INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2017; **109**:1013–1044
Published online 15 August 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.5317

Field-gradient partitioning for fracture and frictional contact in the material point method

Michael A. Homel*,† and Eric B. Herbold

Computational Geosciences Group, Lawrence Livermore National Laboratory, Livermore, CA 94550, USA

SUMMARY

Contact and fracture in the material point method require grid-scale enrichment or partitioning of material into distinct velocity fields to allow for displacement or velocity discontinuities at a material interface. A new method is presented in which a kernel-based damage field is constructed from the particle data. The gradient of this field is used to dynamically repartition the material into contact pairs at each node. This approach avoids the need to construct and evolve explicit cracks or contact surfaces and is therefore well suited to problems involving complex 3-D fracture with crack branching and coalescence. A straightforward extension of this approach permits frictional 'self-contact' between surfaces that are initially part of a single velocity field, enabling more accurate simulation of granular flow, porous compaction, fragmentation, and comminution of brittle materials. Numerical simulations of self contact and dynamic crack propagation are presented to demonstrate the accuracy of the approach. Copyright © 2016 John Wiley & Sons, Ltd.

Received 18 December 2015; Revised 3 June 2016; Accepted 6 June 2016

KEY WORDS: material point method (MPM); fracture; cracks; contact; friction; fragmentation; comminution; mesoscale modeling; continuum damage; CPDI

1. INTRODUCTION

Dynamic fracture resulting from the impact of brittle materials has received increasing attention following improvements in armor technology dating back to World War I [1]. Many aspects of dynamic fracture persist as frontiers for research in numerical methods and range from crack curvature, bifurcation, and coalescence to shear banding and comminution. These phenomena are particularly complex to interpret because they may occur in response to intrinsic (e.g., material properties or heterogeneity) and/or extrinsic factors (e.g., loading/boundary conditions and environmental changes). Impact simulations often require treatment of frictional contact, complex continuum constitutive response under severe deformation, and 3-D fracture branching and coalescence. Continuum damage models provide a practical approach for simulating brittle failure while avoiding the complexity of explicitly tracking fracture surface interactions.

The need for simulating dynamic fracture and frictional contact arises in diverse applications including modeling of hypervelocity penetration [2–4], ballistic protection [1, 5, 6], planetary defense [7–11], machining [12], traumatic injury [13], and geotechnical applications [14]. In modeling fragmentation, the energy dissipation during the fracture process and subsequent frictional contact will determine the post-failure velocity and size distribution of the fragments [15, 16]. Similarly, in modeling penetration, the granular flow of failed material near the penetrator significantly affects the penetration response [17–19]. In many cases, the continuum response of the bulk material has a high degree of complexity and can be adequately described only by continuum models that maintain material state history variables [20–23].

*Correspondence to: Michael A. Homel, Computational Geosciences Group, Lawrence Livermore National Laboratory, Livermore, CA 94550, USA.

†E-mail: homel1@llnl.gov

The need to model large-deformation and fracture also arises in mesoscale[‡] simulations of porous and brittle materials, which facilitate the development of higher-fidelity, mechanics-based continuum models ([24–26]). Proper treatment of fracture and subsequent frictional sliding at fracture surfaces is important since crack interactions may significantly affect the dynamic strength of brittle materials under confinement [18, 27].

A variety of computational methods to simulate fracture and large deformation have been developed for applications in solid mechanics, but each has its limitations with respect to either accuracy or robustness when simulating large-deformation fracture with subsequent frictional contact in complex 3-D geometries. A promising approach is the material point method (MPM) [28], which tracks Lagrangian particles on an Eulerian grid, and which has been previously shown to be accurate for solid mechanics problems involving high rates [2, 29], large-deformations [30], and complex constitutive models [31].

A single velocity-field MPM solution can produce artificial 'welding' of material interfaces or nonphysical localization when the constitutive response includes softening or damage [32]. Using multiple velocity fields for separate bodies, frictional contact is readily achieved [33], and fracture can be simulated using either an explicit description of cracks [34, 35] or cohesive laws [36]. However, such methods are challenging to implement for evolving topologies of complex intersecting fracture networks in 3-D. A new method is presented here to simulate fracture and frictional self-contact in the MPM, without requiring an explicit description of the crack geometry or an *a priori* definition of contacting bodies.

1.1. An overview of the material point method

The MPM is a mixed Eulerian–Lagrangian approach in which the material state is tracked at Lagrangian particles and the equations of motion are solved on an Eulerian background grid [28]. A complete description of the MPM including a connection to the traditional finite element 'weak' form and a generalization of various MPM approaches is given by Sadeghirad *et al.* [37]. Key steps are summarized in the following discussion to provide a framework for discussing the modifications needed to achieve the desired fracture and contact response.

At the start of the time step, the mass and momentum are mapped from the particles to a background grid. The mass (m_i) and velocity $(\boldsymbol{v}_i)$ for the i^{th} node of the background grid are:

$$m_i = \sum_p \overline{S_{ip}^{\star}} m_p, \tag{1}$$

$$\boldsymbol{v}_i = \frac{1}{m_i} \sum_p \overline{S_{ip}^{\star}} m_p \boldsymbol{v}_p, \tag{2}$$

where m_p and $\boldsymbol{v}_p$ are the mass and velocity of the p^{th} particle and $\overline{S_{ip}^{\star}}$ is the average of the i^{th} grid shape function $S_i^{\star}$ over the p^{th} particle domain. In the original formulation of the MPM [28], this average was computed with single-point integration, but this 'Dirac delta' formulation led to spurious cell-crossing errors. The generalized interpolation material point method [38] improves accuracy by representing the particle domain as a finite domain but does not fully eliminate cell-crossing errors. The convective particle domain interpolation (CPDI) method [39] improves accuracy of the force calculations and particle-grid mapping over earlier generalized interpolation material point methods by representing the particle integration domain as a deforming parallelepiped, for which the shape function averages can be readily computed. In our work, we use

[‡] The term 'mesoscale' has been assigned a range of meanings by various authors. In the context of this paper, we will use the term to refer to simulations that are intermediate in scale between continuum and atomistic representations; that is, they explicitly resolve morphological features such as individual pores, grains, aggregates, phase heterogeneity, and cracks but may still apply a continuum representation to material regions.

DOI: 10.1002/nme

the CPDI formulation along with a domain scaling modification [32] that allows for efficient parallelization, while controlling nonphysical numerical fracture. The background grid is a structured, Cartesian mesh, with tri-linear shape functions, which simplifies evaluation of $S_i^{\star}$.

The internal and external forces (neglecting surface tractions) at the i^{th} grid node are

$$\boldsymbol{f}_i^{ext} = \sum_p \overline{S_{ip}^{\star}} \boldsymbol{b}_p m_p, \tag{3}$$

$$\boldsymbol{f}_i^{int} = -\sum_p \overline{\nabla S_{ip}^{\star}} \cdot \boldsymbol{\sigma}_p V_p, \tag{4}$$

where $\boldsymbol{b}_p$ is the body force acting on the p^{th} particle, V_p is the particle volume, $\boldsymbol{\sigma}_p$ is the Cauchy stress at the particle, and $\overline{\nabla S_{ip}^{\star}}$ is the average of the gradient of the i^{th} grid shape function over the p^{th} particle domain. The accelerations are computed on the background grid,

$$\boldsymbol{a}_i = \frac{\boldsymbol{f}_i^{int} + \boldsymbol{f}_i^{ext}}{m_i}, \tag{5}$$

from which the updated nodal velocities are computed,

$$\overline{\boldsymbol{v}}_i^{n+1} = \boldsymbol{v}_i^n + \boldsymbol{a}_i \Delta t. \tag{6}$$

In Equation (6), the updated velocity, $\overline{\boldsymbol{v}}_i^{n+1}$, neglects the effect of contact forces. As described in the following section, contact forces modify the velocity field to compute a corrected $\boldsymbol{v}_i^{n+1}$ from $\overline{\boldsymbol{v}}_i^{n+1}$. The updated grid velocity field is used to compute the updated velocity gradient at the particle,

$$\boldsymbol{L}_p^{n+1} = \Sigma_i \boldsymbol{v}_i^{n+1} \otimes \overline{\nabla S_{ip}^{\star}}. \tag{7}$$

The updated particle velocity gradient is used to update the particle deformation gradient[§],

$$\boldsymbol{F}_p^{n+1} = \boldsymbol{F}_p^n + \left(\boldsymbol{L}_p^{n+1} \cdot \boldsymbol{F}_p^n\right) \Delta t. \tag{8}$$

The constitutive model is then called with the updated deformation gradient and velocity gradient to compute the updated particle stress $\boldsymbol{\sigma}_p^{n+1}$ and internal state variables. This is the ‘update stress last’ formulation, which has previously been shown to be most accurate for dynamic simulation with explicit time-stepping schemes [40]. The end-of-step nodal velocities and accelerations are used to update the position and velocity of the particles,

$$\boldsymbol{v}_p^{n+1} = \boldsymbol{v}_p^n + \Sigma_i \overline{S_{ip}^{\star}} \boldsymbol{a}_i \Delta t, \tag{9}$$

$$\boldsymbol{x}_p^{n+1} = \boldsymbol{x}_p^n + \Sigma_i \overline{S_{ip}^{\star}} \boldsymbol{v}_i^{n+1} \Delta t. \tag{10}$$

There are four key attributes of the MPM [41]:

(1) Model generation is relatively simple and eliminates the need for mesh generation. For example, it is straightforward to accommodate voxelized phase topologies directly from X-ray computed microtomography data.
(2) The material state history is stored at the particles, so that advection errors are eliminated.
(3) The background grid is reset at each time step, so the method tolerates large deformation.
(4) Frictional contact is efficient and easily implemented as described in the following section.

[§] To improve robustness and accuracy in extreme deformation (for finite element methods in general, not just the MPM), the integration of $\boldsymbol{F}_p$ can be computed with smaller substeps or using a higher-order integration. This substepping does not affect the algorithm time step and has negligible computational expense.

1.1.1. Traditional approach for contact in the material point method. The MPM algorithm outlined in Equations 1–10 computes the response for material in a single velocity field. Within this velocity field, the method naturally produces a no-slip contact condition that prevents interpenetration between materials. This behavior is similar to the response of an Eulerian hydrocode, which also assumes that each computational cell contains a single velocity gradient when multiple materials are present.

If frictional contact is an important consideration between two or more regions or material groups, then the material in each contact group is assigned to its own velocity field. The solution within a contact-group velocity field is identical to that for a single-field (SF) system, except when material from multiple velocity fields maps to the same node. In this case, a contact force $\boldsymbol{f}_i^{\mathrm{ct}}$ is computed at the shared node, which modifies the velocity in Equation 6. This is illustrated in Figure 1.

The nodal contact force is computed to: (i) prevent interpenetration; (ii) allow frictional sliding if the slip condition is met; and (iii) allow separation of the contact pair. For a given node at which two or more velocity fields have nonzero mass, there is a pairwise nodal interaction forces between each field. In our work, we dynamically repartition material into separate velocity fields at each node to define fracture surfaces or to accommodate frictional self-contact. Once the material has been appropriately partitioned, the subsequent contact algorithm for the step is identical to that for contact between predetermined material pairs as described by Bardenhagen *et al.* [42]. The method was later modified [2] to improve accuracy by rigorous enforcement of an interpenetrability condition through the step. The frictional contact algorithm is briefly summarized here to facilitate a discussion of the fracture and self-contact modifications.

For velocity fields $\varsigma = 1, 2$, we define an outward surface normal based on the normalized local density gradient in the field,

$$\boldsymbol{n}_{\varsigma i} = \frac{\sum_p \overline{\nabla S^*_{ip}} m_{\varsigma p}}{\| \sum_p \overline{\nabla S^*_{ip}} m_{\varsigma p} \|}, \tag{11}$$

where $m_{\varsigma p}$ is the mass of the p^{th} particle in the ς^{th} velocity field.

