delta t) &= \delta_t(0)\cos\phi + \frac{d_t}{m}\frac{\sin\phi}{\phi} + \frac{f_t \Delta t\,(1-\cos\phi)}{m\phi^2} & (78) \end{aligned}$$

The final discontinuity can be used in either total or incremental energies above. When calculations are done after updating the momenta:

$$d_t = d_t' - f_t \Delta t \qquad \text{where} \qquad d_t' = \Delta \boldsymbol{p}_a' \cdot \hat{t} \tag{79}$$

and $\Delta \boldsymbol{p}_a'$ is the update momentum term. The force and displacement become:

$$\begin{aligned} f_{t,a}^{(INT)} &= \frac{2m\delta_t(0)\,(1-\cos\phi)}{\Delta t} + \frac{2d_t'}{\Delta t}\left(1 - \frac{\sin\phi}{\phi}\right) + f_t\left(\frac{2\sin\phi}{\phi} - \frac{2(1-\cos\phi)}{\phi^2} - 1\right) & (80) \\ \delta_t(\Delta t) &= \delta_t(0)\cos\phi + \frac{d_t'}{m}\frac{\sin\phi}{\phi} + \frac{f_t \Delta t}{m}\left(\frac{1-\cos\phi}{\phi^2} - \frac{\sin\phi}{\phi}\right) & (81) \end{aligned}$$

6.2.1 Numerical Implementation

The options for implementation are:

1. The interface calculations can be done after the mass and momentum calculation, which is before forces are known. We use the above analysis with $f_t = 0$. This approach is correct to first order in time step. It has been used in `OSParticulas` with good results.

2. Alternatively, the calculations can be done after the mass and momentum update and then add the second order terms involving f_t. New calculations suggest the post-update method is better by a small amount, which corresponds to it being a second order calculation of interfacial force. It is the current `OSParticulas` method.

3. Attempts to combine the first two (*i.e.*, update momenta by interfacial force in mass and momentum update for input to constitutive laws, and then again after momentum and add energy calculations) gives very poor results. It is not clear why (because first phase momenta and restored prior to update), but it is a very large effect (and poor results). In other words, used metho 1 or 2, but no combination of the methods.

4. Incremental energy calculations are very bad. Again, the reasons are not certain but perhaps are caused by uncertainty in finding initial separation. In other words, subtracting $\delta_t(0)^2$ does not compensate enough for actual initial energy and accumulation of errors grow with time step. The calculations with the two layer example are probably a worst-case problem because interfacial energy is small overall.

5. By either approach, special handling is needed for stiff interfaces or for when $\phi = \kappa_t \Delta t$ is too large. Interface calculations are assumed required to screen out high frequency interfaces by switching to stick conditions whenever $\phi > \phi_{max}$. The results do not appear very sensitive to cutoff for values of $\phi_{max} < \pi/2$, although $\phi_{max} = \pi$ shows instabilities. Current code uses $\phi < \pi/2$. When $\phi > \phi_{max}$, the interface uses stick conditions (by leaving stick momentum in the input $\Delta \boldsymbol{p}_a$ and by setting interface force and energy to zero).

6.3 Compare to Old Method

The interface force in my paper was calculated using only the initial discontinuity or:

$$f_{t,a}^{(INT,0)} = D_t A \delta_t(0) = m_{red} \kappa_t^2 \delta_t(0) = \frac{m \delta_t(0) \phi^2}{\Delta t} \tag{82}$$

This is identical to the $\delta(0)$ term in harmonic oscillator analysis by expanding $(1 - \cos\phi)$ to two terms. The old method limited forces to value equal to mean of high-frequency forces, but it may not be best way to limit forces. The limit based on ϕ appears better.

If above is considered zeroth order, maybe improved result, and one that would work for non-linear laws, could be to use the zeroth order force to estimate change in discontinuity during the time step. Substituting the force:

$$\delta_t(t) = \delta_t(0) \left(1 - \frac{\phi^2 t^2}{(\Delta t)^2}\right) + \frac{d_t t}{m_{red}} + \frac{f_t t^2}{2 m_{red}} \tag{83}$$

The mean displacement during the time step is:

$$\frac{1}{\Delta t} \int_0^{\Delta t} \delta_t(t) dt = \delta_t(0) \left(1 - \frac{\phi^2}{6}\right) + \frac{d_t}{2m} + \frac{f_t \Delta t}{6m} \tag{84}$$

A force based on mean displacement is:

$$f_{t,a}^{(INT,1)} = \frac{2m\delta_n(0)}{\Delta t} \left(\frac{\phi^2}{2} - \frac{\phi^4}{12}\right) + \frac{2d_t}{\Delta t} \left(\frac{\phi^2}{4}\right) + f_t \left(\frac{\phi^2}{6}\right) \tag{85}$$

A result that plots better is

$$f_{t,a}^{(INT,1')} = \frac{f_{t,a}^{(INT,0)} + f_{t,a}^{(INT,1)}}{2} = \frac{2m\delta_n(0)}{\Delta t} \left(\frac{\phi^2}{2} - \frac{\phi^4}{24}\right) + \frac{2d_t}{\Delta t} \left(\frac{\phi^2}{8}\right) + f_t \left(\frac{\phi^2}{12}\right) \tag{86}$$

Using updated force, the displacement is

$$\delta_t(t) = \delta_t(0) \left(1 - \frac{t^2}{(\Delta t)^2} \left(\frac{\phi^2}{2} - \frac{\phi^4}{24}\right)\right) + \frac{d_t t}{m_{red}} \left(1 - \frac{\phi^2 t}{8 \Delta t}\right) + \frac{f_t t^2}{2 m_{red}} \left(1 - \frac{\phi^2}{12}\right) \tag{87}$$

I tried another round, but it did not improve the results. The above first order looks like a good option to full solution.

