 CMES, vol.92, no.3, pp.271-299, 2013

Modeling Imperfect Interfaces in the Material Point Method using Multimaterial Methods

J. A. Nairn[1]

Abstract: The "multimaterial" version of the material point method (MPM) extrapolates each material to its own velocity field on a background grid. By reconciling momenta on nodes interacting with two or more materials, MPM is able to automatically handle contact without any need for special contact elements. This paper extends multimaterial MPM to automatically handle imperfect interfaces between materials as well. The approach is to evaluate displacement discontinuity on multimaterial nodes and then add internal forces and interfacial energy determined by an imperfect interface traction law. The concept is simple, but implementation required numerous corrections to make the analysis mesh independent, to work for any stiffness interfaces, and to find the interfacial normal vector. Several examples illustrated the need and demonstrated the validity of the various corrections. A composite mechanics problem found the bulk modulus of a particulate filled composite as a function interface quality. This calculation revealed a scaling effect — interfaces in nanoparticle composites must be better than interfaces in the corresponding microparticle composites for the nanocomposite mechanical properties to be as good as the conventional composite properties.

Keywords: Material point method, MPM, imperfect interfaces, multimaterial, contact, nanocomposites

1 Introduction

Modeling interfaces, which are often finite-thickness interphases, in composite materials is difficult. In numerical modeling, it would seemingly be straight-forward to discretize interphase zones and thereby explicitly model all effects. This approach has two problems. First, interphase zones may be much smaller than the bulk materials. Resolving both bulk materials and a thin interphase would require a highly refined model, which may exceed computational capacity. Second, interphase properties may be unknown and/or may vary within a transition zone from

[1] Wood Science & Engineering, Oregon State University, Corvallis, OR, 97331, USA

one material to another. Furthermore, finding interphase properties is not just a matter of extra experiments. More commonly, an interphase is not accessible as a bulk material, which means its properties would be unmeasurable. New methods for interphase modeling are needed to overcome these challenges. This need is especially important in nanocomposites because the amount of interphase per unit volume of reinforcement greatly exceeds the amount of interphase in composites with micron or larger reinforcement phases. As a consequence, interphases are expected to play a larger role, good or bad, in nanocomposite properties.

One way to model interphases is to abandon attempts for explicit modeling and instead replace 3D interphases with 2D interfaces (Hashin (1990)). The interphase effects are reduced to modeling the response of 2D interfaces due to tractions normal and tangential to the interfacial surface, which can be modeled by interface traction laws. Elimination of 3D interphases removes the resolution problem. The use of interface traction laws replaces numerous unknown and potentially unmeasurable interphase properties with a much smaller number of interface *parameters*. If interface traction laws can be determined, one can potentially model interphase effects well. This approach to interphase modeling was developed for analytical modeling of interface effects in composite materials (Hashin (1991a,b); Nairn and Liu (1997); Nairn (2004)) and for wave transmission at damage planes (Angel and Achenbach (1985)).

Numerical modeling of interfaces requires methods for implementing interface traction laws. In finite element analysis (FEA), one can use interface elements (Nairn (2007)) or specialized cohesive elements (Needleman (1987)). In the material point method (MPM) with explicit cracks (Nairn (2003)), it is similarly possible to implement interfaces (Nairn (2007)) or cohesive law tractions on crack surfaces (Nairn (2009)). In other words, the crack is modeling an interface instead of a crack. Both FEA and crack-based MPM methods require explicit introduction of interface paths prior to the start of the analysis, which complicates discretization and may limit the range of possible simulations. Another approach available in MPM is the so-called "multimaterial" approach where each material extrapolates to its own field of variables on a common grid and multimaterial algorithms adjust the response on every node containing particle information from two or more materials. Previously, this method was used to model contact physics, which, to date, has only considered Coulomb friction or stick conditions (Bardenhagen and Brackbill (1998); Bardenhagen, Guilkey, Roessig, Brackbill, Witzel, and Foster (2001)). This paper extends multimaterial MPM to modeling interfaces with traction laws. A crucial step in the algorithm is finding the normal to the interface. This paper examines the influence of normal vector calculations and offers several strategies for best accuracy. These new normal-vector methods are recommended for both

contact and interface calculations.

2 Multimaterial Contact and Interfaces

2.1 Imperfect Interface Theory

Replacing a 3D interphase with a 2D interface in composite mechanics is known as "imperfect interface theory" (Hashin (1990)). To model 3D interphase effects, the 2D interface is allowed to develop displacement discontinuities. For an isotropic interphase, it suffices to resolve interfacial traction into normal and tangential tractions (T_n and T_t) and assume they are functions of normal and tangential displacement discontinuities ($[u_n]$ and $[u_t]$) at the interface:

$$T_n = f_n([u_n]) \qquad \text{and} \qquad T_t = f_t([u_t]) \tag{1}$$

Interfaces in composite materials may develop potential energy that is needed for effective property analysis (Hashin (1992)). For an elastic interface, interfacial potential energy is

$$\phi_i = \int_{S_i} \left(\int_0^u T \cdot [u] \right) dS \tag{2}$$

where S_i is the interfacial area.

Most analytical modeling and all examples below make the simplest assumption that traction laws are linear and elastic. The tractions become:

$$T_n = D_n[u_n] \qquad \text{and} \qquad T_t = D_t[u_t] \tag{3}$$

and interfacial energy becomes

$$\phi_i = \frac{1}{2} \int_{S_i} \left(D_n[u_n]^2 + D_t[u_t]^2 \right) dS \tag{4}$$

Notice that all 3D interphase properties have been reduced to just two interface parameters, D_n and D_t. These interface stiffnesses range from zero, for a debonded interface with no tractions, to infinity, for a perfect interface with no displacement discontinuity.

2.2 Detecting Multimaterial Interfaces and Contact

In MPM, bodies are discretized into particles on a background grid (Sulsky, Chen, and Schreyer (1994)). In each time step, particle mass and momenta along with other quantities needed for algorithms, are extrapolated to nodes on the grid. In

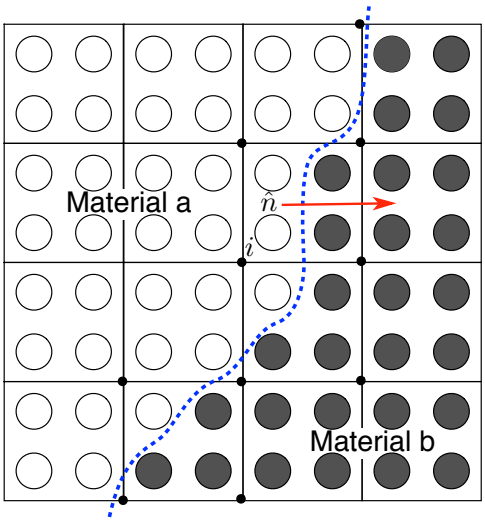

Figure 1: Material points in a 2D MPM model having two different material types on either side of a contact or interfacial surface. The normal vector $\hat{n}$ is defined as directed from material a to material b. The black dots indicate nodes (such as node i) that have multimaterial fields.

multimateral MPM, each material extrapolates to its own field. For example, material-specific mass and momenta would:

$$m_{i,j} = \sum_{p \in j} S_{ip} m_p \tag{5}$$

$$p_{i,j} = \sum_{p \in j} S_{ip} p_p \tag{6}$$

where i is a node, j is a material type, and summations are are over all particles of each type. S_{ip} are generalized shape functions for particle p at node i (Bardenhagen and Kober (2004)) and m_p and p_p are the mass and momentum of particle p. After extrapolation, nodes containing only a single material type proceed by standard MPM methods, but nodes with two or more interacting materials have to detect if the materials are in contact, implement contact laws (if needed), and, in this new algorithm, implement imperfect interfaces.

Figure 1 shows an interface between two materials, a and b, with a normal vector (that varies along the interface), $\hat{n}$, directed from a to b. Nodes indicated with a dot, such as node i, are nodes involving both materials. Two key tasks at such nodes are to determine if the two materials are in contact and to determine their separation. Previous multimaterial MPM assumed materials are in contact when

they are approaching each other, *i.e.*, when $\Delta v_i \cdot \hat{n} < 0$, where $\Delta v_i = v_{i,b} - v_{i,a}$. This condition is necessary, but not sufficient for accurate interface modeling; additional calculations are needed.