For perfect conservation of linear momentum, it is necessary that the surface normals for a contact pair are exactly opposite, so $\boldsymbol{n}_{1i} = -\boldsymbol{n}_{2i}$. This will not generally be the case, so we define corrected surface normals $\boldsymbol{n}^{\star}_{1i} = -\boldsymbol{n}^{\star}_{2i} = \frac{1}{2}(\boldsymbol{n}_{1i} - \boldsymbol{n}_{2i})$, and then normalize the corrected values. In some cases, a mass or volume weighting of the surface normals may improve the response, particularly when one field is present in a very small fraction, but this was not carried out in our present work. Throughout the remainder of this manuscript, we will omit the * superscript with the understanding that all nodal surface normals have been corrected.

The center-of-mass nodal velocity for the contact pair is computed from the total momentum and mass at the node:

$$\boldsymbol{v}_i^{cm} = \frac{\Sigma_{\varsigma=1}^2 m_{\varsigma i} \overline{\boldsymbol{v}}_{\varsigma i}}{\Sigma_{\varsigma=1}^2 m_{\varsigma i}}, \tag{12}$$

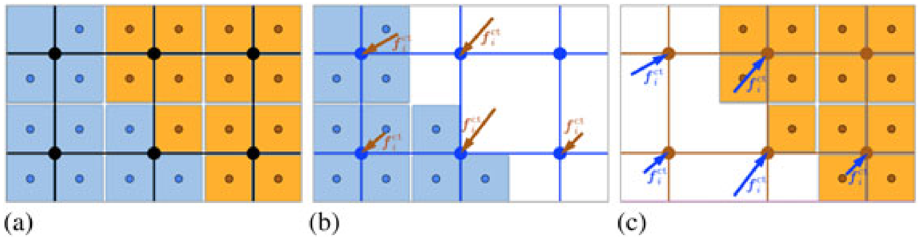

Figure 1. Diagram of the traditional approach for contact with prescribed material contact groups: (a) contact between two material groups, (b) particles in first velocity field with nodal contact force from second field, and (c) particles in second velocity field with nodal contact force from first field.

where $\overline{\boldsymbol{v}}_{\varsigma i}$ indicates the velocity for the ς^{th} velocity field before accounting for contact forces. With contact forces $\boldsymbol{f}^{ct}_{\varsigma i}$, we define

$$\boldsymbol{v}_{\varsigma i} = \overline{\boldsymbol{v}}_{\varsigma i} + \frac{\boldsymbol{f}^{ct}_{\varsigma i}}{m_{\varsigma i}} \Delta t. \tag{13}$$

The compressive normal force at the i^{th} node is subject to the Karush–Kuhn–Tucker conditions, $f_i^{\text{nor}} \geqslant 0$, $g_i \leqslant 0$, and $f_i^{\text{nor}} g_i = 0$, where g_i is the nodal overlap between the contact surfaces [43]. The kinematic constraint is enforced by applying a contact force directly at the node if interpenetration is detected (i.e., if $\sum_{\varsigma=1}^{2} \boldsymbol{n}_{\varsigma i} \cdot \boldsymbol{v}_{\varsigma i} \leqslant 0$). Otherwise, no contact force is applied, which naturally allows separation of contact surfaces. The normal component of the compressive contact force is

$$f^{\text{nor}}_{\varsigma i} = \boldsymbol{f}^{ct}_{\varsigma i} \cdot \boldsymbol{n}_{\varsigma i} = \frac{m_{\varsigma i}}{\Delta t} \left(\boldsymbol{v}_i^{cm} - \overline{\boldsymbol{v}}_{\varsigma i} \right) \cdot \boldsymbol{n}_{\varsigma i}. \tag{14}$$

The tangent force is defined in terms of unit vectors $\boldsymbol{s}^1_{\varsigma i}$ and $\boldsymbol{s}^2_{\varsigma i}$, which form an orthonormal basis with $\boldsymbol{n}_{\varsigma i}$. The orientation of $\boldsymbol{s}^1_{\varsigma i}$ and $\boldsymbol{s}^2_{\varsigma i}$ in the tangent plane is arbitrary for the case of simple isotropic friction, but they must have a consistent definition between contact pairs (i.e., $\boldsymbol{s}^1_{1i} = \boldsymbol{s}^1_{2i}$ and $\boldsymbol{s}^2_{1i} = \boldsymbol{s}^2_{2i}$). The tangential force for non-slip contact is

$$f^{1,\tan}_{\varsigma i} = \boldsymbol{f}^{ct}_{\varsigma i} \cdot \boldsymbol{s}^1_{\varsigma i} = \frac{m_{\varsigma i}}{\Delta t} \left(\boldsymbol{v}_i^{cm} - \overline{\boldsymbol{v}}_{\varsigma i} \right) \cdot \boldsymbol{s}^1_{\varsigma i} \tag{15a}$$

$$f^{2,\tan}_{\varsigma i} = \boldsymbol{f}^{ct}_{\varsigma i} \cdot \boldsymbol{s}^2_{\varsigma i} = \frac{m_{\varsigma i}}{\Delta t} \left(\boldsymbol{v}_i^{cm} - \overline{\boldsymbol{v}}_{\varsigma i} \right) \cdot \boldsymbol{s}^2_{\varsigma i}. \tag{15b}$$

Defining $\boldsymbol{f}^{\tan} = f^{1,\tan}_{\varsigma i} \boldsymbol{s}^1_{\varsigma i} + f^{2,\tan}_{\varsigma i} \boldsymbol{s}^2_{\varsigma i}$, $f^{\tan}_{\varsigma i} = \|\boldsymbol{f}^{\tan}\|$, and $\boldsymbol{s}_{\varsigma i} = \boldsymbol{f}^{\tan} / f^{\tan}_{\varsigma i}$, the nodal frictional contact force is

$$\boldsymbol{f}^{ct}_{\varsigma i} = f^{\text{nor}}_{\varsigma i} \boldsymbol{n}_{\varsigma i} + \min \left(\mu |f^{\text{nor}}_{\varsigma i}|, |f^{\tan}_{\varsigma i}| \right) \operatorname{sgn} \left(f^{\tan}_{\varsigma i} \right) \boldsymbol{s}_{\varsigma i}. \tag{16}$$

The frictional contact capability in the MPM can be readily extended to accommodate an arbitrary number of contact groups by adding additional velocity fields and computing the pair-wise contact forces for each combination of the material groups. For N velocity fields, there are $N(N-1)$ possible contact pairs, so the computational cost and memory requirements for the contact algorithm become more significant as N becomes large. This cost can be mitigated by only computing interactions for fields that are active at a single node. A more important concern is that while pairwise force calculations ensure conservation of linear momentum, they do not strictly prevent interpentrability for $N > 2$ fields. We are currently developing other approaches to address this limitation, but this is beyond the scope of our current discussion.

When using the multi-field method to model frictional contact in granular materials (cf., [44]), it is necessary to initially partition the grains into separate velocity fields so that neighboring grains occupy separate fields. The four-color theorem states that any 2-D domain can be partitioned to avoid edge-to-edge contact of matching fields with no more than four fields [45]. In 3-D, there is no analogous theorem limiting the number of required fields to ensure frictional contact between all entities, and even if the geometry can be initialized such that neighboring particles are in separate velocity fields, subsequent deformation and granular flow can change the contact points.

Figure 2a illustrates that even in 2-D, a large number of velocity fields may be required to ensure frictional contact in a granular system. In contrast, if a surface detection contact approach is used (such as the new approach described in Section 2.4), granular flow can be simulated from a single material group (divided into two velocity fields) with surface-particle flags as illustrated in Figure 2b. Defining contact groups is more complicated for porous materials (Figure 2c) subjected to large deformation that induces 'self-contact' between portions of a pore's surface that are part of a contiguous body. Similar complications arise when maintaining frictional contact at fracture surfaces

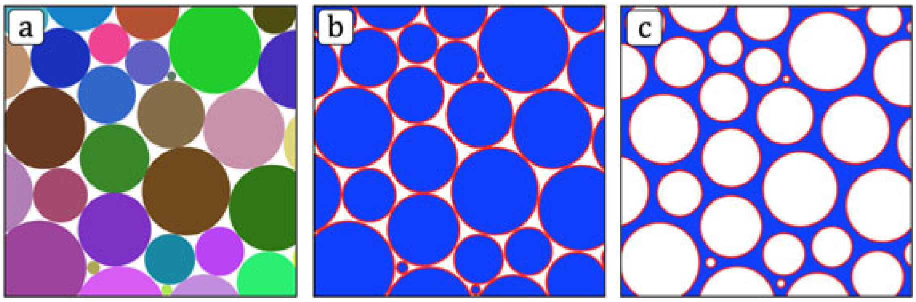

Figure 2. Velocity field partitioning for 2-D frictional contact in granular and porous systems: (a) traditional approaches require prescribing a large number of velocity fields so neighboring grains are in separate fields; (b) gradient partitioning with surface detection produces frictional contact from a single initial material group; (c) gradient partitioning allows frictional contact between internal pore surfaces (or fracture surfaces), which may come into contact following large deformations; because the contact surfaces are not known *a priori*, this frictional 'self-contact' is not possible with the traditional prescribed-field approach to contact in MPM.

(discussed in the following section), which is challenging for existing methods if displacement at the fracture surface creates new contact pairs.

In the current work, we present a method to dynamically repartition material into contact pairs for initial surfaces or those created by fracture processes. This allows for more realistic representation of self-contact in porous materials, as well as a representation of frictional dissipation in comminution and post-fracture granular flow.

1.1.2. Traditional approach for fracture in the material point method. There are an abundance of continuum constitutive models of varying complexity that have been developed to model brittle failure in a broad spectrum of materials [46]. Such continuum damage/softening models are readily implemented in 3-D, naturally accommodate branching and coalescence of failure surfaces and can produce realistic patterns of brittle failure when combined with spatially heterogeneous and size-dependent strength [47]. However, when implemented in an MPM framework, these continuum models may not perform as desired.

The need to make 'close observations in the material failure zone when softening occurs' was noted by Sulsky *et al.* [28] when the MPM was first introduced. The potential for poor performance is because multiple material points within a single grid cell share a common velocity field but do not necessarily have common material properties or loading history; spurious stresses may occur when both failed and intact material share a grid cell. This velocity-field-smearing effect can produce unexpected and undesirable behavior as described in [32]. For example, a tensile member containing a single layer of failed (or even deleted) particles can still support a tensile load, when the particle width is less than the grid cell spacing (as is generally the case). As a result, simulating fracture via particle deletion (akin to the element deletion practice in FEM [48]) can introduce significant error in addition to the loss of mass and momentum in the system. Additionally, the localization can be affected by the relative motion of the failed particles to the background grid [32].

An accurate treatment of fracture requires a representation of strong discontinuities in the velocity field. Enrichment of the MPM velocity field has been explored by introducing additional degrees of freedom at the particles [37], as a way to describe interfaces between regions with different material properties. This type of approach has not yet been extended to model propagating fractures.

Fracture has been successfully modeled in the MPM by Nairn *et al.* [34, 35], through introduction of explicit cracks that are used to partition the particles into two contacting velocity fields at each node, depending on the relative locations of the particle and crack to the node. This is illustrated in Figure 3. Once the particles have been partitioned at a fracture node, the solution for the step is identical to that for frictional contact described in the previous section, except that the crack geometry can be used to exactly define the surface normal, rather than relying on a mass-gradient calculation.