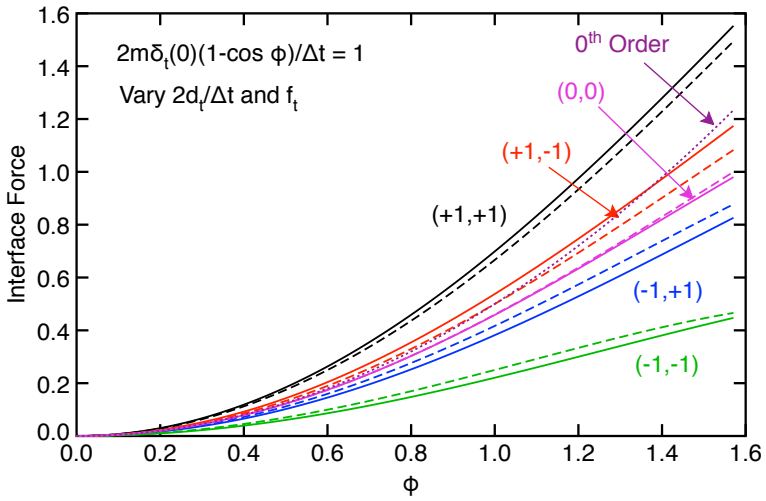

Figure 4: Plot of sample force curves for $2m\delta_n(0)\phi^2/\Delta t = 1$ and various values of $(2d_t/\Delta t,\ f_t)$ as indicated. The zeroth order result does not depend on d_t or f_t and is therefore a single curve.

Figure 4 compares first order force, $f_{t,a}^{(INT,1')}$, to full method found by solving differential equation over the full range of ϕ allowed for stability.. The curves give interface force for several relative values of $\delta_t(0)$, d_t, and f_t. For a wide range of values, the approximate method is always close to the full harmonic oscillator solution. The two solutions are virtually identical for $d_t,\ f_t << \delta_t(0)$ (*i.e.*, the (0,0) curves). The solutions are also always when $\phi \sim 0$ as well (*i.e.*, compliant interface with low interface forces). Note that old zeroth method is the dotted line. It falls within all data, is good for compliant interface and for $d_t,\ f_t << \delta_t(0)$ provided $\phi < 1$.

The final results for implementation (based on updated momenta and using position at end from final force) are:

$$f_{t,a}^{(INT,1')} = \frac{m\delta_n(0)\phi^2}{\Delta t}\left(1-\frac{\phi^2}{12}\right)+\frac{d_t'}{\Delta t}\left(\frac{\phi^2}{4}\right)-f_t\left(\frac{\phi^2}{6}\right) \tag{88}$$

$$\delta_t(\Delta t) = \delta_t(0)\left(1-\frac{\phi^2}{2}\left(1-\frac{\phi^2}{12}\right)\right)+\frac{d_t'}{m}\left(1-\frac{\phi^2}{8}\right)-\frac{f_t\Delta t}{m}\left(\frac{1}{2}-\frac{\phi^2}{12}\right) \tag{89}$$

6.3.1 Extension to Non-Linear Laws

The revised old method in previous section is OK, but could just use full harmonic oscillator analysis instead. For non-linear laws, however, the differential equation probably cannot be solved and the method of the previous section could guide an approximate method. Repeating the steps for arbitrary $F_t(\delta(t))$ lead to:

1. Define incremental frequency as:

$$\kappa_t = \sqrt{\frac{|F_t'(\delta_t(0))|}{m_{red}}} \qquad \text{and} \qquad \phi = \kappa_t\Delta t \tag{90}$$

If $\phi > \pi/2$ (or maybe a lower cutoff) proceed with stick conditions. But, if $F_t'(\delta_t(0)) < 0$ (*i.e.*, law with softening region), the linearized differential equation would involve exponential

terms that would blow up for large ϕ. In this case, the physical interpretation is that high frequency corresponds to failure and the interface should be treated as debonded (both force and momentum change set to zero). For low ϕ, should be possible to stably model softening.

2. Find the zeroth order force, $f_{t,a}^{(INT,0)} = F_t(\delta(0))$, and calculate displacement discontinuity and its mean value:

$$\delta_t(t) = \delta_t(0) - \frac{F_t(\delta_t(0))t^2}{2m_{red}} + \frac{d_t t}{m_{red}} + \frac{f_t t^2}{2m_{red}} \tag{91}$$

$$\left\langle \delta_t^{(1)} \right\rangle = \delta_t(0) - \frac{F_t(\delta_t(0))\Delta t}{6m} + \frac{d_t}{2m} + \frac{f_t \Delta t}{6m} \tag{92}$$

3. Find force based on updated mean displacement

$$f_{t,a}^{(INT,1)} = F_t\left(\left\langle \delta_t^{(1)} \right\rangle\right) \tag{93}$$

Also check stability by finding $F_t'\left(\left\langle \delta_t^{(1)} \right\rangle\right)$ and switch to stick (or debonding if softening) if needed.

4. Find adjusted force (this combination was based on best approach for linear problem and one that give exact first two Taylor series terms for the $\cos\phi$ term):

$$f_{t,a}^{(INT,1')} = \frac{F_t(\delta(0)) + F_t\left(\left\langle \delta_t^{(1)} \right\rangle\right)}{2} \tag{94}$$

5. Find displacement and end of time step based on adjusted force:

$$\delta_t(\Delta t) = \delta_t(0) - \frac{f_{t,a}^{(INT,1')}\Delta t}{2m} + \frac{d_t}{m} + \frac{f_t \Delta t}{2m} \tag{95}$$

Also check stability by finding $F_t'(\delta_t(\Delta t))$ and switch to stick (or debonding if softening) if needed.