First define, $\Delta p_{i,a}$ as the momentum change applied to material a on node i for its velocity to change to the center of mass velocity:

$$\Delta p_{i,a} = m_{i,a} v_i^{(c)} - p_{i,a} \tag{7}$$

where $v_i^{(c)}$ is the center of mass velocity:

$$v_i^{(c)} = \frac{\sum_j p_{i,j}}{m_i^{(c)}} = \frac{\sum_j m_{i,j} v_{i,j}}{m_i^{(c)}} \tag{8}$$

and $m_i^{(c)} = \sum_j m_{i,j}$ is the total nodal mass. To anticipate handling nodes with more than two materials, we replace material b by a *virtual* material b that lumps all materials except a (*i.e.*, $m_{i,b} = m_i^{(c)} - m_{i,a}$ and $p_{i,b} = p_i^{(c)} - p_{i,a}$). Using Eq. (7), the velocity difference is:

$$\begin{aligned} \Delta v_i = v_{i,b} - v_{i,a} &= \frac{p_i^{(c)} - p_{i,a}}{m_i^{(c)} - m_{i,a}} - \frac{p_{i,a}}{m_{i,a}} \\ &= \frac{m_i^{(c)}}{m_{i,a}(m_i^{(c)} - m_{i,a})} \Delta p_{i,a} \end{aligned} \tag{9}$$

The first calculation is to determine if $\Delta v_i \cdot \hat{n} < 0$, which is equivalent to determining if $\Delta p_{i,a} \cdot \hat{n} < 0$. If this check fails, the two materials are assumed to be separated (*i.e.*, not in contact). When a multimaterial MPM simulation is only modeling contact, the velocity field for material a is left unchanged. When implementing imperfect interfaces, however, tractions may exist for both contact and separated conditions, so this algorithm continues.

The next task is to calculate the separation between the two materials, which is done by extrapolating particle position to the grid using standard MPM methods:

$$m_{i,j} x_{i,j} = \sum_{p \in j} S_{ip} m_p x_p \tag{10}$$

Again lumping all materials except a into virtual b, the separation vector is:

$$\delta_i = x_{i,b} - x_{i,a} = \frac{m_i^{(c)}}{m_i^{(c)} - m_{i,a}} \left(x_i^{(c)} - x_{i,a} \right) \tag{11}$$

where $x_i^{(c)} = \sum_j m_{i,j} x_{i,j} / m_i^{(c)}$ is the center of mass position. Contact is now determined by the component of the separation vector normal to the surface or

$$\delta_n = \delta_i \cdot \hat{n} \tag{12}$$

If $\delta_n < 0$, the materials are in contact, otherwise they are separated. But, inherent imprecision of surfaces in MPM (or any particle method) causes a problem. Consider the two surfaces in Fig. 1. When calculations start, their nodal positions extrapolated to common nodes will show a positive separation. Calculations with MPM shape functions show that the calculated separation based on positions of two materials precisely in contact is $\delta_n \approx 0.8\Delta x$ rather than zero (Lemiale, Hurmane, and Nairn (2010)). Two approaches can resolve this issue:

1. *Extrapolate displacements* instead of position using

$$m_{i,j} x_{i,j} = \sum_{p \in j} N_i(x_p) m_p \left(x_p - x_p^{(0)} \right) \tag{13}$$

where $x_p^{(0)}$ is the initial position for particle p. Now the separation for two materials in contact when calculations start is zero and subsequent $\delta_n < 0$ will indicate contact.

2. *Define a contact-position offset*, such as $\delta_{con} = 0.8\Delta x$, and correct the normal separation using:

$$\delta_n' = \delta_i \cdot \hat{n} - \delta_{con} \tag{14}$$

(see below for revised correct in irregular meshes).

Both options should be available in multimaterial calculations. The first (or displacements) method requires material contact situations to exist in the initial state and requires relative displacements between materials to be modest. This situation is common in interface modeling. The second (or offset) method must be used when these requirements are not met, which is common in many contact problems. All calculations in this paper involved interfaces and small displacements and thus used the displacements method.

If both $\Delta p_{i,a} \cdot \hat{n} < 0$ and $\delta_n < 0$, the two materials are determined to be in contact; otherwise they are separated. Some codes implement an additional screening based on total nodal volume at a multimaterial node (Guilkey (2010)). If that volume is less than a chosen cutoff fraction of the element volume, then both contact and interface calculations are skipped. These nodes appear at interfaces near free edges. Although they might actually by in contact or have an interface, in can be better to ignore them (see results and discussion section for examples).

2.3 Imperfect Interface and/or Contact Forces

The next task is to implement imperfect interface or contact laws. In multimaterial simulations, each pair of materials can be described by contact (*e.g.*, friction) or as an imperfect interface, but not both. If a pair is interacting only by contact and is separated, no changes are made at that node; otherwise calculations proceed as follows:

1. Find the tangent vector in the direction of the surface opening displacement. Given the normal separation, δ_n, and normal vector, $\hat{n}$, the tangent vector is

$$\delta_t \hat{t} = \delta_i - \delta_n \hat{n} \tag{15}$$

   where $\delta_t = \delta_i \cdot \hat{t}$ is displacement difference in the tangential direction.

2. If two materials are interacting by contact, change the momentum for material a. If they interact by stick, change material a's momentum by $\Delta p_{i,a}$. If they interact by frictional contact and $\Delta p_{i,a} \cdot \hat{t} > -\mu \delta_n$, where μ is the coefficient of friction, the material pair is sliding and the momentum change should be

$$\Delta p'_{i,a} = \delta_n (\hat{n} - \mu \hat{t}) \tag{16}$$

   otherwise the material pair is stuck and material a's momentum changes by $\Delta p_{i,a}$. This step is identical to previously-derived MPM contact (Bardenhagen, Guilkey, Roessig, Brackbill, Witzel, and Foster (2001))

3. If the two materials are connected by an interface, this algorithm applies an internal force to the node rather then changing the nodal momentum. The logic is that an interface is a material property leading to interfacial tractions and should behave similarly to material constitutive laws that lead to stresses and internal forces. The internal force for material j on node i, which is added to other MPM internal forces, is:

$$f_{i,j}^{(int)} = (D_n \delta_n \hat{n} + D_t \delta_t \hat{t}) A_i \tag{17}$$

   and the interfacial energy at node i is

$$U_i^{(int)} = (D_n \delta_n^2 + D_t \delta_t^2) A_i \tag{18}$$

   where A_i is interfacial contact area. These terms assume a linear interface law; any other interfacial traction law could be substituted at this step.

Calculation of A_i needs to account for the number of interacting nodes along the interface, which is influenced by location and orientation of the interface as illustrated in Fig. 2 for 2D calculations. The needed correction is found by extrapolating particle volume (or particle area in 2D):

$$\Omega_{i,j} = \begin{cases} \sum_{p\in j} S_{ip} V_p & \text{3D} \\ \sum_{p\in j} S_{ip} A_p & \text{2D} \end{cases} \tag{19}$$

where V_p or A_p are volume or area of the deformed particle domain. Consider a 2D interface close to a grid line when particles are close to their starting positions (1/4 points of cells with S_{ip} as indicated; see Fig. 2A). The nodal domains at nodes i and j would be $\Omega_{i,a} = \Omega_{i,b} = A_{cell}/2$, $\Omega_{j,a} = 0$, and $\Omega_{j,b} = A_{cell}$. Here $A_{cell} = \Delta x \Delta y$, where Δx and Δy are grid spacings in the x and y directions. The total nodal domains are $\Omega_i = \Omega_{i,a} + \Omega_{i,b} = \Omega_j = A_{cell}$. The contacting area is obviously $A_i = t\Delta y$, where t is thickness, or $A_i = t\Omega_i/\Delta x$. Furthermore, node i is a multimaterial node, but node j is not. In contrast, when the interface is at element midplanes, both nodes i and j are multimaterial nodes with nodal domains of $\Omega_{i,a} = \Omega_{j,b} = 7A_{cell}/8$, $\Omega_{i,b} = \Omega_{j,a} = A_{cell}/8$, and $\Omega_i = \Omega_j = A_{cell}$ (see Fig. 2B). If contacting areas are unchanged, *i.e.*, $A_i = t\Omega_i/\Delta x$ and $A_j = t\Omega_j/\Delta x$, the total interfacial force would be double the force in Fig. 2A because it involves twice as many nodes. For these two cases, however, total interfacial force should be the same. One solution is to scale the contact area by the distance from the interface to the node as estimated by the mismatch in nodal domains for the two materials. A reasonable function that gets the correct scaling for both Fig. 2A and B, and correctly drops to zero as one material vanishes is:

$$A_i = \frac{t\sqrt{2\Omega_i \min(\Omega_{i,a}, \Omega_{i,b})}}{\Delta x} \tag{20}$$

This scaling results in $A_i \approx tA_{cell}/\Delta x$ and $A_j \approx 0$ for Fig. 2A and $A_i \approx A_j \approx tA_{cell}/(2\Delta x)$ for Fig. 2B; the total interfacial force becomes independent of interface location. In 3D problems, Ω_i is volume and therefore t is dropped in Eq. (20). In axisymmetric MPM (Nairn and Guilkey (2013)), the t is replaced by r_i, which is the radial position of the node, to provide interface forces per radian.