DOI: 10.1002/nme

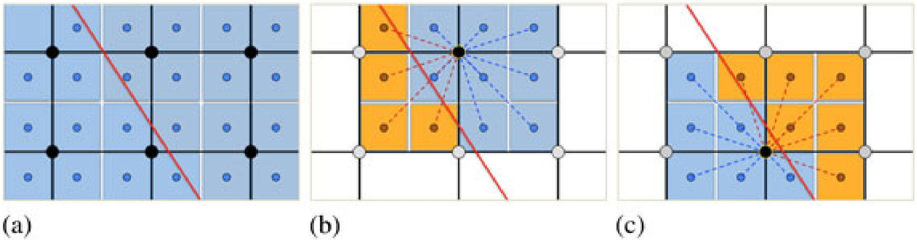

Figure 3. Partitioning of material into two fields around a fracture depends on whether the crack lies between a particle and a node [34]: (a) explicitly defined fracture in a plane of material, (b) particle partitioning at top center node, and (c) particle partitioning at bottom center node. The field to which a given particle maps may vary from node to node.

With explicit cracks, a variety of criteria can be used to nucleate or evolve the fracture surfaces; the explicit fracture approach has been shown to work well for single cracks in 2-D or 3-D [35], but it becomes difficult to implement for complex 3-D networks of branching and coalescing fractures. Additionally, while it is possible to update the crack position and orientation through time, this does not provide a framework for subsequent generalized frictional contact of fracture surfaces that have separated from their original opposing face to contact another body. Other approaches involving cohesive zones between regions [49] suffer similar limitations.

In the current work, we demonstrate a method for dynamic repartitioning of material into contact groups based on the current distribution of damage in the material. This approach mitigates the errors of a single velocity field, allowing the use of continuum damage models without the complexity of explicitly tracking fractures. Frictional effects can be readily enforced for failed surfaces during subsequent contact, regardless of whether the surfaces were originally generated from the same fracture.

2. METHODS

We propose a novel method for achieving fracture and fragmentation in the MPM, by using the gradients of a scalar damage field to partition the material into contact pairs along fracture surfaces. The damage-field gradient (DFG) partitioning method is inherently 3-D and naturally supports branching and coalescence of fracture networks. Partitioning material based on the gradient of an arbitrary scalar field enables frictional 'self-contact', meaning sliding contact between surfaces that were originally part of the same velocity field (e.g., pore surfaces or evolving fracture surfaces). The field-gradient partitioning method can be implemented efficiently within an existing MPM framework, it is amenable to efficient parallelization, and in most cases, it requires only a moderate increase in computational expense.

2.1. Construction of the scalar damage field

The particles along a fracture are partitioned into contact pairs based on the gradient of a continuous damage field, constructed from the some scalar damage measure ($\mathcal{D}_p$) in the particle.[¶] We construct a smooth scalar field by summing particle-centered kernel functions,

$$\mathcal{D}(\boldsymbol{x}) = \sum_{p=1}^{n_p} \mathcal{D}_p \omega\left(\bar{r}\right), \tag{17}$$

where ω is a kernel function with support radius r_p. For this work, a simple spherically symmetric cubic kernel is defined based on a normalized distance $\bar{r} = \left|\left|\boldsymbol{x} - \boldsymbol{x}_p\right|\right| / r_p$,

[¶] Generally, the term 'damage' refers to a loss of stiffness, whereas 'softening' refers to a loss of strength. In this manuscript, we shall use the term 'damage' more generally as a measure of material failure, so that a 'fully damaged region' can be interpreted as a failure surface, that is, a fracture.

DOI: 10.1002/nme

$$\omega(\overline{r}) = \begin{cases} 1 - 3\overline{r}^2 + 2\overline{r}^3 & : 0 \leqslant \overline{r} \leqslant 1 \\ 0 & : \overline{r} > 1 \end{cases}. \tag{18}$$

Except where otherwise noted, the neighbor radius is the diagonal length of a background grid cell, $r_p = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$ in 3-D and $r_p = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ for 2-D plane strain. As shown in Figure 4a, the simple kernel field has significant edge effects, which would complicate the analysis of damage and fracture. To eliminate edge effects, we divide by the kernel sum,

$$\mathcal{S}(\boldsymbol{x}) = \sum_{p=1}^{n_p} \omega\left(\left|\left|\boldsymbol{x} - \boldsymbol{x}_p\right|\right| / r_p\right). \tag{19}$$

With this definition, the normalized kernel field, $\overline{\mathcal{D}}(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x})/\mathcal{S}(\boldsymbol{x})$, accurately captures only the effects of the damage geometry, as shown in Figure 4c. The normalized field is undefined outside of the support domain of the kernel functions where $\mathcal{S}(\boldsymbol{x}) = 0$, but the damage field and its gradient are only evaluated at particle centers, where the field is well defined.

This type of kernel field construction is similar to that used in smoothed-particle hydrodynamics (SPH) [50], but here, we are only using the direction of the DFG to locate particles relative to fracture surfaces, and the kernel field is not used in the solution of the momentum equation. As such, we do not require that the kernel functions have partition of unity and do not require the more complex corrections used in the reproducing kernel particle method [51]. The kernel sum itself can be used to identify surface particles, as shown in Figure 4b; this can be useful when initializing a model for frictional self contact, as described in Section 2.4.

2.2. *Field-gradient material partitioning*

Once the scalar damage field is defined, we then partition the particles into contact pairs by the location of the particles relative to the fracture surface. While it is possible to use topology simplification (cf., [52]) to explicitly identify the fracture surfaces and intersection points, doing so would introduce unnecessary cost and complexity. Instead, we compare the direction of the DFG for the particles mapping to a node. If opposing gradients exist, the node is potentially at a contact surface and must allow for the possibility of frictional sliding or separation. Once the damage field has been constructed, we partition the particles at each node into one of two contact groups by comparing the direction of the DFG at the particle center, $\overline{\mathcal{D}}_p = \overline{\mathcal{D}}(\boldsymbol{x}_p)$.

It is physically intuitive to partition the material into subregions and to compute contact between these subregions, but the contact forces in the MPM are computed node-by-node, and it is not necessary that the ordering of the contact partitioning (i.e., whether the $\varsigma = 1$ field is above or

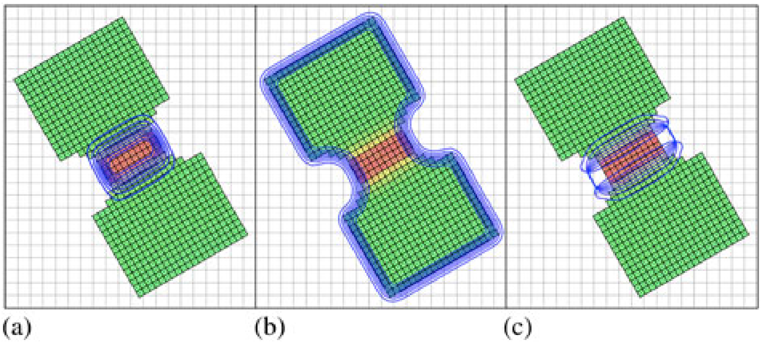

Figure 4. Damage field for a tensile bar. Particles colors indicate damage; overlaid contours show the (a) *unnormalized* kernel damage, field $\mathcal{D}(\boldsymbol{x})$, (b) surface detection with kernel sum, $\mathcal{S}(\boldsymbol{x})$, and (c) *normalized* kernel damage field, $\overline{\mathcal{D}}(\boldsymbol{x})$. Normalization removes the edge effects so that the damage-field gradient over the domain of the solid material describes the orientation of the failure surface.

DOI: 10.1002/nme

below the crack) be consistent along the fracture surface; it is only necessary that the orientation of the surface normal for each field at the node be consistent with the direction of the normal to the fracture or contact surface. Thus, node-wise partitioning can be applied as was performed for explicit cracks (Figure 3).

Evaluating the gradient of the damage field at a grid node, $\overline{\mathcal{D}}_g = \overline{\mathcal{D}}(\boldsymbol{x}_g)$, we can partition the particle data mapped to each node based on the sign of the dot product of the damage gradient at the particle and at the node, $\textbf{sgn}\left[(\nabla\overline{\mathcal{D}}_p)\cdot(\nabla\overline{\mathcal{D}}_g)\right]$. In many cases, this gives the desired partitioning, but this simple method fails when the grid node lies at the maximum of the scalar field and $\nabla\overline{\mathcal{D}}_g = \mathbf{0}$. This pathological case can occur quite frequently at the border of a fully damaged region.

As we view the contours of a damage field (Figure 5), we can readily identify the desired orientation of a fracture at a grid node, but this is because our eye captures the trends over a region, not simply the gradient at a point. To capture this nonlocal effect, and since the partitioning method depends only on the direction of $\nabla\overline{\mathcal{D}}_g^\star$ and not its magnitude, we formally define a nonlocal gradient $\nabla\overline{\mathcal{D}}_g^\star$ as a multiple of the unit vector $\left(\boldsymbol{n}_i^\star\right)$ that maximizes the integral of $|\boldsymbol{n}_i^\star\cdot\nabla\mathcal{D}(\boldsymbol{x})|$ over the neighborhood surrounding the grid cell. In general, there will be at least two solutions for $\boldsymbol{n}_i^\star$.

Finding the optimal value of $\boldsymbol{n}_i^\star$ is computationally expensive, but we have found that for determining the partitioning orientation it is sufficient to define the nonlocal DFG to be the gradient of the damage field evaluated at the point $\boldsymbol{x}^{\max}$ within some radius (r_g) for which the gradient has the greatest magnitude: $\nabla\overline{\mathcal{D}}_i^\star = \nabla\overline{\mathcal{D}}(\boldsymbol{x}_i^{\max})$, where

$$\boldsymbol{x}_i^{\max} = \underset{x\in R}{\arg\max}\,[\nabla\mathcal{D}(x)] \ni \|\boldsymbol{x}-\boldsymbol{x}_i\| < r_g. \tag{20}$$

This formulation was chosen because it allows for an efficient approximate evaluation (Algorithm 1). For a given node, $\nabla\overline{\mathcal{D}}_g^\star$ is set equal to the value of $\nabla\overline{\mathcal{D}}_p^\star$ for the particle (mapping to that node) that has the largest DFG magnitude, $\left\|\nabla\overline{\mathcal{D}}_p^\star\right\|$.