6. Calculate total energy based on final discontinuity (note this approach only allows elastic laws and here would find area under curve up to final displacement discontinuity. Energy is only needed for other calculations. It works for composite analysis, but not anything else.

7. Summarizing and using post-update terms:

$$\left\langle \delta_t^{(1)} \right\rangle = \delta_t(0) - \frac{F_t(\delta_t(0))\Delta t}{6m} + \frac{d_t'}{2m} - \frac{f_t \Delta t}{3m} \tag{96}$$

$$f_{t,a}^{(INT,1')} = \frac{F_t(\delta(0)) + F_t\left(\left\langle \delta_t^{(1)} \right\rangle\right)}{2} \tag{97}$$

$$\delta_t(\Delta t) = \delta_t(0) - \frac{f_{t,a}^{(INT,1')}\Delta t}{2m} + \frac{d_t'}{m} - \frac{f_t \Delta t}{2m} \tag{98}$$

with stability checks done for $\delta(0)$, $\left\langle \delta_t^{(1)} \right\rangle$, and $\delta_t(\Delta t)$, although maybe mean one can be skipped. Implementation could use these equations, or just stope with zeroth order force and displacements (or an option to pick). Class to implement any non-linear law only needs to provide $F_t(\delta)$, $F_t'(\delta)$, and methods to check stability (for stick, stable, or debonding) and to find energy.

6.4 Bilinear Imperfect Inteface

The linear interface contact law in `OSParticulas` is bilinear in the normal direction (and therefore nonlinear) with different interface parameters in tension (D_{nt}) and compression (D_{nc}). We start with the interfaces initially separated or $\delta_n(0) > 0$. The situation evolves as a linear interface and remains that way as long as $\delta_n(t)$ remains positive. Assuming all linear, final separation is

$$\delta_t(\Delta t) = \frac{d_n}{m_{red}\kappa_{nt}} \sin \kappa_{nt}\Delta t + \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right) \cos \kappa_{nt}\Delta t + \frac{f_n}{m_{red}\kappa_{nt}^2} \quad \text{with} \quad \kappa_{nt} = \sqrt{\frac{D_{nt}A}{m_{red}}} \tag{99}$$

If $\delta_t(\Delta t) > 0$, use fully linear analysis in previous section (using normal direction parameters).

6.4.1 Intrastep Contact

If $\delta_t(\Delta t)$ from above is negative, the separation becomes zero during the time step at $0 < t_c < \Delta t$ found by solving

$$0 = \frac{d_n}{m_{red}\kappa_{nt}} \sin \kappa_{nt}t_c + \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right) \cos \kappa_{nt}t_c + \frac{f_n}{m_{red}\kappa_{nt}^2} \tag{100}$$

Two forms useful below are:

$$\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2} = -\frac{d_n}{m_{red}\kappa_{nt}} \tan \kappa_{nt}t_c - \frac{f_n}{m_{red}\kappa_{nt}^2} \sec \kappa_{nt}t_c \tag{101}$$

$$\frac{d_n}{m_{red}\kappa_{nt}} = -\left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right) \cot \kappa_{nt}t_c - \frac{f_n}{m_{red}\kappa_{nt}^2} \csc \kappa_{nt}t_c \tag{102}$$

Based on stability arguments, we know that $0 < \kappa_{nt}t_c < \pi/2$ and therefore both $\sin \kappa_{nt}t_c$ and $\cos \kappa_{nt}t_c$ are positive. We can solve (with $d_n = d_n' - f_n\Delta t$) as follows:

$$-\left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right) \cos \kappa_{nt}t_c = \frac{d_n}{m_{red}\kappa_{nt}} \sin \kappa_{nt}t_c + \frac{f_n}{m_{red}\kappa_{nt}^2} \tag{103}$$

$$-\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right) \cos \kappa_{nt}t_c = \kappa_{nt}d_n \sin \kappa_{nt}t_c + f_n \tag{104}$$

$$\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 \left(1 - \sin^2 \kappa_{nt}t_c\right) = \kappa_{nt}^2 d_n^2 \sin^2 \kappa_{nt}t_c + 2\kappa_{nt}f_n d_n \sin \kappa_{nt}t_c + f_n^2 \tag{105}$$

or a quadratic equation

$$0 = a \sin^2 \kappa_{nt}t_c + b \sin \kappa_{nt}t_c + c \tag{106}$$

$$a = \left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 + \kappa_{nt}^2 d_n^2 \tag{107}$$

$$b = 2\kappa_{nt}f_n d_n \tag{108}$$

$$c = f_n^2 - \left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 \tag{109}$$

The solution is positive terms from:

$$\sin \kappa_{nt}t_c = \frac{-\kappa_{nt}f_n d_n \pm \left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right) \sqrt{\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 + \kappa_{nt}^2 d_n^2 - f_n^2}}{\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 + \kappa_{nt}^2 d_n^2} \tag{110}$$

$$\cos \kappa_{nt}t_c = \frac{-f_n \left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right) \mp \kappa_{nt}d_n \sqrt{\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 + \kappa_{nt}^2 d_n^2 - f_n^2}}{\left(m_{red}\kappa_{nt}^2\delta_n(0) - f_n\right)^2 + \kappa_{nt}^2 d_n^2} \tag{111}$$