Figure 2C shows a tilted interface, which sometimes needs an additional correction. Imagine an ellipsoid centered on a node and consider the plane through that node with normal vector $\hat{n}$. Assume that the volume associated with the interfacial contact matches the cell volume such that

$$V_{cell} = \Delta x \Delta y \Delta z = h_\perp A_c \tag{21}$$

where A_c is the contacting area and $h_\perp$ is an effective thickness of the contacting volume perpendicular to the contacting surface. Once $h_\perp$ is known, A_i can be adjusted by replacing Δx with $h_\perp$. Three $h_\perp$ methods were tried:

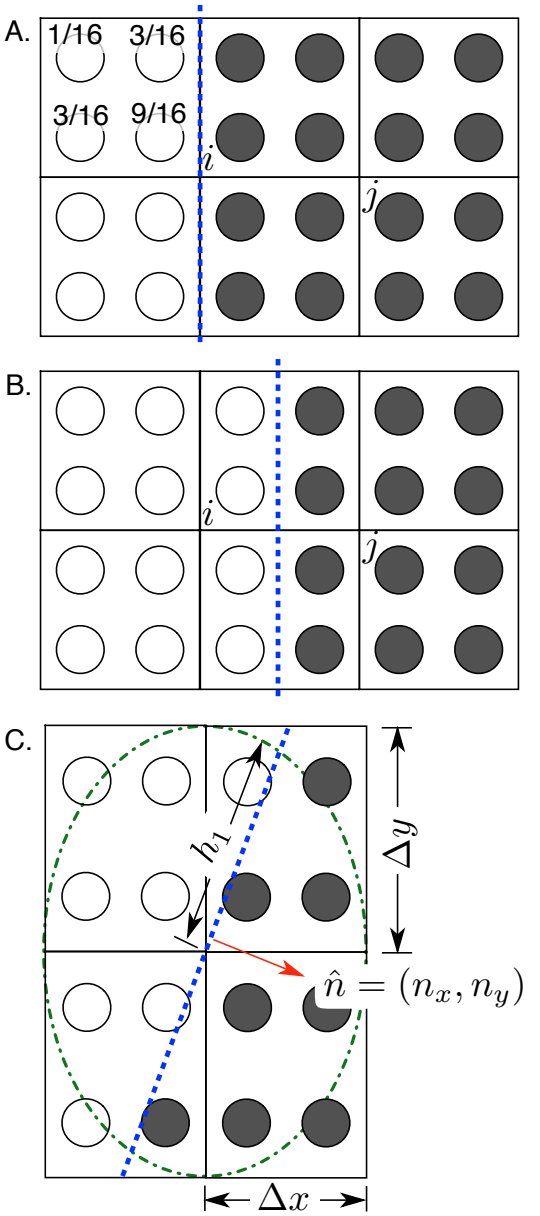

Figure 2: Various orientations of an interface including at a grid line (A), an elements mid plane (B), and titled with respect to global *x*-*y*-*z* axes. The numbers on particles in A are values of shape functions when particles are near their initial positions. C shows an ellipse around the center node circumscribed by neighboring cells.

1. Let h_1 and h_2 be distances from the origin to the ellipsoid along the tangential vector, $\hat{t}$, and along a second tangent defined by $\hat{n} \times \hat{t}$, then $h_\perp = \Delta x \Delta y \Delta z / (h_1 h_2)$.

2. Let $h_\perp$ be the distance along the normal vector to the ellipsoid.

3. Let $h_\perp = \Delta x \Delta y \Delta z / A_c$ where A_c is the cross sectional area that the plane with normal $\hat{n}$ makes with the entire element.

Calculations (see results and discussion) showed that method 1 best accounts for tilted planes. That method evaluates to

$$h_\perp = \begin{cases} \sqrt{n_x^2 \Delta x^2 + n_y^2 \Delta y^2} & \text{2D} \\ \Delta x \Delta y \Delta z \sqrt{\left(a_1^2 + a_2^2 + a_3^2\right)\left(b_1^2 + b_2^2 + b_3^2\right)} & \text{3D} \end{cases} \tag{22}$$

where n_x and n_y are normal vector components, $a_i = \hat{t}_i^2 / \Delta x_i^2$, $b_i = (\hat{n} \times \hat{t})_i^2 / \Delta x_i^2$, and $i = x$, y, and z. For a regular mesh with $\Delta x = \Delta y = \Delta z$, $h_\perp$ is always equal to Δx; *i.e.*, this extra correction is only needed for meshes with unequal cell lengths. The 2D result applies to axisymmetric MPM as well. Note that $h_\perp$ is also useful for correcting extrapolated positions by changing to $\delta_{con} = 0.8 h_\perp$.

Finally, as $D_n \to \infty$ and $D_t \to \infty$, the interface becomes perfect with zero displacement discontinuity, but in explicit integrations, large D_n or D_t will eventually be too stiff to converge. A practical solution is to monitor internal forces. If they exceed the force required to move the particle to the center of mass, that node is processed differently. Integrating nodal velocity over time step k, nodal position for the center of mass updates by midpoint rule to:

$$u_{i,c}^{(k+1)} = u_{i,c}^{(k)} + \frac{\Delta t}{m_{i,c}^{(k)}} \left(p_{i,a}^{(k)} + p_{i,b}^{(k)} + \frac{\Delta t}{2} (f_{i,a}^{(k)} + f_{i,b}^{(k)}) \right) \tag{23}$$

where $f_{i,j}^{(k)}$ are MPM forces for each material besides interface forces. The center of mass evolves independently of interface tractions. Materials a and b at the interface, however, evolve separately by:

$$u_{i,a}^{(k+1)} = u_{i,a}^{(k)} + \frac{\Delta t}{m_{i,a}^{(k)}} \left(p_{i,a}^{(k)} + \frac{\Delta t}{2} (f_{i,a}^{(k)} + f_{i,a}^{(int,k)}) \right) \tag{24}$$

$$u_{i,b}^{(k+1)} = u_{i,b}^{(k)} + \frac{\Delta t}{m_{i,b}^{(k)}} \left(p_{i,b}^{(k)} + \frac{\Delta t}{2} (f_{i,b}^{(k)} - f_{i,a}^{(int,k)}) \right) \tag{25}$$

where force conservation was used to find interface force on material b as opposite of the force on material a. The interface force that would cause materials a and b to evolve to the center of mass position (*i.e.*, $u_a^{(k+1)} = u_b^{(k+1)} = u_c^{(k+1)}$) is:

$$f_{i,a}^{(int,k)} = \frac{2m_{red}}{(\Delta t)^2}\left(u_b^{(k)} - u_a^{(k)}\right) + \frac{2\Delta p_{i,a}}{\Delta t} + \Delta f_i. \tag{26}$$

where

$$m_{red} = \frac{m_{i,a}^{(k)} m_{i,b}^{(k)}}{m_{i,a}^{(k)} + m_{i,b}^{(k)}} \text{ and } \Delta f_i = \frac{m_{i,a}^{(k)} f_{i,b}^{(k)} - m_{i,b}^{(k)} f_{i,a}^{(k)}}{m_{i,a}^{(k)} + m_{i,b}^{(k)}} \tag{27}$$

If calculated interfacial force in the normal or tangential direction exceeds the normal or tangential components of the force in Eq. (26), that node direction should be converted to a node with stick contact and having zero interfacial energy (*i*.e., a perfect interface); otherwise the interfacial force is added to internal forces. It is convenient in MPM codes to implement interface calculations after extrapolating material velocity fields, but before other forces are known (*i.e.*, along with contact calculations). In this location, Δf_i will be unknown, but because it is second order (when Δt is small), it can be ignored.

A code refinement for efficiency is to "flag" either D_n or D_t as being infinite or perfect. When these flags are set, forces are always limited and therefore that direction reverts to stick contact with zero interfacial energy. Because the forces are always limited, the above calculations for maximum force can be skipped.

Prior work on imperfect interfaces in MPM implemented them on explicit cracks (Nairn (2007)). That paper did not address the need to correct contact area by relative volumes, to account for tilt in irregular meshes, or to limit forces for stiff interfaces. Those corrections, which are derived above, should be used for interfaces on cracks as well. Because the examples in Nairn (2007) always used cracks along grid lines in regular meshes with either sufficiently low stiffness or "flagged" perfect interfaces, those examples did not need these new corrections.

2.4 Rigid Materials

A useful feature in MPM is to allow rigid material particles that move at a prescribed velocity. Because they are rigid, their velocity cannot be changed by forces, but they can apply forces to nonrigid particles through contact or interfacial interactions. For interactions between non-rigid material a and rigid material b, the above

equations change to:

$$v_i^{(c)} = v_{i,b} \tag{28}$$

$$\Delta p_{i,a} = -m_{i,a}\left(v_{i,a} - v_{i,b}\right) \tag{29}$$

$$\Delta v = \frac{\Delta p_{i,a}}{m_{i,a}} \tag{30}$$

$$\delta_i = x_{i,b} - x_{i,a} \tag{31}$$

The remaining equations are the same except momentum changes and internal forces are applied only to material a.