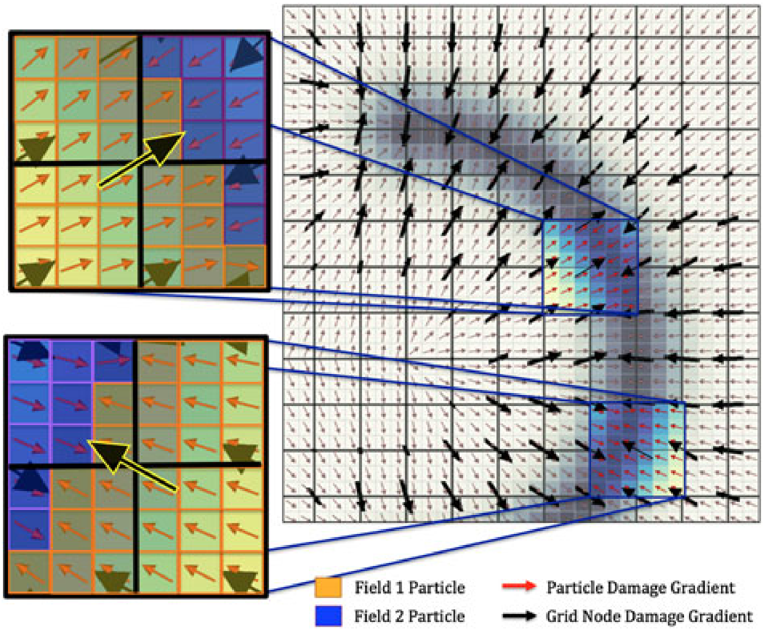

Figure 5. Node-wise damage-field gradient partitioning around a curved crack using a nonlocal gradient measure at the grid node. Inset images show the particle partitioning around a single grid node. The direction of the nonlocal damage gradient at the grid node $(\nabla\overline{\mathcal{D}}_i^\star)$ is shown as a highlighted black arrow. The surrounding particles are colored based on their partitioning at the node, as determined from the relative orientation of $\nabla\overline{\mathcal{D}}_i^\star$ to $\nabla\overline{\mathcal{D}}_p^\star$

Algorithm 1 Computing the approximate nonlocal grid-node damage-field gradient, $\nabla\overline{\mathcal{D}}_i^\star$

```
for $i = 1..n_i$ do                                   ▷ Loop over all grid nodes
    $\nabla\overline{\mathcal{D}}_i \leftarrow \mathbf{0}$      ▷ Initialize damage gradient at the grid to zero
end for
for $p = 1..n_p$ do                                   ▷ Loop over all particles
    for $i = i_{p1}..i_{pn}$ do                       ▷ Loop over mapped grid nodes for current particles
        if $\|\nabla\overline{\mathcal{D}}_p\| > \|\nabla\overline{\mathcal{D}}_i\|$ then   ▷ Compare magnitudes of DFG at particle and grid node
            $\nabla\overline{\mathcal{D}}_i \leftarrow \nabla\overline{\mathcal{D}}_p$   ▷ Overwrite grid damage-field gradient with particle value
        end if
    end for
end for
```

The partitioning of a curved-crack damage field using the approximate nonlocal grid-node DFG is shown in Figure 5. In our simulations, we defined the neighbor radius to be equal to the diagonal length of the background grid cell. The red arrows show the scaled DFG at the particles (there are three particles per grid cell in each direction), and the black arrows show the nodal DFG $\left(\nabla\overline{\mathcal{D}}_g^\star\right)$, which is approximately normal to the crack surface along the entire length of the crack, as desired. While it is necessary that $\nabla\overline{\mathcal{D}}_g^\star$ be approximately normal to the crack surface, the direction can be reversed without affecting the algorithm. At the crack tip, the direction of $\nabla\overline{\mathcal{D}}_g^\star$ is no longer normal to the semicircular crack, but here the separability condition (described in the next section) would not be met, so the orientation of $\nabla\overline{\mathcal{D}}_g^\star$is irrelevant.

Once the DFG has been defined at the particles and grid nodes, it can be used to partition particle contributions at each grid node as shown in the inset frames of Figure 5. This is performed within the loops over particles in which data are mapped between the particles and grid. For a given particle-node pair, the dot product of the DFG at the particle and grid node is computed, the sign of which is used to determine the field to/from which the particle will map. An example of this is given in Algorithm 2, which provides pseudocode for mapping particle mass to the grid. Similar algorithms are used for mapping the momentum, accumulating forces, and for mapping velocity and acceleration from the grid to the particles.

Algorithm 2 Pseudo-code for implementation of damage-field gradient partitioning in a particle-to-grid mapping loop

```
for $p = 1..n_p$ do                                   ▷ Loop over all particles
    for $i = i_{p1}..i_{pn}$ do                       ▷ Loop over mapped grid nodes for current particles
        if $\nabla\overline{\mathcal{D}}_p \cdot \nabla\overline{\mathcal{D}}_i \geqslant 0$ then
            $\varsigma \leftarrow 0$                  ▷ Set active field to zero
        else
            $\varsigma \leftarrow 1$                  ▷ Set active field to one
        end if
        $m_{\varsigma i} \leftarrow m_{\varsigma i} + \overline{S_{ip}^\star} m_p$   ▷ Add mass to appropriate velocity field
    end for
end for
```

It is important to note that while the DFG vector is approximately normal to the failure surface, it is used only to define the topology by partitioning the material into contact groups, and it does not define the surface normal. Once the material is partioned, the normal for subsequent contact interactions is defined by Equation 11. As such, we expect the accuracy of the surface normals to be comparable with that of the surface representation in generalized MPM contact, improving with the resolution of the discretization relative to the curvature of the surface.

This approach for material partitioning assumes that a local contact plane can be defined at each node to describe the contact interactions of two material regions (i.e., the fracture faces) at that node. For crack intersection and branching, there can be three or more regions of material meeting at a single point. However, a crack in a smeared-damage model has a finite width, because fracture and contact are resolved only to the resolution of the grid cell. As a result, crack intersections generally manifest as grid cells containing fully damaged particles. One could design a perfectly symmetric problem, for which the crack plane orientation within the intersection plane is arbitrary, but in practice, any deformation of the material will perturb the damage field, allowing the fracture surface to orient in a direction to allow separation/sliding at the intersection point. Section 3.4 shows simulation results including crack branching and coalescence. Here, the crack paths are dictated by the spatial heterogeneity, stress state, and current fracture topology, and appear to be largely independent of mesh bias.

2.3. Separability condition

The DFG is used to partition material into two fields at each node containing material from both sides of a damaged region. A *separability condition* is defined to determine whether the damage state at the node should be treated as a crack (i.e., two fields in contact) or whether the material state should locally be described by a single velocity field.

For the topology to be consistent with the constitutive model predictions, the material should be allowed to separate only once a 'surface' of failure particles has formed. To estimate if this has occurred, we first evaluate the average damage for each velocity field at a given node from the mass-weighted average of the particle damages:

$$\overline{\mathcal{D}}_{\varsigma i} = \frac{\sum_{\varsigma p} \phi_{ip} \mathcal{D}_p m_p}{\sum_{\varsigma p} \phi_{ip} m_p}, \tag{21}$$

where $\sum_{\varsigma p}$ indicates a sum over all particles in the ς^{th} velocity field mapping to the i^{th} node. We also compute the maximum value $\left(\overline{\mathcal{D}}_i^{\max}\right)$ of damage for the particles mapping to the node in each velocity field.

For the examples in this paper, separability at a node requires that (i) at least one particle mapping to the node is fully damaged, i.e., $\overline{\mathcal{D}}_i^{\max} = 1$ and (ii) the minimum value of the average damage exceeds a specified separability threshold $\min\left(\overline{\mathcal{D}}_{1i}, \overline{\mathcal{D}}_{2i}\right) > \overline{\mathcal{D}}^{\min}$, for $0 < \overline{\mathcal{D}}^{\min} < 1$. The effect of the separability limit on fracture response is demonstrated in Section 3.3.

Algorithm 3 provides pseudocode for computing pairwise frictional contact forces with the separability criterion in the DFG-partitioning method. This assumes that the both the average damage $\left(\overline{\mathcal{D}}_{1i}, \overline{\mathcal{D}}_{2i}\right)$ and the maximum damage $\left(\overline{\mathcal{D}}_{1i}^{\max}, \overline{\mathcal{D}}_{2i}^{\max}\right)$ have been previously computed. User inputs are the separability threshold $(0 < \overline{\mathcal{D}}^{\min} < 1)$ and the coefficient of friction $(\mu > 0)$.

This example assumes there are only two velocity fields, those prescribed by the DFG partitioning. In general, there may be an arbitrary number of 'contact groups', regions of the domain that are initialized to be separate velocity fields so that a specific contact law (e.g., a different coefficient of friction) can be prescribed between each of these groups. In this case, each of these initial fields can then be further partitioned into one of two damage state fields. When computing field interactions, the separability condition is always met for two velocity fields that were initially different contact groups.

In this work, we have considered only scalar damage models (Section 2.7); however, it may be possible to extend the DFG-partitioning method to accomodate models with anisotropic tensor damage. The DFG method identifies potential separation/sliding surfaces based on regions of damaged material. If an anisotropic damage model is used, the results will only be physically reasonable if the constitutive model failure plane aligns with the separation surfaces. For certain anisotropic modes of failure, this may not always be the case. To account for anisotropy in a tensor damage model, a separability criterion could be developed that accounts for the alignment of the failure planes in the material and the DFG. This is a potential area for future development that would extend the applicability of this method to a broader class of materials.

Algorithm 3 Frictional contact algorithm with separability criterion

```
for $i = 1..n_i$ do                                                          ▷ Loop over all grid nodes
    if $(m_{1i} > \epsilon_m) \wedge (m_{2i} > \epsilon_m)$ then              ▷ If two fields with finite mass map to the $i^{\text{th}}$ node
        if $\max\left(\overline{\mathcal{D}}_{1i}^{\max}, \overline{\mathcal{D}}_{2i}^{\max}\right) = 1 \wedge \min\left(\overline{\mathcal{D}}_{1i}, \overline{\mathcal{D}}_{2i}\right) > \overline{\mathcal{D}}^{\min}$ then
            separable = true                                                  ▷ Allow sliding, separation
        else
            separable = false                                                 ▷ Treat as single field
        end if
        $\boldsymbol{q}^{\text{cm}} = \boldsymbol{q}_{1i} + \boldsymbol{q}_{2i}$                              ▷ Contact pair nodal momentum
        $m^{\text{cm}} = m_{1i} + m_{2i}$                                     ▷ Contact pair nodal mass
        $\boldsymbol{v}^{\text{cm}} = \boldsymbol{q}^{\text{cm}}/m^{\text{cm}}$                              ▷ Contact pair nodal velocity
        $\boldsymbol{n}_1^{\text{cm}} = (\boldsymbol{n}_{1i} - \boldsymbol{n}_{2i})/\|\boldsymbol{n}_{1i} - \boldsymbol{n}_{2i}\|$          ▷ Corrected surface normal
        $\boldsymbol{n}_2^{\text{cm}} = -\boldsymbol{n}_1^{\text{cm}}$                                 ▷ Corrected surface normal
        for $\varsigma = 1..2$ do                                             ▷ Loop over all velocity fields
            $\left[\boldsymbol{s}^{\text{cm,A}}, \boldsymbol{s}^{\text{cm,B}}\right] \leftarrow f(\boldsymbol{n}_\varsigma^{\text{cm}})$          ▷ Compute unique orthonormal basis
            if separable then                                                 ▷ Allow sliding & separation
                if $(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}) \cdot \boldsymbol{n}_\varsigma^{\text{cm}} > 0$ then
                    $f^{\text{nor}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{n}_\varsigma^{\text{cm}}$          ▷ Normal force component
                    $f^{\text{tan,A}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{s}^{\text{cm,A}}$          ▷ Tangential force component
                    $f^{\text{tan,B}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{s}^{\text{cm,B}}$          ▷ Tangential force component
                    $f^{\text{tan}} = \|f^{\text{tan,A}}\boldsymbol{s}^{\text{cm,A}} + f^{\text{tan,B}}\boldsymbol{s}^{\text{cm,B}}\|$          ▷ Tangential force magnitude
                    $\boldsymbol{s}^{\text{cm}} = \left(f^{\text{tan,A}}\boldsymbol{s}^{\text{cm,A}} + f^{\text{tan,B}}\boldsymbol{s}^{\text{cm,B}}\right)/f^{\text{tan}}$          ▷ Tangential force direction
                    $\boldsymbol{f}_{\varsigma i}^{\text{ct}} \mathrel{+}= f^{\text{nor}}\boldsymbol{n}_\varsigma^{\text{cm}} + \min\left(\mu|f^{\text{nor}}|, |f^{\text{tan}}|\right) f^{\text{tan}}\boldsymbol{s}^{\text{cm}}$          ▷ Add contribution
                end if
            else                                                              ▷ Treat as single field
                $f^{\text{nor}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{n}_\varsigma^{\text{cm}}$          ▷ Normal force component
                $f^{\text{tan,A}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{s}^{\text{cm,A}}$          ▷ Tangential force component
                $f^{\text{tan,B}} = \frac{m_{\varsigma i}}{\Delta t}\left(\boldsymbol{v}^{cm} - \bar{\boldsymbol{v}}_{\varsigma i}\right) \cdot \boldsymbol{s}^{\text{cm,B}}$          ▷ Tangential force component
                $\boldsymbol{f}_{\varsigma i}^{\text{ct}} \mathrel{+}= f^{\text{nor}}\boldsymbol{n}_\varsigma^{\text{cm}} + f^{\text{tan,A}}\boldsymbol{s}^{\text{cm,A}} + f^{\text{tan,B}}\boldsymbol{s}^{\text{cm,B}}$          ▷ Add contribution
            end if
        end for
    end if
end for
```

2.4. *Self contact*

One promising application for the MPM is mesoscale simulation, where complex microstructure models can be constructed directly from the voxelized data produced by CT scans. MPM is particularly well suited for modeling porous and granular materials, where localized large deformation, self contact, and frictional effects may significantly affect the bulk material response. Another area of emerging interest relates to low-density metamaterials, produced with additive manufacturing and having 3-D multiscale lattice structures. In some cases, these nanotruss structures are able to withstand extreme deformation, which induces frictional self contact under normal operation [53]. Simulating this behavior is a challenging test for traditional numerical methods.