In pre-update calculations (when $f_n = 0$ and therefore $d_n < 0$ for all contact cases), a simpler solution is

$$\tan \kappa_{nt} t_c = \frac{-m_{red}\kappa_{nt}\delta_n(0)}{d_n} \qquad \text{or} \qquad t_c = \frac{1}{\kappa_{nt}} \tan^{-1}\left(\frac{-m_{red}\kappa_{nt}\delta_n(0)}{d_n}\right) \tag{112}$$

Or from general solution with $f_n = 0$

$$\sin \kappa_{nt} t_c = \frac{m_{red}\kappa_{nt}\delta_n(0)}{\sqrt{\left(m_{red}\kappa_{nt}\delta_n(0)\right)^2 + d_n^2}} \qquad \text{and} \qquad \cos \kappa_{nt} t_c = \frac{-d_n}{\sqrt{\left(m_{red}\kappa_{nt}\delta_n(0)\right)^2 + d_n^2}} \tag{113}$$

6.4.2 Intrastep Contact of Debonded Interface

If the normal direction is debonde, contact occurs when:

$$0 = \frac{f_n t_c^2}{2m_{red}} + \frac{d_n t_c}{m_{red}} + \delta_n(0) \tag{114}$$

which is a new quadratic equation for t_c. If almost debonded ($\kappa_{nt} t_c$ small) and $f_n \neq 0$:

$$0 = a t_c^2 + b t_c + c \tag{115}$$

$$a = \frac{f_n}{2m_{red}} + \left(\frac{d_n^2}{2m_{red} f_n} - \delta_n(0)\left(1 - \frac{m_{red}\delta_n(0)}{2f_n}\kappa_{nt}^2\right)\right)\kappa_{nt}^2 \tag{116}$$

$$b = \frac{d_n}{m_{red}} \tag{117}$$

$$c = \delta_n(0)\left(1 - \frac{m_{red}\delta_n(0)}{2f_n}\kappa_{nt}^2\right) \tag{118}$$

If almost debonded ($\kappa_{nt} t_c$ small) and $f_n = 0$:

$$0 = a t_c^2 + c, \qquad a = \frac{d_n^2}{m_{red}^2} + \kappa_{nt}^2 \delta_n(0)^2, \qquad c = \delta_n(0)^2 \tag{119}$$

$$t_c = -\frac{\delta_n(0)}{\sqrt{\frac{d_n^2}{m_{red}^2} + \kappa_{nt}^2 \delta_n(0)^2}} \tag{120}$$

6.4.3 Post Contact When Perfect in Compression

If compression is a perfect interface, the calculations stop when contact occurs. The effective force distributed of the entire time step is

$$f_{n,a}^{(INT)} = \frac{2m_{red}\delta_n(0)}{(\Delta t)^2} + \frac{2d_n}{\Delta t} + f_n \tag{121}$$

The energy is zero. For interface discontinuity to remain in contact (*i.e.*, zero) after coming into contact, a momentum change is applied (for post-contact stick) equal to:

$$d_n^{(c)} = m_{red}\frac{d\delta_n(t_c)}{dt} = d_n \cos \kappa_{nt} t_c - m_{red}\kappa_{nt}\delta_n(0) \sin \kappa_{nt} t_c + f_n t_c \frac{\sin \kappa_{nt} t_c}{\kappa_{nt} t_c} \tag{122}$$

In post-contact calculations in terms of $d'_n = d_n + f_n \Delta t$, the results are:

$$f_{n,a}^{(INT)} = \frac{2m\delta_n(0)}{\Delta t} + \frac{2d'_n}{\Delta t} - f_n \tag{123}$$

$$d_n^{(c)} = d'_n \cos\kappa_{nt} t_c - m_{red}\kappa_{nt}\delta_n(0)\sin\kappa_{nt} t_c + f_n t_c \left(\frac{\sin\kappa_{nt} t_c}{\kappa_{nt} t_c} - \frac{\Delta t}{t_c}\cos\kappa_{nt} t_c\right) \tag{124}$$

For debonded interface prior to contact

$$d_n^{(c)} = m_{red}\frac{d\delta_n(t_c)}{dt} = d_n + f_n t_c = d'_n + f_n(t_c - \Delta t) \tag{125}$$

6.4.4 Post Contact with new Interface Parameter

The post-contact displacement discontinuity and condition of zero discontinuity are:

$$\begin{aligned}\delta_n(t) = {} & \delta(0) + \frac{d_n t}{m_{red}} + \frac{f_n t^2}{2m_{red}} - \frac{1}{m_{red}}\int_0^{t_c} dt_2 \int_0^{t_1} dt_1 D_{nt} A \delta_n(t_1) \\ & - \frac{1}{m_{red}}\int_{t_c}^{t} dt_2 \left(\int_0^{t_c} dt_1 D_{nt} A\delta_n(t_1) + \int_{t_c}^{t_2} dt_1 D_{nc} A\delta_n(t_1)\right)\end{aligned} \tag{126}$$

$$\delta_n(t_c) = 0 = \delta(0) + \frac{d_n t_c}{m_{red}} + \frac{f_n t_c^2}{2m_{red}} - \frac{1}{m_{red}}\int_0^{t_c} dt_2 \int_0^{t_1} dt_1 D_{nt} A\delta_n(t_1) \tag{127}$$