2.5 Finding the Normal Vector

Experience in contact and interface simulations supports this assertion — MPM contact and interface calculations are very accurate provided the normal vector is accurate (Lemiale, Hurmane, and Nairn (2010)). Original MPM contact found the normal from the mass gradient for each material (Bardenhagen, Guilkey, Roessig, Brackbill, Witzel, and Foster (2001)). Because methods described below compare gradients between materials, which might have different densities, it is preferable to use a domain gradient:

$$\|n_{i,a}\| n_{i,a} = -\nabla \Omega_{i,a} \tag{32}$$

where $n_{i,a}$ is an unnormalized, outward directed normal for material a at node i. In MPM, a good way to find domain gradients is to extrapolate using shape function gradients:

$$g_{i,j} = \begin{cases} \sum_{p\in j} G_{ip} V_p & \text{3D} \\ \sum_{p\in j} G_{ip} A_p & \text{2D} \end{cases} \tag{33}$$

where G_{ip} are generalized shape function gradients (Bardenhagen and Kober (2004)). In axisymmetric MPM, $g_{i,j}$ would treat nodes at $r = 0$ as being on the edge with a non-zero r component. But in reality, such nodes are at the core of the object and therefore by symmetry, the r component must be set to zero (Nairn and Guilkey (2013)). Curiously, $g_{i,j} = -\nabla \Omega_a(x_i)$ leading to $\|\hat{n}_{i,a}\| \hat{n}_{i,a} = g_{i,j}$ (Bardenhagen (2012)).

Unfortunately, MPM calculations for material a based solely on $g_{i,a}$ frequently give poor results. The following replacement options are proposed:

Maximum Volume Gradient (MVG): find the normal from material a or b, whichever has the domain gradient with the largest magnitude. The logic is that higher gradients are likely closer to an interface and likely more accurate than when the magnitude is smaller.

Average Volume Gradient (AVG): find the normal from a domain-weighted average of gradients from the two materials — $(\Omega_{i,a} g_{i,a} - \Omega_{i,b} g_{i,b})/(2\Omega_i)$

Specific Normal (SN): if the normal can be predetermined, use that specified result rather than domain gradients.

Rigid Material Volume Gradient (RMVG): find the normal from the rigid material's domain gradient. In problems with rigid materials, especially fixed rigid particles, the gradient from that material is often more reliable than the gradient from non-rigid particles.

Results below compare calculations using methods MVG, AVG, and SN. Unfortunately, the best method depends on the problem. MPM code therefore needs to allow all these options, and potentially may need more options for new problems.

2.6 Two or More Materials

Contact and interfaces in MPM work best when only two materials interact at multimaterial nodes. This situation is most likely to get accurate normals. For efficiency, contact or interface calculations only need to be done once. Once completed for material a, the negative momentum and force changes can be applied to material b's field variables without any additional calculations. This approach conserves momentum, balances forces, and is more efficient. When calculating interfacial energy (*e.g.*, Eq. (4)), that energy should be added only once for each pair.

When more than two materials are present, the following algorithm allows calculations to proceed:

1. Iterate through each non-rigid material and find momentum changes and internal forces for each one using the method above that lumps all remaining materials as a virtual material b. Apply changes to each material separately, but increment interfacial energy only once for each multimaterial node.

2. If a simulation has different properties between different material pairs (*e.g.*, different coefficients of friction or interfacial properties), use the properties for interaction between material a and the remaining material with the most volume.

3. If one of the materials is a rigid material, implement rigid contact between each non-rigid material and the rigid material. In this case, include interfacial energy for each interaction.

Handling nodes with more than two materials will always be approximate. In some situations, the approximations are due to resolution, *i.e.*, the model is not resolving

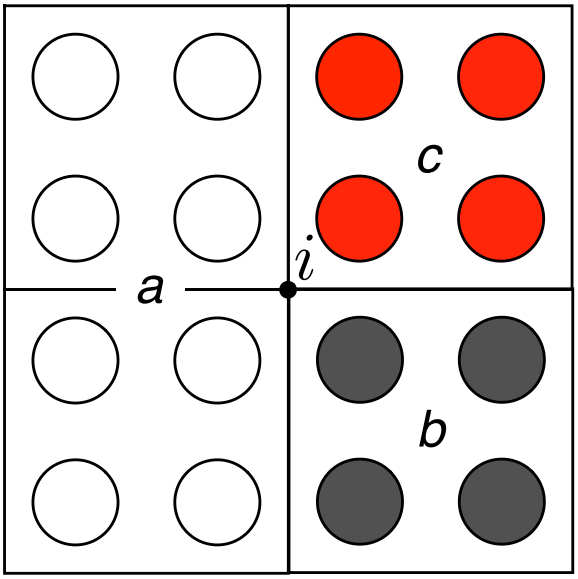

Figure 3: Three different materials, a, b, and c, interacting at a single multimaterial node i.

the materials. But no resolution increase can avoid multiple material nodes such as node i in Fig. 3. At such nodes, the above algorithm adjusts momenta and applies internal forces, but it does not conserve momentum, apply balanced forces, or find accurate interfacial energy. The separate material fields simply have insufficient information for highest accuracy calculations. In general increased resolution should help. As resolution increases, the number of nodes along interfaces increases, but the number of nodes similar to Fig. 3 remains constant. The goal should be to have the number of multimaterial nodes with more than two materials be much less than the number with two materials.

3 Results and Discussion

Figure 4A shows two identical layers connected by an imperfect interface. The isotropic elastic layers had $E = 2400$ MPa, $\nu = 0.33$, and were $4 \times 4 \times 20$ mm. One layer was loaded to constant 0.5 MPa stress on top while the entire bottom was fixed. The interface was assumed to be perfect in the normal direction ($D_n = \infty$) while D_t was varied. The normal vector used the SN method with $\hat{n} = (1,0)$ (normal vector effects are discussed later). In these dynamic MPM calculations, the load was ramped up linearly over 0.02 ms and then held constant; grid damping was used to converge to static results after about 0.4 ms.

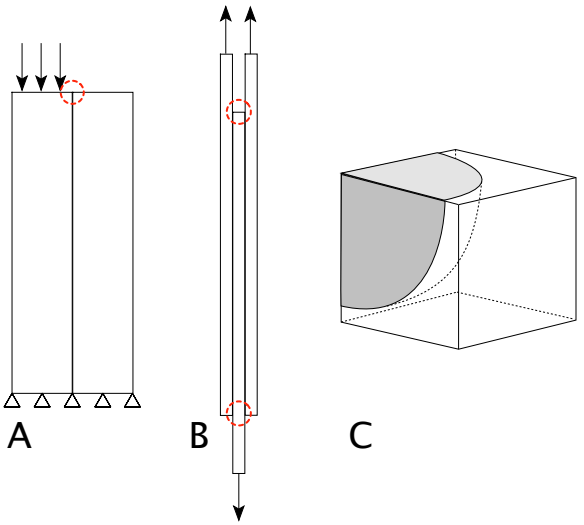

Figure 4: Three sample specimens. A. Two layer specimens loaded on top of one layer. B. A double lap shear (DLS) specimen with loading on two ends. C. 1/8 of a 3D model for a spherical particle embedded in a matrix in a cubic array. The dotted circles in A and B show edge regions that cause problems when calculating interface normals.

The results for total interfacial energy per unit volume (by Eq. (4)) are given in Fig. 5. The FEA curve is a finite element analysis result using imperfect interface elements (Nairn (2007)) with a regular grid of 0.25 mm elements. The open symbols are 2D MPM results for background cells of 0.5 mm (squares), 0.25 mm (triangles), and 0.1 mm (circles). The MPM results changed much less between 0.25 mm and 0.1 mm than between 0.5 mm and 0.25 mm suggesting they are closed to converged for 0.25 mm. The converged results agree reasonably well with FEA calculations. The differences were probably caused by differences in the way FEA and MPM apply boundary conditions on the top of the specimen. The solid symbols are 3D calculations to verify 3D interface methods. In 3D, a single tangential force is applied in the direction of tangential motion. The dotted curve is a prior MPM method for MPM imperfect interfaces by using traction laws on an explicit crack surface along the interface (Nairn (2007)). The crack calculation used 0.25 mm grid and was essentially identical to the 0.25 mm grid results here based on multimaterial MPM methods.