We have already seen that by defining a kernel damage field, the material on either side of a crack can be partitioned into contact pairs. With a slight modification, we can also achieve frictional self-contact with field-gradient partitioning. Using the magnitude of the kernel sum field ($\mathcal{S}(\boldsymbol{x})$ from Equation 19) evaluated at a particle, we can detect surface particles in an initial porous or granular structure. We define, for each particle, a surface flag s_p that is set to 1 for particles at the surface and

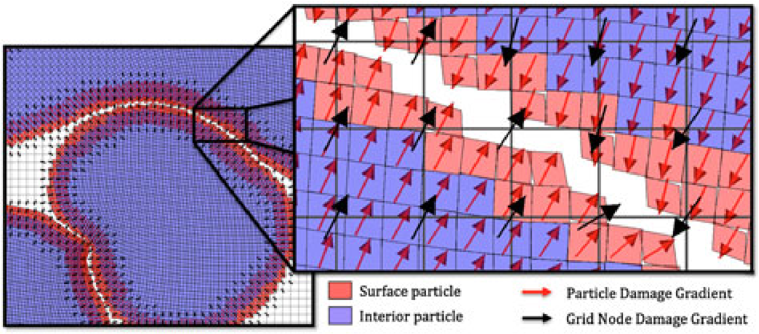

Figure 6. Illustration of the damage-field gradient partitioning method using prescribed surface flags rather than accumulated damage to produce frictional contact at a material interface.

0 for interior particles. Such kernel-based surface detection algorithms are common in mesh-free methods, (cf., [54]). If the grains are initially in contact (but not bonded), it would be necessary to 'paint in' the surface flag at initialization. With this approach, surface particles are readily identified, as shown in Figure 4.

With the surface flag so-defined, the DFG can be constructed with Equation (17), using $\max(s_p, \mathcal{D}_p)$ in place of $\mathcal{D}_p$. In this way, frictional contact and separation will occur between surface particles, even if those particles are undamaged. By using the kernel fields to detect surface particles, the thickness of the surface-flagged region is consistent with the kernel radius in the DFG construction, which seems to improve the surface partitioning in large-deformation frictional contact. An example of this is shown in Figure 6, which shows the direction of the DFG for the particles and grid nodes along a contact surface between two initially separate bodies. This image is generated from the self-contact test case presented in Figure 9.

2.5. *Parallelization*

Massively parallel implementation of the MPM using domain decomposition has been previously demonstrated in the Uintah computational framework to produce nearly linear scaling (both strong and weak) up to 256 K processing cores [55]. In this approach, the background grid is partitioned spatially into subdomains that each reside on a single processor, with the corresponding particles residing on the same processor. MPI communications are used to synchronize grid node quantities at the processor boundaries, and particles are moved between processors as needed. In our GEOS implementation of the MPM, MPI commincations handle domain decomposition, and openMP multithreading has been used to accelerate the sorting algorithm and the particle-to-grid mapping calculations within a single subdomain. In a mixed parallelization implementation, openMP multithreading allows for larger subdomains, which reduces the fraction of computational and memory resources devoted to inter-processor communications.

Construction of the damage field requires that a particle neighbor list be constructed and maintained across processor boundaries. A variety of methods exist for efficient generation of particle neighbor lists, with OctTree data structures [56] being the preferred method for sparse or heterogeneously distributed points in 3-D. If the material points have a fairly uniform distribution in the domain (as may be the case for mesoscale simulations), it can be more efficient to use a simple uniform binning in the sorting algorithm.

Parallelized construction and maintenance of neighbor lists is a well-understood problem, which is required for a variety of meshless particle methods (e.g., SPH and RKPM) as well as for contact detection in traditional Lagrangian finite element. While the cost of a neighbor search can be significant as the particle count increases, the number of particles contained in the computational domain of a single processor is typically small enough that the sorting cost is not prohibitive. It is also not generally necessary to update the neighbor list at each time step. One straightforward approach to determine when re-sorting is needed is as follows. After sorting, the current position for each particle is stored as the 'last-sorted position'. At each subsequent time step, the displacement of each

DOI: 10.1002/nme

particle relative to this last-sorted position is computed. If this displacement exceeds some allowable limit (e.g., a small fraction of the neighbor radius) the neighbor list is updated. Using this methodology for the test cases in this paper, the cost of the neighbor list construction becomes minimal, as described in Section 3.5.

2.6. MPM implementation

The DFG-partitioning method described herein is applicable to a wide variety of MPM implementations (single-point, GIMP, CPDI, etc.). For this work, we consider a 3-D MPM implementation in the massively parallel GEOS [57, 58] computational framework. This code uses convective particle domain interpolation (CPDI) [39] with domain scaling [32]. Scaling is applied to failed particles so that the particle diagonal length does not exceed the grid cell spacing Δx, which prevents non-physical stretching of failed material across the domain. An update-stress-last [40] formulation is used with modifications to account for frictional contact [33]. Unless otherwise indicated, fracture surfaces are partitioned (using the DFG-partitioning method) into contact pairs, so all material separation is physical, as opposed to numerical fracture.

Particles are initialized with a resolution of two particles-per-cell in each direction. Results are visualized with VisIt [59], which renders each particle based on its scaled CPDI integration domain.

2.7. Constitutive model

A key benefit of the MPM relative to Eulerian or ALE methods is that by avoiding advection errors, it becomes possible to accurately use more sophisticated constitutive models for large-deformation solid mechanics. However, for the examples presented herein, we use a relatively simplistic phenomenological model of the constitutive response, so that these examples can be more easily reproduced for comparative benchmarking.

A hyperelastic, compressible neo-Hookean model is used to compute the elastic trial stress [60],

$$\boldsymbol{\sigma}^E = \left(\lambda \frac{\ln J}{J} - \frac{\mu}{J}\right)\mathbf{I} + \frac{\mu}{J}\mathbf{F}\cdot\mathbf{F}^T, \tag{22}$$

where λ is the Lame parameter, μ is the shear modulus, J is the determinant of $\mathbf{F}$, and $\mathbf{I}$ is the identity tensor. The eigenvalues of $\boldsymbol{\sigma}^E$, $\left[\sigma_1^E, \sigma_2^E, \sigma_3^E\right]$, are compared to a maximum principal failure stress, σ_f. If the failure stress is exceeded (e.g., if $\max(\sigma_i^E) > \sigma_f$), the scalar damage measure for the particle is updated.

For verification of the fracture response, it is useful to employ a constitutive model wherein the softening behavior is tied to a directly measureable physical property. To this end, we present several test cases using a Rankine damage model with linear strain-softening as described by Cervera *et al.* [61]. In this model, the failure response depends on the maximum tensile stress relative to the user-defined strength σ_f, with the rate of softening determined by Mode I fracture energy G_f and a characteristic length l^{ch}.

$$\mathcal{D}^{n+1} = (1 + H_s)\left(1 - \frac{\sigma_f}{\max(\sigma_i^E)}\right), \tag{23}$$

where the brittlness factor $\overline{H}_s = \sigma_f^2/2EG_f$ is scaled by a characteristic length to give $H_s = \overline{H}_s l^{\mathrm{ch}}/(1-\overline{H}_s l^{\mathrm{ch}})$. This provides a tie to classical fracture mechanics and allows us to compare the dynamic fracture response of our method with that of other traditional approaches using the same constitutive model parameters (Section 3.2) and to verify that our method produces dissipation in crack formation that is consistent with the specified fracture energy (Section 3.3).

Linear elastic fracture mechanics can adequately describe many simple systems but has limited value for many practical systems where nonlinearity, heterogenieties, or complex topologies dominate the response. In many cases, empirical damage models offer a practical alternative.

For example, linear elastic fracture mechanics predicts that the crack propagation velocity should approach the Rayleigh wave speed in the material, but in many cases, the observed critical fracture propagation speed is well below this theoretical value (commonly on the order of $0.4v_r$ [62]). This discrepency has been attributed to the formation of microfractures, which increase the energy release per unit length of the macroscale crack feature. For continuum modeling at the macroscale where it is not practical to resolve the microscale fracture formation, we can achieve the desired wave speed through introduction of a 'time-to-failure' model, where the time-to-failure is defined as the transit time for a crack to propagate across a particle, $\tau = c_f/\ell$, where $c_f$is the crack velocity and ℓ is the characteristic length. In our implementation, c_f is a constitutive model parameter, and the characteristic length is $l^{\text{ch}} = \sqrt{\Delta x^2 + \Delta y^2}$ in 2-D or $l^{\text{ch}} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ in 3-D. This approach should give a predictable value of the crack velocity, even as the mesh is refined. As such, a comparison of the crack speed observed in the simulation with that prescribed in the input parameters provides a verification test for the method.

For the time-to-failure model, the scalar damage is incremented as

$$\mathcal{D}^{n+1} = \min\left[1, \mathcal{D}^n + \frac{\Delta t}{\tau}\right]. \tag{24}$$

When using a smeared (continuum) damage model to simulate brittle failure, it is necessary to regularize the solution to avoid mesh dependence, for example, by introducing spatial heterogeneity to describe the aleatory uncertainty in the material strength (cf., [47]). A classical approach is to define the particle strength, σ_f, to be both spatially variable and size dependent according to a Weibull distribution [63, 64],

$$\frac{\sigma_f}{\overline{\sigma_f}} = \left(\frac{\overline{V}}{V}\frac{\ln\mathcal{R}}{\ln 1/2}\right)^{1/m}, \tag{25}$$

where V is the particle volume, $\overline{\sigma_f}$ and $\overline{V}$ are the reference failure stress and corresponding reference volume, m is the Weibull modulus, and $\mathcal{R}$ is a uniformly distributed random number $0 < \mathcal{R} < 1$. The 2-D computational examples in this work are all plane strain, but it is necessary to specify a material thickness to define the particle volume. While there is some evidence that a Weibull distribution may not be physically representative of flaw distributions in many engineering materials (cf., [65]), the need for physically motivated spatial heterogeneity is clear.

In Section 3.4, we present results of a fragmentation simulation using the time-to-failure model with spatial heterogeneity. This is not meant to describe a particular material but is intended simply as a demonstration that our approach can produce mesh-insensitive results for practical engineering models.