Subtracting the second equation (which is zero) from the first gives:

$$\begin{aligned}\delta_n(t) &= \frac{d_n(t-t_c)}{m_{red}} + \frac{f_n(t^2 - t_c^2)}{2m_{red}} - \frac{1}{m_{red}}\int_{t_c}^{t} dt_2 \left(\int_0^{t_c} dt_1 D_{nt} A\delta_n(t_1) + \int_{t_c}^{t_2} dt_1 D_{nc} A\delta_n(t_1)\right) \\ &= \frac{d_n(t-t_c)}{m_{red}} + \frac{f_n(t^2 - t_c^2)}{2m_{red}} - \frac{t-t_c}{m_{red}}\int_0^{t_c} dt_1 D_{nt} A\delta_n(t_1) - \frac{1}{m_{red}}\int_{t_c}^{t} dt_2 \int_{t_c}^{t_2} dt_1 D_{nc} A\delta_n(t_1) \\ &= \frac{d_n^{(c)}(t-t_c)}{m_{red}} + \frac{f_n(t-t_c)^2}{2m_{red}} - \frac{1}{m_{red}}\int_{t_c}^{t} dt_2 \int_{t_c}^{t_2} dt_1 D_{nc} A\delta_n(t_1)\end{aligned} \tag{128}$$

where $d_n^{(c)}$ is the component of momentum change required to stick after coming into contact and therefore $d_n^{(c)}/m_{red}$ is the interfacial separation velocity at the time of contact. It can be found two ways:

$$\frac{d_n^{(c)}}{m_{red}} = \frac{d\delta_t(t_c)}{dt} = \frac{d_n}{m_{red}}\cos\kappa_{nt} t_c - \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right)\kappa_{nt}\sin\kappa_{nt} t_c \tag{129}$$

$$= \frac{d_n}{m_{red}} + \frac{f_n t_c}{m_{red}} - \frac{D_{nt} A}{m_{red}}\int_0^{t_c} dt_1 \delta_n(t_1) \tag{130}$$

$$\begin{aligned}= {} & \frac{d_n}{m_{red}} + \frac{f_n t_c}{m_{red}} \\ & - \frac{D_{nt} A}{m_{red}\kappa_{nt}^2}\left(-\frac{d_n}{m_{red}}\cos\kappa_{nt} t_1 + \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right)\kappa_{nt}\sin\kappa_{nt} t_1 + \frac{f_n t}{m_{red}}\right)\Bigg|_0^{t_c}\end{aligned} \tag{131}$$

which leads to identical result. If using post-update terms then

$$\frac{d_n^{(c)}}{m_{red}} = \frac{d'_n - f_n\Delta t}{m_{red}}\cos\kappa_{nt} t_c - \delta_n(0)\kappa_{nt}\sin\kappa_{nt} t_c + \frac{f_n}{m_{red}\kappa_{nt}}\sin\kappa_{nt} t_c \tag{132}$$

$$= \frac{d'_n}{m_{red}}\cos\kappa_{nt} t_c - \delta_n(0)\kappa_{nt}\sin\kappa_{nt} t_c + \frac{f_n t_c}{m_{red}}\left(\frac{\sin\kappa_{nt} t_c}{\kappa_{nt} t_c} - \frac{\Delta t}{t_c}\cos\kappa_{nt} t_c\right) \tag{133}$$

We next shift the time axis to $t' = t - t_c$, to get

$$\delta_n(t') = \frac{d_n^{(c)} t'}{m_{red}} + \frac{f_n t'^2}{2m_{red}} - \frac{1}{m_{red}} \int_0^{t'} dt'_2 \int_0^{t_2} dt'_1 D_{nc} A \delta_n(t'_1) \tag{134}$$

The general solution to the differential equation is

$$\delta_t(t') = \frac{f_n}{m_{red}\kappa_{nc}^2} + C_1 \sin \kappa_{nc} t' + C_2 \cos \kappa_{nc} t' \qquad \text{where} \qquad \kappa_{nc} = \sqrt{\frac{D_{nc}A}{m_{red}}} \tag{135}$$

The initial conditions are

$$\delta_t(0) = 0 = \frac{f_n}{m_{red}\kappa_{nc}^2} + C_2 \qquad \text{and} \qquad \frac{\delta_t(0)}{dt} = C_1 \kappa_{nc} = \frac{d_n^{(c)}}{m_{red}} \tag{136}$$

The final solution becomes

$$\delta_t(t') = \frac{d_n^{(c)}}{m_{red}\kappa_{nc}} \sin \kappa_{nc} t' - \frac{f_n}{m_{red}\kappa_{nc}^2} \cos \kappa_{nc} t' + \frac{f_n}{m_{red}\kappa_{nc}^2} \tag{137}$$

$$\delta_t(t) = \frac{d_n^{(c)}}{m_{red}\kappa_{nc}} \sin \kappa_{nc}(t - t_c) - \frac{f_n}{m_{red}\kappa_{nc}^2} \cos \kappa_{nc}(t - t_c) + \frac{f_n}{m_{red}\kappa_{nc}^2} \tag{138}$$

The effective interface force for entire step becomes and final discontinuity:

$$f_{n,a}^{(INT)} = \frac{2m\delta_n(0)}{\Delta t} + \frac{2}{\Delta t}\left(d_n - d_n^{(c)} \frac{\sin \kappa_{nc}(\Delta t - t_c)}{\kappa_{nc}\Delta t}\right) + f_n\left(1 - \frac{2\left(1 - \cos \kappa_{nc}(\Delta t - t_c)\right)}{\kappa_{nc}^2 (\Delta t)^2}\right) \tag{139}$$

$$\delta_t(\Delta t) = \frac{d_n^{(c)}}{m_{red}} \frac{\sin \kappa_{nc}(\Delta t - t_c)}{\kappa_{nc}} + \frac{f_n}{m_{red}}\left(\frac{1 - \cos \kappa_{nc}(\Delta t - t_c)}{\kappa_{nc}^2}\right) \tag{140}$$