Total interfacial energy density in two layer specimens was a small fraction of the total energy — $< 2\%$ at most — but the interface had a large affect on axial

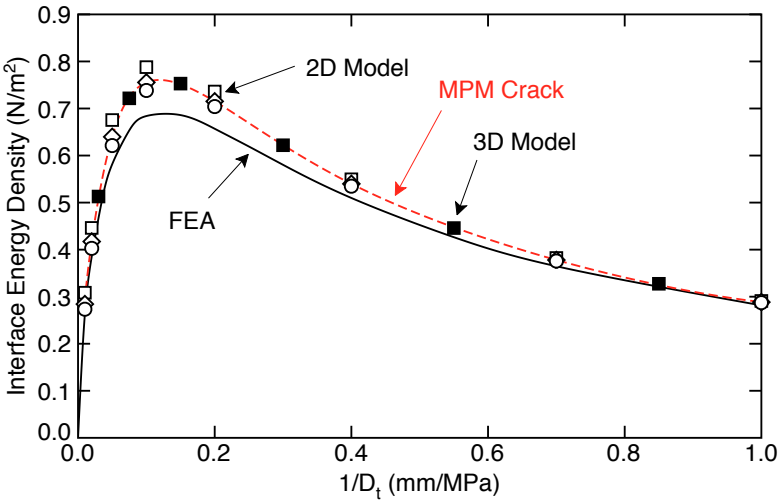

Figure 5: Interfacial energy density in the two layer example as a function of D_t with $D_n = \infty$. The open symbols are MPM results for three different resolutions (0.5, 0.25, and 0.1 mm cells). The solid symbols are 3D MPM results. The dashed curve is MPM with an imperfect interface on an explicit crack. The solid line is finite element analysis.

displacements. Figure 6 shows that the y direction displacements from FEA and MPM for $D_t = 5$ MPa/mm agree well. The interface is sharper by FEA due to nature of element-based solution compared to a particle-based solution in MPM.

A second example modeled a double lap shear (DLS) specimen (see Fig. 4B). The interface plays a larger role in this specimen, which prompted recent work to suggest its use for measuring D_t (Le and Nairn (2012)). By numerical modeling, total energy U can be evaluated as a sum of the strain energy and the interfacial energy. This energy is related to specimen stiffness by $k = P^2/(2U)$, where P is total applied load. By measuring k and comparing to modeling results for U where D_t is the only unknown, one can, in principle, measure D_t. The key specimen requirement is that k changes by measurable amounts in the range of actual interfacial properties. This process worked reasonably well for wood strands with an interface of wood glue (Le and Nairn (2012)).

The calculations here modeled a DLS specimen with wood layers, which were transversely isotropic with the axial wood grain in the y direction. The properties were $E_A = 10000$ MPa, $E_T = 1000$ MPa, $G_A = 500$ MPa, $\nu_A = 0.2$, and $\nu_T = 0.3$. Each layer was 1 mm thick, the bond line was 25 mm long, and free ends on top

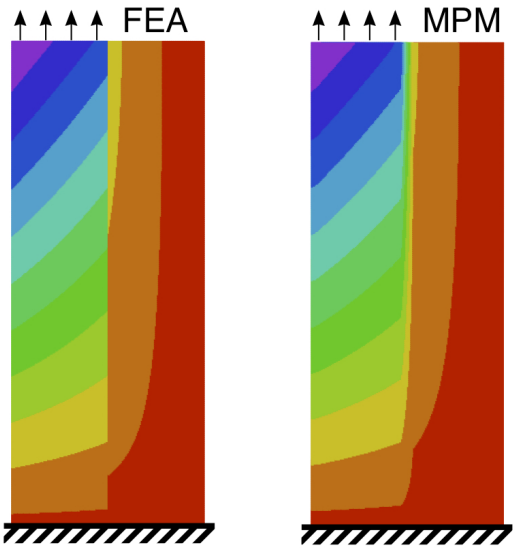

Figure 6: Axial displacements in two layer specimens by MPM or FEA when $D_t =$ 5 MPa/mm. The shading from red to purple shows 11 contours from -0.0005 mm to 0.005 mm.

and bottom were 5 mm long. The two tops were loaded to 1 MPa while the bottom was loaded to 2 MPa (to have balanced total force). The interface was assumed to be perfect in the normal direction ($D_n = \infty$) while D_t was varied. The normal vector used the SN method with $\hat{n} = (1,0)$. The load was ramped up linearly over 0.04 ms and then held constant; grid damping was used to converge to static results after about 0.22 ms.

Figure 7 plots DLS stiffness for wood specimens. The open circles are MPM results using the SN normal method. The filled squares are FEA results. MPM and FEA results are nearly identical. The solid line is an analytical solution based on shear-lag analysis (Nairn (2007)). The theory agrees well with numerical results, which both validates the theory and further validates the numerical implementation of imperfect interfaces. The open diamonds are MPM results using explicit cracks rather than multimaterial interfaces. These results show again that multimaterial mode MPM can accurately model imperfect interfaces. Notice also that over the range analyzed, DLS stiffness changes by over a factor of five. In other words, the interface plays a major role in DLS specimen stiffness. The total interfacial energy increased to 91% of total energy as D_t decreased to 1 MPa/mm. If DLS stiffness can be measured accurately, inversion of the curve in Fig. 7 provides a measure of

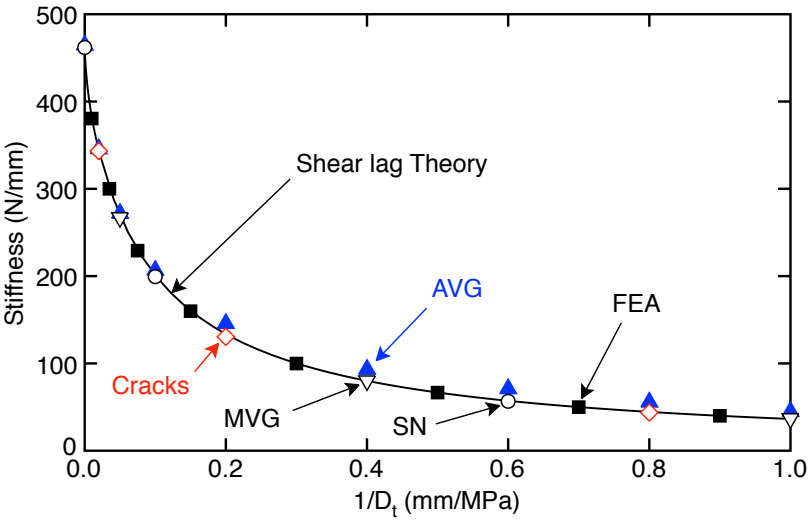

Figure 7: Calculated stiffness for the DLS specimen. The symbols are numerical results using three different normal methods (SN, MVG, and AVG), MPM with explicit cracks (Cracks), and finite element analysis (FEA). The solid line is an analytical shear lag solution.

D_t (Le and Nairn (2012)).

Figure 8 show results for shifted interfaces in two-layer specimens to verify the correction term in Eq. (20). The square symbols show results for the interface along the grid line (see Fig. 2A); the open squares are corrected using Eq. (20) while solid squares use the uncorrected contact area of $t\Omega_i/\Delta x$. Although corrected and uncorrected A_i are the same for nodes close to the interface, as the solution evolves a few particles migrate resulting in small, non-zero shape functions on nodes one element away from the interface (*e.g.*, node j in ig. 2A). The corrected area appropriately results in very small force on such nodes because the minimum volume is low. In contrast, the uncorrected area gives force equal to nodes on the grid line causing interfacial energy to be too high.

The circle symbols in Fig. 8 show results for an interface along cell midplanes (see Fig. 2B); the open circles are corrected using Eq. (20) while solid circles use the uncorrected contact area of $t\Omega_i/\Delta x$. When Eq. (20) is not used the contact area is twice as large resulting in interface results that effectively have D_t twice as large. This doubling of effective D_t causes the x axis in a $1/D_t$ plot to shift right by a factor of 2. Clearly a contact area correction is needed regardless of the location of the interface; furthermore the correction in Eq. (20) is accurate because the results

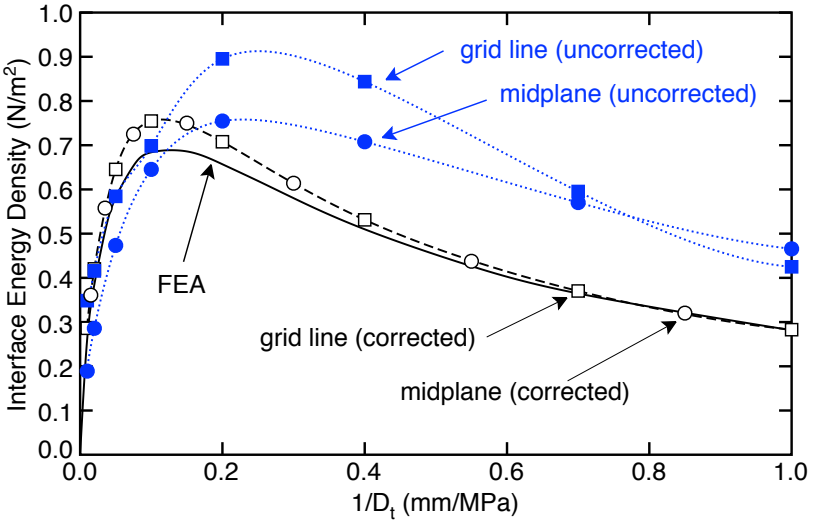

Figure 8: Interfacial energy density in two layer specimens with the interface located either at a grid line or at element midplanes. The open symbols correct the contact area using Eq. (20) while the solid symbols do not. The FEA curve is corresponding result by finite elements analysis for reference.

are independent of interface location and always close to FEA results.