For either the time-to-failure or the Rankine strain-softening models, the updated value for the damage is used to scale back any tensile principal stresses (cf., [61]),

$$\sigma_i^+ = \begin{cases} (1-\mathcal{D})\,\sigma_i^E & : \sigma_i^E > 0 \\ \sigma_i^E & : \sigma_i^E \leqslant 0 \end{cases}. \tag{26}$$

The scaled tensile principal stresses and the unmodified compressive principal stresses are used with the original eigenvectors $(\boldsymbol{v}_i)$ to reconstruct the scaled stress tensor,

$$\boldsymbol{\sigma} = \Sigma_{i=1}^3 \sigma_i^+ \boldsymbol{v}_i \otimes \boldsymbol{v}_i. \tag{27}$$

Both the empirical time-to-failure approach and the strain-softening fracture energy approach are strictly suitable only for Mode I dominated dynamic fracture, but they are sufficient for the demonstration cases in this manuscript. More sophisticated continuum damage models would be needed to apply the DFG-partitioning method to a broader class of problems.

3. RESULTS AND DISCUSSION

We apply the DFG-partitioning method to a variety of dynamic fracture and contact problems to provide confidence in the method and to demonstrate its utility. For these problems, we have used highly simplified constitutive models (Section 2.7), which are sufficient to illustrate key effects in brittle failure, but which are generally not adequate to fully describe the complex rate-dependent fracture response of engineering materials. Nonetheless, the following test cases demonstrate that the DFG-partitioning method significantly enhances the ability of the MPM to represent brittle failure with a continuum damage model.

3.1. Verification of self-contact response

The value of the self-contact capability is that it becomes possible to achieve frictional contact between surface pairs that cannot be easily identified at the initialization of the problem, (e.g., certain porous structures) or when the number of potentially contacting surfaces is so large that it becomes infeasible to prescribe a sufficient number of contact groups (e.g., polydisperse granular materials in 3-D). However to demonstrate the self-contact algorithm, we apply the method to problems for which the correct frictional contact solution can be easily obtained with an initialization of multiple material velocity fields and then show that the same solution is produced from a single material field with kernel-based surface detection.

The first example is a 2-D plane-strain compaction of a compliant hyperelastic billet between stiff steel platens, modeled in a single quadrant as shown in Figure 7. A constant compressive normal force is applied to the platen, causing the billet to compress and expand outward. Parameters for the simulation are given in Table I. Figure 8 shows the results of a SF solution (no-slip contact), a traditional multi-field frictional-contact solution (i.e., the 'correct' solution), and a SF solution with

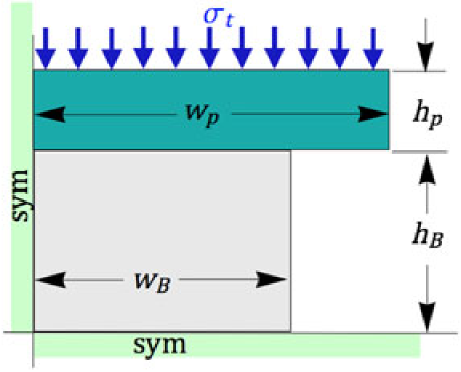

Figure 7. Schematic of the 2-D plane-strain billet self-contact problem.

Table I. Input parameters for the billet self-contact simulation.

Parameter	Description	Value
w_p	Platen width	1.2 mm
h_p	Platen height	0.4 mm
w_b	Billet width	1.0 mm
h_b	Billet height	0.6 mm
σ_t	Applied compressive traction	200 MPa
K_b	Bulk modulus (billet)	158.333 GPa
G_b	Shear modulus (billet)	73.077 GPa
ρ_b	Density (billet)	2.3 g cm^{-3}
K_p	Bulk modulus (platen)	140 GPa
G_p	Shear modulus (platen)	60 GPa
ρ_p	Density (platen)	8 g cm^{-3}
μ	Coefficient of friction	0.04

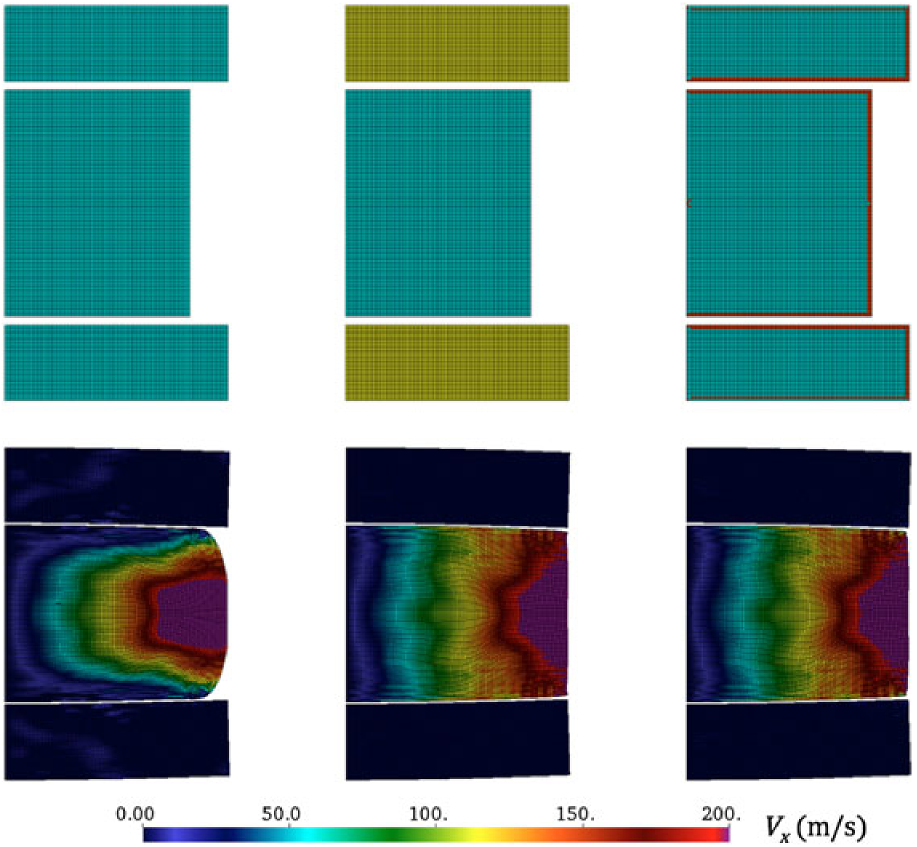

Figure 8. Frictional contact in plane-strain compaction of an elastic billet showing contours of horizontal velocity. Left: single-field solution produces no-slip contact. Center: multiple velocity fields (billet and platen) producing frictional contact. Right: single material field with surface detection at $t = 0$ using density gradient. Field-gradient partitioning produces frictional contact very similar to the multi-material solution.

surface detection and field-gradient partitioning. The field-gradient partitioning method produces a result nearly identical to that of the multi-field solution. This verifies that the DFG-partitioning method correctly models self-contact for problems involving large sliding displacement at a contact surface.

A more challenging problem is illustrated in Figure 9, which shows plane-strain large deformation of a low-stiffness hyperelastic material, for which the contact surfaces move and distort. A distributed horizontal load acts on a wedge of material that moves to the right until it contacts a disk, pushing it to contact a square block. The problem is designed to produce a dramatic difference between the no-slip and frictional results. Again, we find that with surface detection and frictional self-contact, it is possible to reproduce the frictional response, without the need to initially define distinct regions of contacting material.

These examples not only demonstrate the capability of the method to achieve frictional self-contact without *a priori* identification of the contacting regions of material but also support the assertion that the method naturally models frictional contact of fracture surfaces during post-fracture deformation, even if the solid material is undergoing large (potentially inelastic) deformation. This ‘granular flow’ phase occurs in fragmentation and comminution processes and can be exceedingly difficult to model with other numerical methods.

3.2. *Dynamic fracture response*

A widely used benchmark for simulating dynamic fracture is based on experimental results by Kalthoff and Winkler [66]. In this test, a notched target is impacted by a bar with initial velocity

DOI: 10.1002/nme

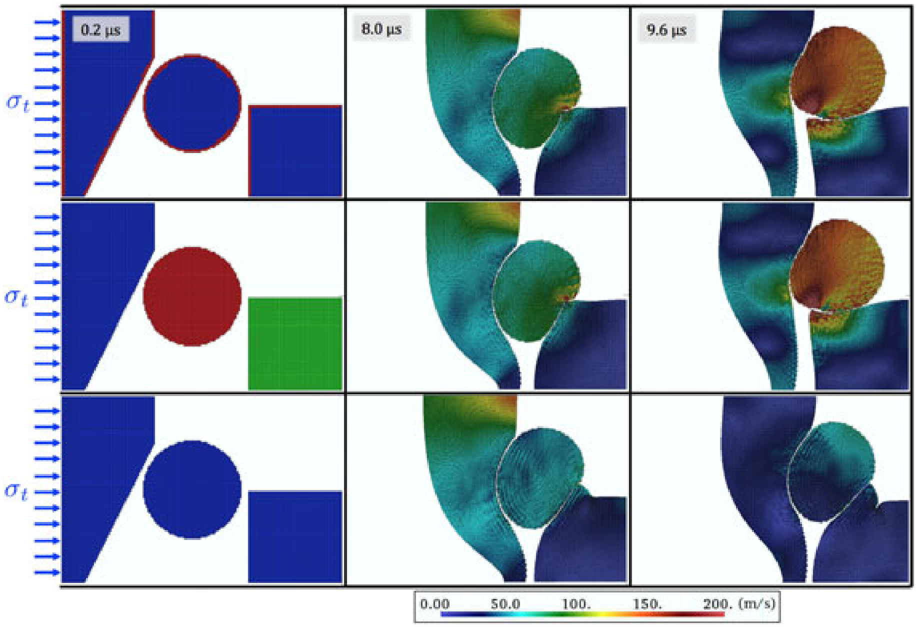

Figure 9. Large-deformation plane-strain frictional contact example showing the initial configuration left and contours of speed (center, right). Bottom: single-field solution produces no-slip contact. Center: multiple material fields are defined *a priori* producing frictional contact. Top: single material field with density-gradient surface detection at $t = 0$. Field-gradient partitioning self-contact produces frictional contact response very similar to the multi-material solution.

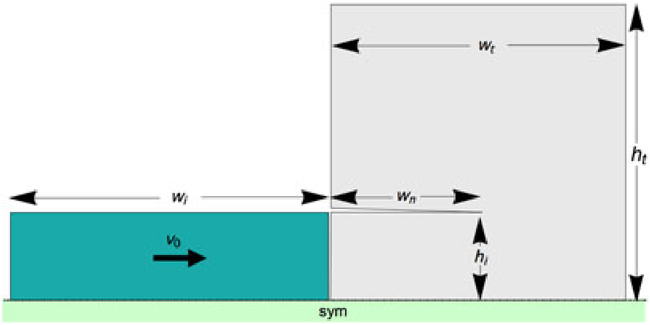

Figure 10. Schematic of the 2-D plane-strain dynamic shear fracture test problem. The experimental results by Kalthoff and Winkler show a crack propagating from the notch at an angle of 70° [66].

v_0, as shown in Figure 10. The impact causes a crack to propagate from the notch at a 70° angle, and both the crack propagation direction and crack velocity are key validation metrics. This test problem was used by Song *et al.* [48] to evaluate the dynamic fracture response of various finite element methods. They found that the XFEM could produce the correct qualitative response but that the computed crack direction differed by 10° from the experimental results; they also found element-deletion and inter-element methods produced unacceptable mesh dependence.