If the compression interface is debonded, a special case is

$$\delta_n(\Delta t) = \frac{d_n^{(c)}(\Delta t - t_c)}{m_{red}} + \frac{f_n(\Delta t - t_c)^2}{2m_{red}} \tag{141}$$

$$f_{n,a}^{(INT)} = \frac{2}{\Delta t}\left(m\delta_n(0) + d_n - \frac{d_n^{(c)}(\Delta t - t_c)}{\Delta t}\right) + f_n\left(1 - \frac{(\Delta t - t_c)^2}{(\Delta t)^2}\right) \tag{142}$$

6.4.5 Expand to Initial Time Step Values

Expanding the trigonometric terms:

$$\begin{aligned}\delta_t(t') = {} & \frac{d_n^{(c)}}{m_{red}\kappa_{nc}}\left(\sin \kappa_{nc} t \cos \kappa_{nc} t_c - \sin \kappa_{nc} t_c \cos \kappa_{nc} t\right) \\ & - \frac{f_n}{m_{red}\kappa_{nc}^2}\left(\cos \kappa_{nc} t \cos \kappa_{nc} t_c + \sin \kappa_{nc} t \sin \kappa_{nc} t_c\right) + \frac{f_n}{m_{red}\kappa_{nc}^2}\end{aligned} \tag{143}$$

which simplifies to:

$$\delta_t(t) = \frac{d_n^{(c)} \cos \kappa_{nc} t_c - \frac{f_n \sin \kappa_{nc} t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} \sin \kappa_{nc} t - \frac{d_n^{(c)} \sin \kappa_{nc} t_c + \frac{f_n \cos \kappa_{nc} t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} \cos \kappa_{nc} t + \frac{f_n}{m_{red}\kappa_{nc}^2} \tag{144}$$

The coefficients simplify with the contact conditions as follows:

$$\frac{d_n^{(c)}\cos\kappa_{nc}t_c - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} = \left[\frac{d_n}{m_{red}}\cos\kappa_{nt}t_c - \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right)\kappa_{nt}\sin\kappa_{nt}t_c\right]\frac{\cos\kappa_{nc}t_c}{\kappa_{nc}} - \frac{f_n\sin\kappa_{nc}t_c}{m_{red}\kappa_{nc}^2} \tag{145}$$

$$= \left[\frac{d_n}{m_{red}}\cos\kappa_{nt}t_c + \left(\frac{d_n}{m_{red}}\tan\kappa_{nt}t_c + \frac{f_n}{m_{red}\kappa_{nt}}\sec\kappa_{nt}t_c\right)\sin\kappa_{nt}t_c\right]\frac{\cos\kappa_{nc}t_c}{\kappa_{nc}} - \frac{f_n\sin\kappa_{nc}t_c}{m_{red}\kappa_{nc}^2}$$

$$= \frac{1}{m_{red}\kappa_{nc}}\left[\left(d_n + \frac{f_n\sin\kappa_{nt}t_c}{\kappa_{nt}}\right)\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}\right] \tag{146}$$

and

$$\frac{d_n^{(c)}\sin\kappa_{nc}t_c + \frac{f_n\cos\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} = \left[\frac{d_n}{m_{red}}\cos\kappa_{nt}t_c - \left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2}\right)\kappa_{nt}\sin\kappa_{nt}t_c\right]\frac{\sin\kappa_{nc}t_c}{\kappa_{nc}} + \frac{f_n\cos\kappa_{nc}t_c}{m_{red}\kappa_{nc}^2} \tag{147}$$

$$= -\left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2} + \frac{f_n\cos\kappa_{nt}t_c}{m_{red}\kappa_{nt}^2}\right)\frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c} + \frac{f_n\cos\kappa_{nc}t_c}{m_{red}\kappa_{nc}^2} \tag{148}$$

If the problem was linear ($\kappa_{nt} = \kappa_{nc} = \kappa_n$) then

$$\frac{d_n^{(c)}\cos\kappa_{nc}t_c - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} = \frac{d_n}{\kappa_n m_{red}} \tag{149}$$

$$\frac{d_n^{(c)}\sin\kappa_{nc}t_c + \frac{f_n\cos\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}} = -\left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_n^2}\right) \tag{150}$$

These results recover the linear result as it should.

The net effective force is found from

$$\delta_t(\Delta) = \frac{d_n^{(c)}\cos\kappa_{nc}t_c - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}}\sin\kappa_{nc}\Delta t - \frac{d_n^{(c)}\sin\kappa_{nc}t_c + \frac{f_n\cos\kappa_{nc}t_c}{\kappa_{nc}}}{m_{red}\kappa_{nc}}\cos\kappa_{nc}\Delta t + \frac{f_n}{m_{red}\kappa_{nc}^2} = \delta_n(0) + \frac{d_n\Delta t}{m_{red}} + \frac{f_n(\Delta t)^2}{2m_{red}} - \frac{f_{n,a}^{(INT)}(\Delta t)^2}{2m_{red}} \tag{151}$$

$$f_{n,a}^{(INT)} = f_n + \frac{2}{\Delta t}\left[d_n\left(1 - \frac{d_n^{(c)}\cos\kappa_{nc}t_c - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}}{d_n}\frac{\sin\kappa_{nc}\Delta t}{\kappa_{nc}\Delta t}\right) - \frac{f_n}{\kappa_{nc}^2\Delta t} + \frac{\delta_n(0)m_{red}}{\Delta t}\left(1 + \frac{d_n^{(c)}\sin\kappa_{nc}t_c + \frac{f_n\cos\kappa_{nc}t_c}{\kappa_{nc}}}{\delta_n(0)m_{red}\kappa_{nc}}\cos\kappa_{nc}\Delta t\right)\right] \tag{152}$$