Imperfect interface modeling needs to work for any interfacial orientation, and not just for the shifting discussed above. To confirm this ability, the DLS example was rerun by rotating the specimen with respect to the x and y axes. The solid line in Fig. 9 is for a regular mesh ($\Delta x = \Delta y$) and the results are independent of interface angle. For an irregular mesh, however, the results need an additional correction to replace Δx in Eq. (20) with $h_\perp$ in Eq. (22). The dotted lines in Fig. 9 show DLS results as a function of angle for a mesh with $\Delta y = 2\Delta x$ using "No Correction" (*i.e.*, $h_\perp = \Delta x$) and the three $h_\perp$ methods described above. The "No Correction" result is poor. Method 1 (see Eq. (22)) worked best and is therefore the recommended method. Even the best method, however, is not as good as the regular mesh result, which may be because the irregular mesh was at a lower resolution in the y direction for this example.

Figure 10 shows the importance of the method used to find the contact normal for two-layer specimens (Fig. 4A) using 0.25×0.25 mm grids. First, look at the "grid" curves, which are when the interface is near a grid line. For this geometry, a specified normal (SN using $\hat{n} = (1,0,0)$) is the control and is the same result as in Fig. 5. If the normals are found by the MVG (grid) method instead, the results are

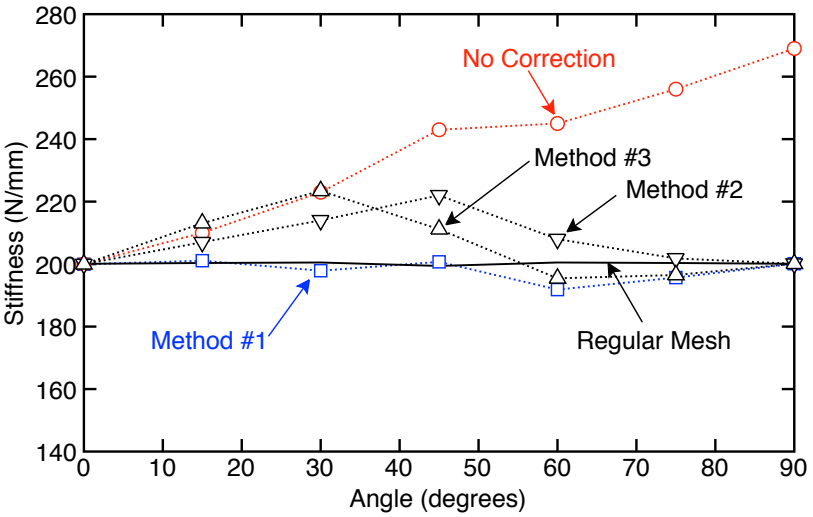

Figure 9: Calculated stiffness for the DLS specimen for tilted specimen as a function of rotation angle. The solid line is for a regular grid. The dotted lines are for an irregular grid with $\Delta y = 2\Delta x$ and various methods for correcting for interface tilt.

poor. The inaccuracies are caused by the single node on the top of the specimen (see circled region in Fig. 4A). At this position the normals for the two materials are at 45° to the interface rather than at $(1,0,0)$. This error can be fixed by averaging the normals because the two 45° normals average to the correct $(1,0,0)$ normal. This fix is confirmed by the AVG (grid) results, which agree with the control.

Next, look at the "mid" results, which are for the interface at element midplanes. Again the SN (mid) results gave the correct result (as already discussed for Fig. 8). But now, neither MVG (mid) nor AVG (mid) gave acceptable results. The problem was again the contacting nodes on the top edge, but now the errors were not fixed by averaging. This example illustrates a potential problem in all MPM contact or interface methods — the normals are often wrong when the contact surface is at a free edge. In many calculations, errors caused by such nodes might be small, but in this two layer specimen they happen to be large. A solution for this specimen (and maybe many MPM calculations) is to ignore those problem nodes. In other words, it can be better to ignore edge nodes then to include them with potentially erroneous normals (Guilkey (2010)). A method to ignore edge nodes is to screen them out based on total nodal domain, Ω_i. For the edge nodes in two layer specimens $\Omega_i \sim A_{cell}/2$, while for all other nodes $\Omega_i \sim A_{cell}$. Edge nodes can therefore be screened out by ignoring interfacial forces and energy for multimaterial nodes with $\Omega_i <$

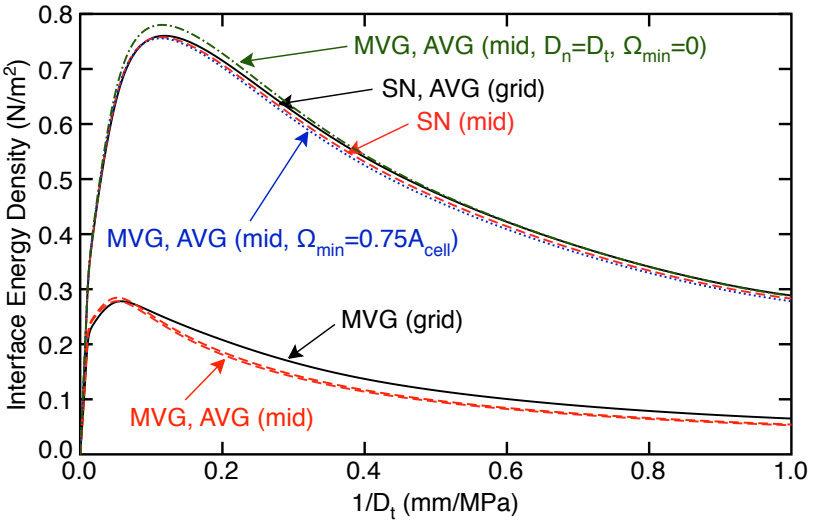

Figure 10: Interfacial normal energy density in two-layer specimens with the interface at a "grid" line or at element "mid" planes. The discrete numerical results were smoothed by a spline interpolant to make it easier to visualize differences between similar results.

$0.75A_{cell}$. The MVG and AVG (mid, $\Omega_{min} = 0.75A_{cell}$) results in Fig. 10 confirms this approach agrees well with the control.

This two-layer specimen may be a pathological specimen that exaggerates the importance of normals (an hence a good example for developing interface methods). The results are very sensitive to the normal because total interfacial energy is very small. Any inaccuracies can lead to large changes in interfacial energy. The two-layer problem was further exacerbated by its perfect interface in the normal direction. Because a perfect interface corresponds to a large D_n, any errors in $\hat{n}$ get amplified by the high interfacial stiffness. Figure 10 shows calculations with $D_n = D_t$ (*i.e.*, when both directions are imperfect) by MVG and AVG (mid) and without volume screening. These results agree with controls, despite inclusion of inaccurate edge nodes. Because D_n is smaller, those inaccuracies have much less affect on the results. The reason they are close to controls that had $D_n = \infty$ are that the normal direction plays little role in this shear-loaded specimen. The slightly higher peak energy, however, is likely the net effect of an imperfect, rather then perfect, normal interface.

Figure 7 shows calculation of DLS stiffness for SN (the control with $\hat{n} = (1,0)$), MVG, and AVG. Now SN and MVG agree well, while AVG has small errors. The

problems for this specimen are nodes where one layer stops while the other one continues (see circled regions in Fig. 4B). At these locations, the continuous layer has the correct normal and is the material with the maximum volume gradient. Hence, the MVG method worked well. In contrast, the AVG method averages a correct normal with an inaccurate normal at 45° and get less accurate results The effect is much smaller in this specimen because the interfacial energy for other nodes is large. Nevertheless, for maximum accuracy, it is important to have accurate normals.

The results in Fig. 7 are for the interface at grid lines. The normals effect is larger when the interface is at cell midplanes, as shown in Fig. 11. Now both MVG and AVG gave poor results. These results can be fixed by the same two methods used for the two layer specimen — screen out edge nodes by ignoring nodes with $\Omega_i < 0.85A_{cell}$ (because those nodes are expected to have $\Omega_i \sim 0.75A_{cell}$) or set $D_n = D_t$ (*i.e.*, allow the interface to be imperfect in the normal direction rather than perfect). Using either method along with any normal method (SN, MVG, or AVG) all gave identical results, which agreed with shear lag theory (see "All Corrected Methods (mid)" symbols).