We model the impact and fracture using the MPM with with a strain-softening constitutive models, using fracture energy $\mathcal{G}_f = 22.13\,\text{N/mm}$ and $\sigma_f = 570\,\text{MPa}$. These material parameters were selected to match those used by Song *et al.* [48] in their XFEM analysis. The mesh resolution over the target domain is 100 × 100 grid cells. The input parameters are given in Table II. The results are given in Figure 11d–e, showing the velocity field and the evolution of damage for the DFG-partitioning method and for a traditional SF solution. The SF and DFG-partitioning solutions produce a single crack with final angle of 60.8° and 59.1° , respectively, which matches closely with the 60° angle reported for the XFEM results; all these results underpredict the fracture angle by

Table II. Input parameters for the Kalthoff–Winkler fracture problem.

Parameter	Description	Value
w_t	Target width	10 mm
h_t	Target height	100 mm
t	Specimen thickness	2 mm
w_n	Notch width	50 mm
h_i	Impactor height	25 mm
w_i	Impactor width	100 mm
v_0	Impactor velocity	$33\ \mathrm{m\,s^{-1}}$
K	Target/impactor bulk modulus	158.333 GPa
G	Target/impactor shear modulus	73.0769 GPa
ρ	Target/impactor density	$8\ \mathrm{g/cm^3}$
σ_f	Target tensile strength	570 MPa
G_f	Target fracture energy	$22.3\ \mathrm{mJ\,mm^{-2}}$
l^{ch}	Characteristic length	1.414 mm
$\overline{\mathcal{D}}^{\min}$	Separability threshold	0.25

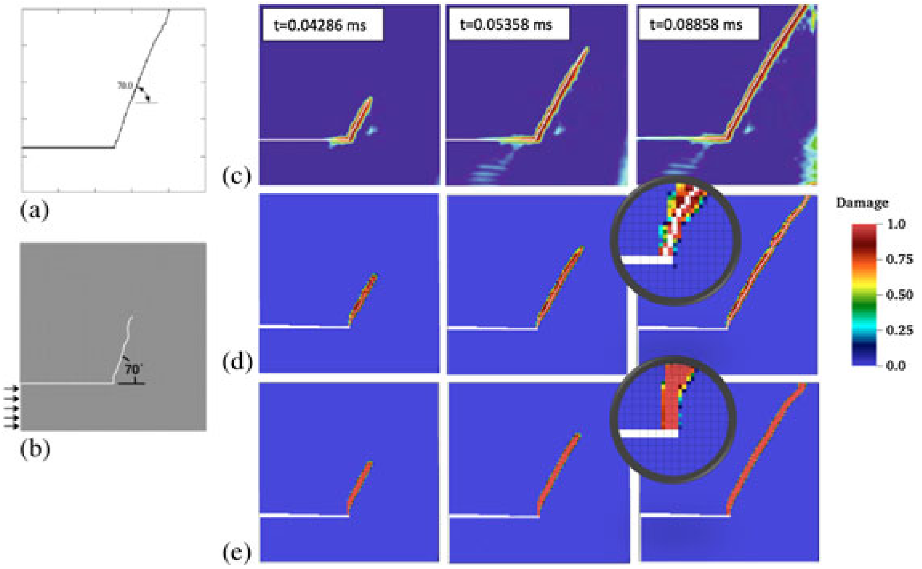

Figure 11. Kalthoff–Winkler dynamic fracture experiment showing (a) the experimentally observed initial crack direction [68], (b) crack geometry predictions using MPM with explicit cracks from Nairn *et al.* [69], (c) crack propagation predictions using XFEM from Song *et al.*[48], (d) crack propagation predictions from the present work using DFG partitioning with crack-tip detail inset, and (e) crack propagation predictions from the present work using a single field solution with crack tip detail inset.

10° . Examining the detail of the crack initialization in the SF and DFG-partitioning MPM solutions, there is some evidence that the SF solution favors a grid-aligned crack, but the effect is small.

In simulating this experiment using XFEM with a loss-of-hyperbolicity crack propagation criterion (using comparable constitutive model parameters to our work), Belytchko *et al.* [67] reported a crack speed in the range of approximately $2.0 - 2.2\,\mathrm{km\,s^{-1}}$ for $40\,\mathrm{us} < t < 50\,\mathrm{us}$, whereas Song's XFEM simulation results give approximately $2.8\,\mathrm{km\,s^{-1}}$. Our DFG-partitioning simulations agree well with Belytchko's result, with a crack speed of $2.0\,\mathrm{km\,s^{-1}}$, whereas the SF simulation gives a somewhat lower value at $1.7\,\mathrm{km\,s^{-1}}$.

We have observed that even when using Lagrangian FEM, the fracture behavior in the Kalthoff–Winkler simulations is highly sensitive to small variations in constitutive model parameters or support conditions; this is particularly evident for the response as the fracture approaches the top boundary. Nonetheless, it is reassuring that the results obtained using MPM with DFG-partitioning are generally comparable with published solutions with the XFEM method, without requiring any parameter tuning. While the SF MPM solution gives a similar crack geometry, it may slightly underpredict the crack speed, perhaps due to spurious velocity-field smearing dissipation at the crack surface.

3.3. Energy dissipation and mesh dependence in a modified charpy impact specimen

The previous section demonstrates that the DFG-partitioning method performs comparably with accepted approaches for simulating fracture in a widely adopted validation test. To provide further verification, we investigate the energy dissipation in crack formation and compare this with the expected value for the specified fracture energy G_f input to the constitutive model.

While it is possible to compute the plastic work dissipated in a continuum damage model, this does not account for 'phantom' dissipation due to the velocity-field smearing error in a SF solution. Thus, to better quantify the ability of the DFG-partitioning method to accurately predict fracture energy, we perform a full-system energy balance on a simplified Charpy impact test. The test comprises an elastic 2-D disk impacting a notched fracture specimen as shown in Figure 12. With this method, we investigate the effects of the constitutive model length scale from Section 2.7, the separability criterion from Section 2.3, and the convergence with mesh refinement.

The MPM conserves mass and linear momentum by construction, but only approximately conserves energy. There are a variety of integration schemes for the MPM, but in general, the less-dissipative methods sacrifice stability. Thus, even with a fully elastic collision, we expect some error in the total energy in the system. To estimate this error, we first compute the response using a hyperelastic constitutive model for both striker and target and track the fluctuations in total energy in the system. We find that for $\Delta x = 1$ mm, while there are significant errors in energy during the initial impact, the total energy in the final state oscillates around a mean value that is 2.5% lower than the initial kinetic energy of the striker. The strain energy is much lower in the brittle fracture simulations, where the kinetic energy approximately equals the total energy in the system. Nonetheless, there is some uncertainty in evaluating the energy dissipation due to fracture formation, as some of this energy may be due to numerical dissipation in the general MPM solution, not the fracture response.

Figure 13 shows the energy change in the system during impact for SF and DFG solutions at three resolutions. The black dashed line indicates the expected final state based on the specified fracture

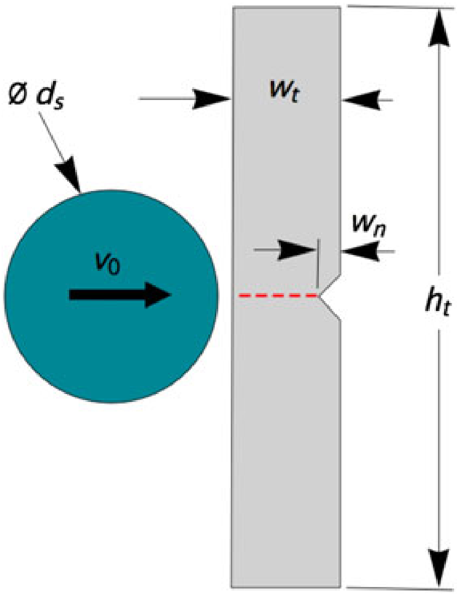

Figure 12. Schematic of the modified Charpy impact test problem, which produces a single fracture along the path indicated by the red dashed line.

Table III. Input parameters for the Charpy impact fracture problem.

Parameter	Description	Value
w_t	Target width	10 mm
h_t	Target height	55 mm
t	Domain thickness	2 mm
w_n	Notch width	2 mm
d_s	Striker diameter	20 mm
v_0	Striker velocity	20 m s^{-}1
K	Bulk modulus	158.333 GPa
G	Shear modulus	73.0769 GPa
ρ	Density	8.0 g cm^3
σ_f	Target tensile strength	1.0 GPa
G_f	Target fracture energy	22.3 mJ mm^{-2}
l^{ch}	Characteristic length	1.414 mm
$\overline{\mathcal{D}}^{min}$	Separability threshold	0.25 mm

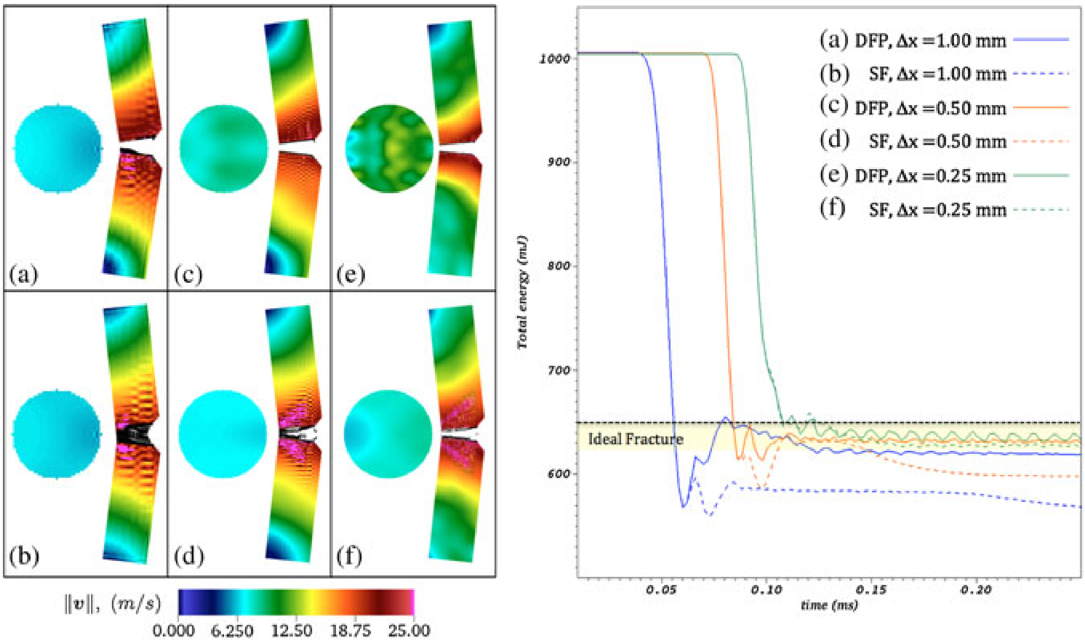

Figure 13. Results showing the energy dissipation for the simplified Charpy impact problem with single field (SF) and damage-field gradient partitioning (DFG). The DFG method produces the correct fracture energy with less mesh dependence.

energy release rate ($\Delta E^{\text{tot}} = G_f A_f = 354.1$ mJ), and the yellow band indicates an estimate of the additional numerical dissipation based on analysis of comparable elastic impacts. At each resolution, the initial gap between the striker and target is slightly different, causing an offset response for each resolution. The DFG results produce approximately the correct fracture energy at all resolutions, with noise due to undamped high-frequency oscillations becoming more evident as the mesh is refined. The SF solution shows a consistently lower post-fracture energy at all resolutions. This is attributed to the velocity-field smearing error, which produces additional dissipation at the fracture surface, as well as additional 'noise' in the velocity field.

While the high-resolution SF and DFG-partitioning results are similar, there is significant error in the low-resolution SF response. When modeling complex fracture systems such as the disk crush examples in the following section, it is not always feasible to apply a high resolution for each fracture in the domain, so the ability of a method to perform reasonably at low resolutions is important.