This result reduces to linear result when $\kappa_{nt} = \kappa_{nc} = \kappa_n$. Inserting revised coefficients for bilinear gives:

$$f_{n,a}^{(INT)} = f_n + \frac{2}{\Delta t}\left[\left(d_n - \left[\left(d_n + \frac{f_n\sin\kappa_{nt}t_c}{\kappa_{nt}}\right)\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}\right]\frac{\sin\kappa_{nc}\Delta t}{\kappa_{nc}\Delta t}\right) - \frac{f_n}{\kappa_{nc}^2\Delta t} + \left(\frac{m_{red}\delta_n(0)}{\Delta t} - \left[\left(\frac{m_{red}\delta_n(0)}{\Delta t} - \frac{f_n}{\kappa_{nt}^2\Delta t}(1 - \cos\kappa_{nt}t_c)\right)\frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c} - \frac{f_n\cos\kappa_{nc}t_c}{\kappa_{nc}^2\Delta t}\right]\cos\kappa_{nc}\Delta t\right)\right] \tag{153}$$

In pre-update when $f_n = 0$, this results reduces to:

$$f_{n,a}^{(INT)} = \frac{2}{\Delta t}\left[d_n\left(1 - \frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c}\frac{\sin\kappa_{nc}\Delta t}{\kappa_{nc}\Delta t}\right) + \frac{\delta_n(0)m_{red}}{\Delta t}\left(1 - \frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c}\cos\kappa_{nc}\Delta t\right)\right] \quad (154)$$

Get f_n term by setting $d_n = \delta_n(0) = 0$ or

$$\begin{aligned} f_{n,a}^{(INT)} = f_n\Bigg[1 - \frac{2}{\Delta t}\left(\frac{\sin\kappa_{nt}t_c}{\kappa_{nt}}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{\sin\kappa_{nc}t_c}{\kappa_{nc}}\right)\frac{\sin\kappa_{nc}\Delta t}{\kappa_{nc}\Delta t} \\ - \frac{2}{\kappa_{nc}^2(\Delta t)^2}\left(1 - \left((1-\cos\kappa_{nt}t_c)\frac{\kappa_{nc}\sin\kappa_{nc}t_c}{\kappa_{nt}\sin\kappa_{nt}t_c} + \cos\kappa_{nc}t_c\right)\cos\kappa_{nc}\Delta t\right)\Bigg] \end{aligned} \quad (155)$$

For energy calculations, we can find

$$\begin{aligned} \delta_t(\Delta) &= \frac{1}{m_{red}\kappa_{nc}}\left[\left(d_n + \frac{f_n\sin\kappa_{nt}t_c}{\kappa_{nt}}\right)\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{f_n\sin\kappa_{nc}t_c}{\kappa_{nc}}\right]\sin\kappa_{nc}\Delta t + \frac{f_n}{m_{red}\kappa_{nc}^2} \\ &+ \left[\left(\delta_n(0) - \frac{f_n}{m_{red}\kappa_{nt}^2} + \frac{f_n\cos\kappa_{nt}t_c}{m_{red}\kappa_{nt}^2}\right)\frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c} - \frac{f_n\cos\kappa_{nc}t_c}{m_{red}\kappa_{nc}^2}\right]\cos\kappa_{nc}\Delta t \end{aligned} \quad (156)$$

In pre-update when $f_n = 0$, this results reduces to:

$$\delta_t(\Delta t) = \frac{d_n}{m_{red}\kappa_{nc}}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c}\sin\kappa_{nc}\Delta t + \delta_n(0)\frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c}\cos\kappa_{nc}\Delta t \quad (157)$$

Get f_n term by setting $d_n = \delta_n(0) = 0$ or

$$\begin{aligned} m_{red}\delta_t(\Delta) &= f_n\Bigg[\left(\frac{\sin\kappa_{nt}t_c}{\kappa_{nt}}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{\sin\kappa_{nc}t_c}{\kappa_{nc}}\right)\frac{\sin\kappa_{nc}\Delta t}{\kappa_{nc}} \\ &+ \frac{1}{\kappa_{nc}^2}\left(1 - \left((1-\cos\kappa_{nt}t_c)\frac{\kappa_{nc}\sin\kappa_{nt}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c} + \cos\kappa_{nc}t_c\right)\cos\kappa_{nc}\Delta t\right)\Bigg] \end{aligned} \quad (158)$$

Total energy is found at the end of the time step. An incremental approach needs to look at results before an after the contact time.

$$\Delta E_{int} = \frac{1}{2}D_{nt}A\int_0^{t_c}\frac{d(\delta_t(t))^2}{dt}\,dt + \frac{1}{2}D_{nc}A\int_{t_c}^{\Delta t}\frac{d(\delta_t(t))^2}{dt}\,dt \quad (159)$$

$$= \frac{1}{2}D_{nt}A\,(\delta_t(t))^2\big|_0^{t_c} + \frac{1}{2}D_{nc}A\,(\delta_t(t))^2\big|_{t_c}^{\Delta t} = \frac{1}{2}A\left(D_{nc}(\delta_t(\Delta t))^2 - D_{nt}(\delta_t(0))^2\right) \quad (160)$$