In summary, the most accurate results are always by SN. When SN is not possible, which happens whenever interfaces might reorient during a calculation, both MVG and AVG can work, but require care. If neither D_n nor D_t are too stiff, both MVG and AVG usually work. As D_n and D_t get higher, however, problems frequently arise at edge nodes. Therefore, in problems with one or more stiff direction and with edge nodes, it is probably essential to ignore edge nodes by ignoring interfacial force and energy at nodes with Ω_i less then a carefully selected critical value. These recommendations apply to MPM contact calculations as well as interface calculations. In fact, MPM contact can be characterizes as $D_n = 0$ in tension but $D_n = \infty$ during contact. In other words, contact calculations always have one stiff direction. Those calculations should use MVG or AVG along with volume screening to ignore edge nodes.

A final example looked at a 3D composite mechanics problem and evaluated the algorithm to limit forces when D_n or D_t get high. The problem, shown in Fig. 4B, is 1/8 of a spherical particle embedded in a cubic array. The particle had radius of 700 μm in a cube of side 1000 μm to model a particulate filled composite with particle volume fraction $V_p = 17.96\%$. The particles were glass ($E = 70,000$ MPa, $\nu = 0.25$) embedded in a polymer matrix ($E = 2400$ MPa, $\nu = 0.33$). The interface was imperfect in both directions with $D_n = D_t$. The normal was found by the SN method; if the origin is at the center of the particle, the normal at node i is in the direction (x_i, y_i, z_i). The cube had zero displacement on the symmetry planes and was loaded to applied strain of $\varepsilon_0 = 0.5\%$ strain in all three directions on the cube's

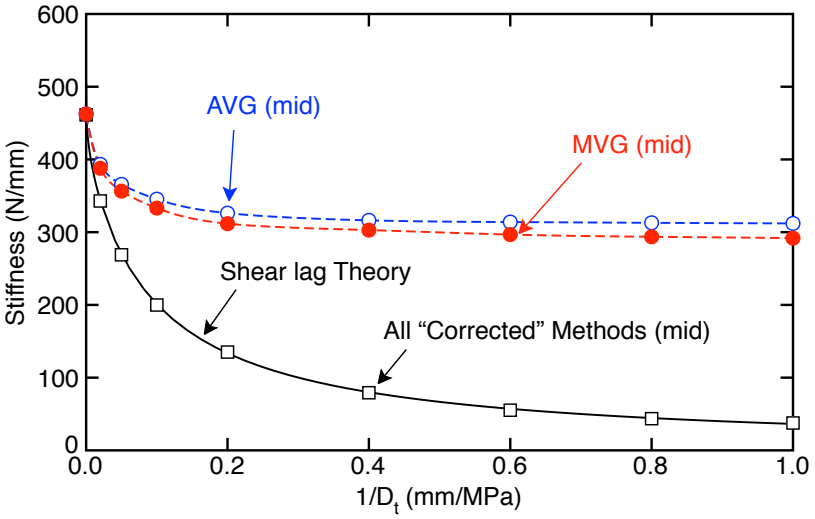

Figure 11: The stiffness of a DLS specimen with the interface at element "mid" planes. The symbols are MPM results. The solid line is an analytical shear lag solution.

outer surfaces. The surfaces were loaded at a rate of about 0.1% of the polymer wave speed and then held fixed until grid damping settled on a static result.

By composite variational mechanics, the bulk modulus is related to total energy, U, by

$$U = \frac{V}{2} K \left(\frac{\Delta V}{V_0} \right)^2 \tag{34}$$

With an imperfect interface, U is a sum of strain energy and interfacial energy (kinetic energy is zero in this static limit). For this problem, $\Delta V / V_0 = 3\varepsilon_0$ and $V = l^3$ (where l is the side of the cube). The bulk modulus can therefore be found from:

$$K = \frac{2(U_{strain} + U_{interface})}{9 l^3 \varepsilon_0^2} \tag{35}$$

Figure 12 shows K found by energy analysis using various element sizes and with or without the recommended method to limit forces when D_n or D_t are high. The horizontal line labeled "CSA" is the predicted modulus for a particulate composite with a perfect interface using the "composites spheres assemblage" model (Hashin (1962)). When forces are not limited, K rises above the theoretical limit and keeps

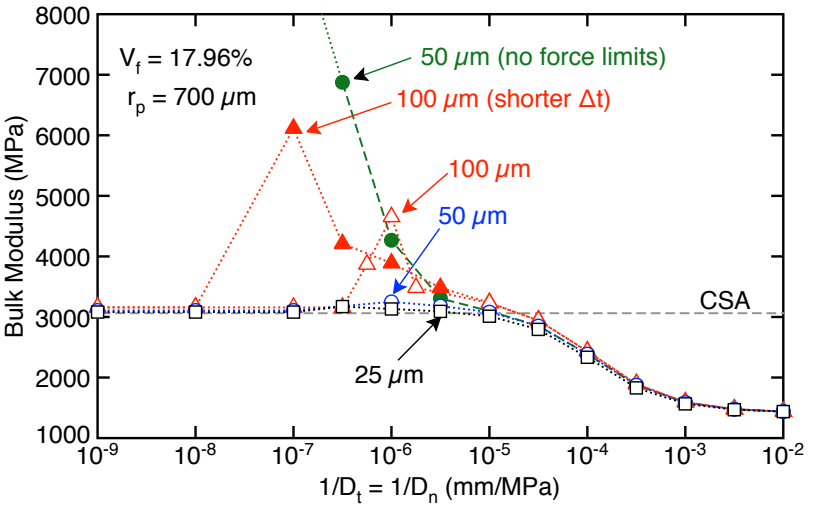

Figure 12: Bulk modulus in a cubic array of spherical particles (i.e., a particulate composite) of microparticles ($r_p = 700$ μm) and nanoparticles ($r_p = 700$ nm) determined by energy methods and using various cell sizes, time steps, and with or without force limitation (as indicated). The dashed horizontal line is the analytical composite spheres assemblage (CSA) result for a perfect interface.

rising until the calculations become unstable. All other results limited forces and remained stable for any values of D_n and D_t, but the results for a crude mesh (100 μm cells) had an erroneous peak in the transition region around $1/D_t = 1/D_n = 10^{-6}$ mm/MPa. The reason for the peak is that 100 μm cells do not resolve a spherical surface of radius 700 μm well enough for accurate energy results. The calculations were improved by either a reduction in the time step (which shifts the peak to higher D_t) or by smaller cells. Smaller cells worked much better because the peak is nearly gone for 50 μm cells and is entirely gone for 25 μm cells. These converged results are composite mechanics analysis for bulk modulus of a particulate composite as a function of interface properties. The results are close to the theoretical limit for high $D_t = D_n$, decrease as D_t and D_n decreases, and reach a low plateau value for a very poor interface (*i.e.*, debonded interface for $D_t = D_n \to 0$)

The particulate composite results have uniform average stress, $\langle \sigma_{xx} \rangle = \langle \sigma_{yy} \rangle = \langle \sigma_{zz} \rangle = P$, where P is pressure; thus K can also be found from

$$K = \frac{P}{3\varepsilon_0} \tag{36}$$

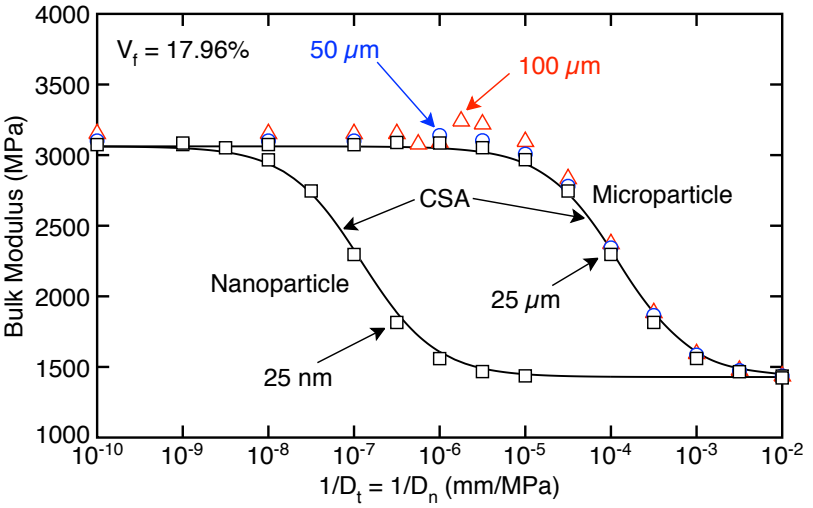

Figure 13: Bulk modulus in a cubic array of spherical particles (i.e., a particulate composite) of microparticles ($r_p = 700$ μm) and nanoparticles ($r_p = 700$ nm) determined by average stress over applied volumetric stain and using various cell sizes (as indicated). The solid lines are analytical results using a composite spheres assemblage (CSA).

Figure 13 shows K found from pressure for various element sizes and two different particle sizes. For microparticles (700 μm) all element sizes worked well, although 100 μm was rather crude and had minor errors in the transition region around $1/D_t = 1/D_n = 10^{-6}$ mm/MPa. The results for 50 μm or 25 μm were nearly identical and gave smooth curves. The solid lines are analytical modeling for a composites spheres assemblage with an imperfect interface (Hashin (1991b)). The numerical modeling agrees with the analytical modeling.