Summarizing, the interfacial energy with $\phi = \kappa_{nc}\Delta t$ and $m = m_{red}/\Delta t$ is

$$\begin{aligned} f_{n,a}^{(INT)} &= \frac{2}{\Delta t}\left[d_n\left(1 - \frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c}\frac{\sin\phi}{\phi}\right) + m\delta_n(0)\left(1 - \frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c}\cos\phi\right)\right] \\ &+ f_n\Bigg[1 - \frac{2}{\Delta t}\left(\frac{\sin\kappa_{nt}t_c}{\kappa_{nt}}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{\sin\kappa_{nc}t_c}{\kappa_{nc}}\right)\frac{\sin\phi}{\phi} \\ &- \frac{2\left(1 - \left((1-\cos\kappa_{nt}t_c)\frac{\kappa_{nc}\sin\kappa_{nc}t_c}{\kappa_{nt}\sin\kappa_{nt}t_c} + \cos\kappa_{nc}t_c\right)\cos\phi\right)}{\phi^2}\Bigg] \end{aligned} \quad (161)$$

and interface separation is

$$\begin{aligned}\delta_t(\Delta t) &= \delta_n(0)\frac{\kappa_{nt}\sin\kappa_{nc}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c}\cos\phi + \frac{d_n}{m}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c}\frac{\sin\phi}{\phi} \\ &+ \frac{f_n}{m}\left[\left(\frac{\sin\kappa_{nt}t_c}{\kappa_{nt}}\frac{\cos\kappa_{nc}t_c}{\cos\kappa_{nt}t_c} - \frac{\sin\kappa_{nc}t_c}{\kappa_{nc}}\right)\frac{\sin\phi}{\phi}\right. \\ &+ \left.\frac{\Delta t\left(1-\left((1-\cos\kappa_{nt}t_c)\,\frac{\kappa_{nc}\sin\kappa_{nt}t_c}{\kappa_{nc}\sin\kappa_{nt}t_c}+\cos\kappa_{nc}t_c\right)\cos\phi\right)}{\phi^2}\right] \end{aligned} \tag{162}$$

6.4.6 Debonded Interface

Application of above equations can have numerical issues for debonded interfaces with zero stiffness. For pure linear, the trivial solution if zero interfacial stress. But, for bilinear, it might transition into a bonded (or perfect interface). Again starting with $\delta_n(0) > 0$, the only concern is if contact occurs in the time step. The discontinuity without contact is:

$$\delta_n(t) = \delta_n(0) + \frac{d_n t}{m_{red}} + \frac{f_n t^2}{2m_{red}} \tag{163}$$

If $\delta_n(\Delta t) > 0$, interface remains debonded with zero force and energy for time step. If negative, then contact occurs at:

$$0 = 2m_{red}\delta_n(0) + 2d_n t_c + f_n t_c^2 \tag{164}$$

$$t_c = \frac{-d_n \pm \sqrt{d_n^2 - 2m_{red}\delta_n(0)f_n}}{f_n} \tag{165}$$

$$= -\frac{d_n}{f_n} \pm \text{sign}(f_n)\sqrt{\left(\frac{d_n}{f_n}\right)^2 - \frac{m_{red}\delta_n(0)}{f_n}} \tag{166}$$

The velocity at time of contact is

$$\frac{d\delta_n(t_c)}{dt} = \frac{d_n}{m_{red}} + \frac{f_n t_c}{m_{red}} < 0 \tag{167}$$

The result proves that one or the other of d_n and f_n must be negative (or both). The three possibilities are:

$$\begin{array}{lllll} d_n < 0 \text{ and } f_n < 0 & \Longrightarrow & -\frac{d_n}{f_n} < 0 (\text{sign}(f_n) = -1) & \Longrightarrow & \text{use} - \text{root} \\ d_n > 0 \text{ and } f_n < 0 & \Longrightarrow & -\frac{d_n}{f_n} < t_c & \Longrightarrow & \text{use} - \text{root} \\ d_n < 0 \text{ and } f_n > 0 & \Longrightarrow & -\frac{d_n}{f_n} > t_c & \Longrightarrow & \text{use} - \text{root} \end{array} \tag{168}$$

The contact velocity becomes:

$$m_{red}\frac{d\delta_n(t_c)}{dt} = -\sqrt{d_n^2 - m_{red}\delta_n(0)f_n} \tag{169}$$

A special case when $f_n = 0$ is

$$t_c = -\frac{m_{red}\delta_n(0)}{d_n} \qquad \text{and} \qquad \frac{d\delta_n(t_c)}{dt} = \frac{d_n}{m_{red}} \tag{170}$$

In summary, for all possibilities:

$$t_c = \begin{cases} -\frac{m_{red}\delta_n(0)}{d_n} & f_n = 0 \\ \frac{-d_n \pm \sqrt{d_n^2 - 2m_{red}\delta_n(0)f_n}}{f_n} & \text{otherwise} \end{cases} \tag{171}$$

$$d_n^{(c)} = \frac{d_n}{m_{red}} + \frac{f_n t_c}{m_{red}} = \frac{d'_n}{m_{red}} + \frac{f_n(t_c - \Delta t)}{m_{red}} \tag{172}$$

$$= -\frac{1}{m_{red}}\sqrt{d_n^2 - m_{red}\delta_n(0)f_n} \tag{173}$$

To finish the step, proceded to post-contact methods above.

6.4.7 Other?

For interface starting in compression and moving into tension, repeat the above analysis but interchange tension and compression interface properties.

For interface starting perfect ($D < 0$ in `OSParticulas`, it stays perfect as stick contact). For imperfect interface moving into contact, can apply force to reach contact and then adjust momentum to stick after contact using $d_n^{(c)}$. In other words, change input momentum change to the new value at the point of contact and also apply the force.