The "Nanoparticle" curve shows the modulus for a particulate composite of nanospheres (with radius 700 nm) as a function of interface properties using 25 nm cells. Hashin (1991b) suggests that $D_n = (K_i + 4G_i/3)/t_i$ where K_i and G_i are bulk and shear modulus of the interphase and t_i is its thickness. Physically, K_i, G_i, and t_1 are determined by interactions between the phases (*e.g.*, between glass and polymer), but they should not be affected by radius of curvature of the surface. In other words, D_n and D_t should be independent of particle size. When this relation holds, Fig. 13 shows that bulk modulus of a nanoparticle composite is always lower than the corresponding microparticle composite. Nanocomposites research frequently notes a scaling effect that nanocomposites have much more interface per unit volume re-

inforcement than conventional composites and nanocomposites hype assumes this extra interface must be beneficial. Contrary to that hype, the straightforward composite mechanics in Fig. 13 shows that extra interface is always detrimental for mechanical properties. A better picture emerges by realizing the interface in composites is burdened with transferring stress between phases. The extra interfacial area in nanocomposites means their interfacial properties carry an extra burden. In other words, when making new nanocomposites, the interface better be excellent, otherwise the mechanical properties are likely to be poor. Furthermore, interface-quality needs are much higher for nanocomposites then when making a corresponding composite with larger phases. For example, consider a glass-polymer system with $D_t = D_n = 10^6$ MPa/mm in Fig. 13. For a nanocomposite, this material would have very poor interface with K near the plateau for a completely debonded interface. In contrast, a composite with microparticles would have K near the theoretical limit for a perfect interface.

All examples in this paper used a linear elastic interface (*e.g.*, $T_n = D_n[u_n]$) and calculated elastic interfacial energy (see Eq. (4)). Obviously it is trivial to replace these assumptions with any non-linear interfacial traction laws and energy integral. For example, the normal direction might use a bilinear law where D_n is very large or perfect in compression (to avoid interpenetration) but has a lower stiffness in tension. But, interpenetration does not need to be prohibited. A 2D imperfect interface represents an actual 3D interphase and it is possible to compress such an interphase, which would correspond to a small negative $[u_n]$ when loaded in compression. An interfacial traction law that is history dependent, such as a plasticity based load with hardening or an interface that fails at some critical discontinuity, requires new methods. In multimaterial MPM used here, interacting nodes are re-calculated on each time step and carry no history. Such methods do not allow history dependent traction laws. One solutions is to use explicit cracks in MPM with tractions laws. By this approach, traction laws are carried on crack particles and therefore can track their history. Those traction laws can implement history-dependent plasticity and failure at critical crack opening displacements. In other words, they can be used as a particle-based approach to cohesive zone elements in finite element analysis.

4 Conclusions

Extension of multimaterial MPM methods for imperfect interfaces is simple in concept — calculate displacement discontinuities at interfaces and add internal forces determined by interfacial tractions laws. But, implementation details are subtle and important. If these details are ignored, the results can be poor. When they are included, however, imperfect interface modeling is a robust extension of MPM. An advantage over other numerical methods (*e.g.*, FEA) is that imperfect interfaces are

dynamically evaluated on each time step, thereby avoiding the need to use contact or interface elements in the mesh. The key details needed for accurate MPM interface modeling are:

Grid Independence: For results to be independent of interface location within the MPM grid, interfacial contact area has to be corrected by comparing nodal domains for the two material using Eq. (20). For an irregular mesh (*i.e.*, $\Delta x \neq \Delta y$ or $\neq \Delta z$), an additional correction is needed to account for angled interfaces, which is done by replacing Δx in Eq. (20) with $h_\perp$ in Eq. (22).

Stiff Interface Issues: As D_n or D_t approach ∞, their high stiffness would make it difficult to get converged results. A good solution is to limit interfacial forces to the maximum allowed forces defined by Eq. (26).

Interface Normals: Calculation of interfacial normal is crucial to accurate results and the optimum method is problem specific. The SN method is always best and robust, but only works for specialized problems. For other problems, MVG and AVG provide two alternatives. If neither D_n nor D_t are too high, both work well. If either D_n or D_t become stiff, however, problems can arise. These problems are usually confined to edge nodes, which can be screened out by ignoring interface calculations at nodes with Ω_i below some cutoff value. These conclusions apply to MPM contact methods as well, which can be regarded as always having $D_n = \infty$ during contact.

Interface Energy: Interfacial energy has no role in evolution of MPM results, but it is useful to track for composite mechanics calculations. The examples in this paper assumed a linear elastic imperfect interface with energy given by Eq. (4). If different traction laws are used, the interfacial energy calculation has to be changed, otherwise the a sum of strain energy, kinetic energy, and interfacial energy would not correspond to the actual total energy in the object.

Acknowledgement: This work was supported in part by the National Research Initiative of the United States Department of Agriculture (USDA) Cooperative Research, Education and Extension Service, Grant 2006-35504-17444 and by a grant from the National Science Foundation (award 1161305). The author also thanks Scott Bardenhagen and Jim Guilkey for many helpful discussions.

References

Angel, Y. C.; Achenbach, J. D. (1985): Reflection and transmission of elastic waves by a periodic array of cracks. *J. Appl. Mech.*, vol. 52, pp. 33–41.

Bardenhagen, S. (2012): personal communication, 2012.

Bardenhagen, S. G.; Brackbill, J. U. (1998): Dynamic stress bridging in granular material. *Journal of Applied Physics*, vol. 83, pp. 5732–5740.

Bardenhagen, S. G.; Guilkey, J. E.; Roessig, K. M.; Brackbill, J. U.; Witzel, W. M.; Foster, J. C. (2001): An improved contact algorithm for the material point method and application to stress propagation in granular materials. *Computer Modeling in Engineering & Sciences*, vol. 2, pp. 509–522.

Bardenhagen, S. G.; Kober, E. M. (2004): The generalized interpolation material point method. *Computer Modeling in Engineering & Sciences*, vol. 5, pp. 477–496.

Guilkey, J. (2010): personal communication, 2010.

Hashin, Z. (1962): The elastic moduli of heterogeneous materials. *J. Appl. Mech.*, vol. 29, pp. 143–150.

Hashin, Z. (1990): Thermoelastic properties of fiber composites with imperfect interface. *Mech. of Materials*, vol. 8, pp. 333–348.

Hashin, Z. (1991): Composite materials with viscoelastic interphase: Creep and relaxation. *Mech. of Materials*, vol. 11, pp. 135–148.

Hashin, Z. (1991): Thermoelastic properties of particulate composites with imperfect interface. *Journal of the Mechanics and Physics of Solids*, vol. 39, no. 6, pp. 745–762.

Hashin, Z. (1992): Extremum principles for elastic heterogenous media with imperfect interfaces and their application to bounding of effective moduli. *Journal of the Mechanics and Physics of Solids*, vol. 40, no. 4, pp. 767–781.

Le, E.; Nairn, J. (2012): Measuring interfacial stiffness of adhesively-bonded wood. submitted to *Wood Science & Technology*, 2012.

Lemiale, V.; Hurmane, A.; Nairn, J. A. (2010): Material point method simulation of equal channel angular pressing involving large plastic strain and contact through sharp corners. *Computer Modeling in Eng. & Sci.*, vol. 70, no. 1, pp. 41–66.

Nairn, J. A. (2003): Material point method calculations with explicit cracks. *Computer Modeling in Engineering & Sciences*, vol. 4, pp. 649–664.

Nairn, J. A. (2004): Generalized shear-lag analysis including imperfect interfaces. *Advanced Composite Letters*, vol. 13, pp. 263–274.

Nairn, J. A. (2007): Numerical implementation of imperfect interfaces. *Computational Materials Science*, vol. 40, pp. 525–536.

Nairn, J. A. (2009): Analytical and numerical modeling of r curves for cracks with bridging zones. *Int. J. Fract.*, vol. 155, pp. 167–181.

Nairn, J. A.; Guilkey, J. E. (2013): Axisymmetric form of the generalized interpolation material point method. *Int. J. for Numerical Methods in Engineering., submitted*.

Nairn, J. A.; Liu, Y. C. (1997): Stress transfer into a fragmented, anisotropic fiber through an imperfect interface. *Int. J. Solids Structures*, vol. 34, pp. 1255–1281.

Needleman, A. (1987): A continuum model for void nucleation by inclusion debonding. *J. Appl. Mech.*, vol. 54, pp. 525–531.

Sulsky, D.; Chen, Z.; Schreyer, H. L. (1994): A particle method for history-dependent materials. *Comput. Methods Appl. Mech. Engrg.*, vol. 118, pp. 179–186.