DOI: 10.1002/nme

The separability criterion introduced in Section 2.3 affects the nature of localization, such that for a low threshold value of $\overline{\mathcal{D}}^{\min}$, the crack becomes a single row of failed particles (or even a discontinuous row of failed material, with one failed particle per grid cell), and for a higher threshold, the localization becomes a row of failed material on each side of the crack, or a full grid cell of failed material on each side of the crack. The arbitrariness of this separability threshold seems undesirable, but it is important to understand that a conventional SF MPM solution also prescribes an arbitrary localization domain (that of one grid cell or more, depending on the relative alignment and orientation of the particles and background grid [32]).

The characteristic length in the constitutive model also affects the nature of the localization response and should ideally be set equal to the width of a smeared crack in the continuum field. The effect of varying the characteristic length and separability threshold is shown in Figure 14. Both of these parameters affect the energy dissipated in fracture formation and should be selected to be consistent with the assumptions of the constitutive model. In this case, selecting a value of $l_{\text{ch}} = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{2}$mm and $\overline{\mathcal{D}}^{\min} = 0.25$ gives reasonable results as was shown Figure 13. The characteristic length determines the softening response and thus has a strong impact on the fracture energy, regardless of whether a SF or DFG-partitioning solution is used. Decreasing the value of $\overline{\mathcal{D}}^{\min}$ has no perceptible effect, but there is a change in dissipation as $\overline{\mathcal{D}}^{\min}$ is increased towards 1. Generally, this effect is less significant than that of varying the characteristic length. With these parameters, the analyst is afforded a degree of control over the localization response, which can be tuned to more precisely reproduce a fracture energy release rate consistent with the constitutive modeling assumptions.

The results of the Charpy impact and Kalthoff–Winkler test problems demonstrate that for these problems the DFG-partitioning method can produce a dynamic fracture response consistent with the assumptions of the constitutive model and comparable with that of widely accepted approaches for simulating fracture. In the following section, we demonstrate that using this approach along with spatially variable material strength can produce reasonably mesh-independent results in simulating a complex fragmentation/comminution test case.

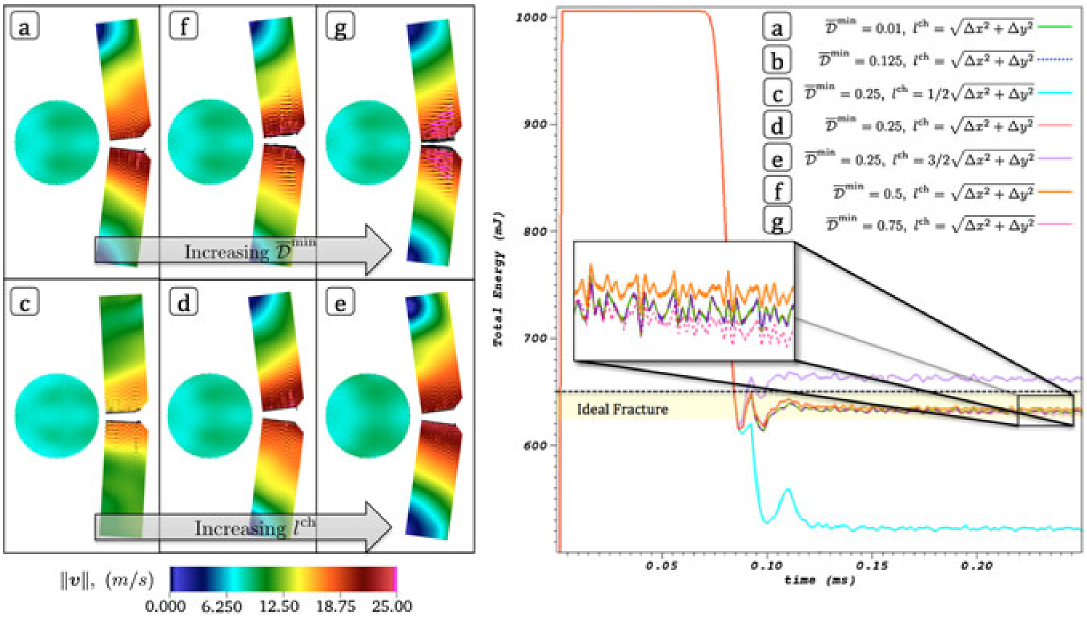

Figure 14. Results of the simplified Charpy impact simulation using damage field partitioning with $\Delta x = 0.5$ mm with various values of $\overline{\mathcal{D}}^{\min}$ and l^{ch}. Left: contours of velocity magnitude with damage overlaid in black. Right: energy dissipation for each test with the inset frame shows the detail of the results for $l_{\text{ch}} = \sqrt{\Delta x^2 + \Delta y^2}$ at various values of $\overline{\mathcal{D}}^{\min}$; results for $\overline{\mathcal{D}}^{\min} = 0.01, 0.125$, and 0.25 are coincident.

DOI: 10.1002/nme

3.4. Fragmentation and compaction response

In presenting methods to model fragmentation, it is common to simulate problems such as an expanding ring, comprising mode I fracture, and without any post-fracture contact. However, in assessing the suitability of the DFG-partitioning method for simulating comminution or the interactions of fragments after failure, it is necessary to investigate the response of the method in a problem entailing mixed-mode fracture, crack interactions, large-displacement frictional contact, and large material deformation. To this end, we simulate the initial fracture and post-fracture response in a dynamic, 2-D plane-strain, indirect-tension ‘Brazilian’ test, and compare the response of the DFG-partitioning method with that with traditional SF MPM.

A brittle disk is compressed between two elastic platens under constant-stress loading, as illustrated in Figure 15. The platens are defined to have a density and stiffness approximating steel, and there is frictional contact between the disk and platens. The disk is modeled with an empirical time-to-failure model, and Weibull-distributed strength is assigned to the disk to produce realistic brittle failure patterns. The model parameters are given in Table IV.

After an initial elastic preload, the material fails near the contact points, consistent with the location of the maximum principal stress predicted by Hertzian contact theory [70]. For simulating brittle failure, the crack path under a given stress field is dictated by the distribution of weak material points in the high-stress region. In our work, we have assumed a Weibull distribution of strength, which is selected so that the probability of failure initiating in some domain at a given stress is constant, regardless of the discretization resolution. The failure progression and crack patterns are expected to vary between statistical realizations of the random perturbations in strength, but generally, the size and velocity of the fragments should be similar between realizations.

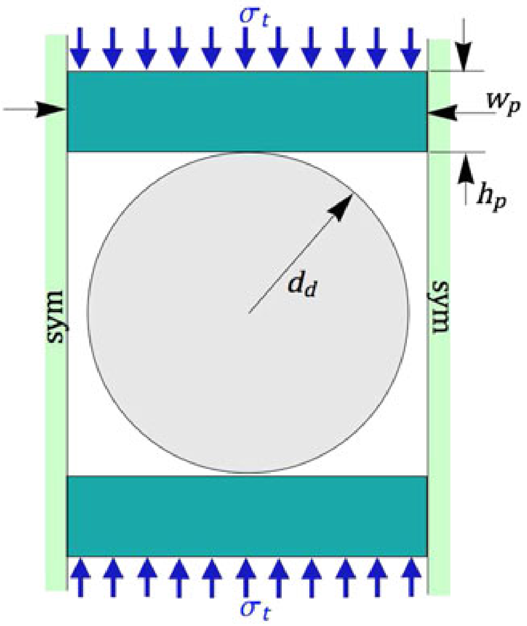

Figure 15. Schematic of the 2-D plane-strain dynamic Brazilian simulation.

DOI: 10.1002/nme

Table IV. Input parameters for the dynamic fragmentation/compaction simulation.

Parameter	Description	Value
w_p	Platen width	12 mm
h_p	Platen height	2 mm
t	Domain thickness	2 mm
d_d	Disk diameter	10 mm
σ_t	Applied compressive traction	650 MPa
$\sigma_{ref,d}$	Reference tensile strength (disk)	140 MPa
$V_{ref,d}$	Reference sample volume (disk)	8 cm^3
m_d	Weibull modulus (disk)	6.0
K_d	Bulk modulus (disk)	158.333 GPa
G_d	Shear modulus (disk)	73.077 GPa
ρ_d	Density (disk)	2.3 g cm^{-3}
$c_{\mathrm{f,d}}$	Crack speed (disk)	2 km s^{-1}
K_p	Bulk modulus (platen)	260 GPa
G_p	Shear modulus (platen)	180 GPa
ρ_p	Density (platen)	8 g cm^{-3}

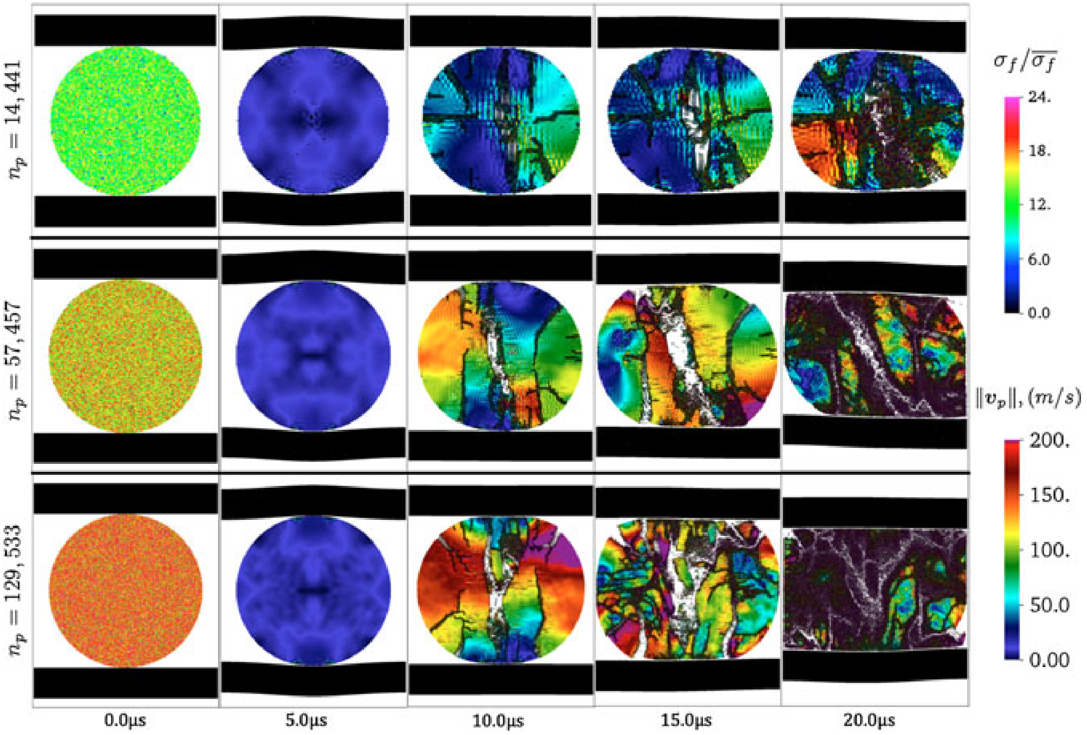

Figure 16. Comparison of dynamic Brazilian simulations of a brittle material using a *traditional single-field solution* at three resolutions. The left column shows the Weibull-distributed relative particle strength ($\sigma_f/\overline{\sigma_f}$) in the initial condition; the other frames show particles colored by velocity magnitude ($\|v_p\|$) with damage overlaid in black. The velocity-field smearing error masks the effect of the stress concentration at the initial failure point, suppresses crack formation, and causes nonphysical dissipation in granular flow. This dissipation reduces with refinement, producing a pronounced mesh dependence in the fragment size and velocity.

A common failure progression is as follows. One or more cracks grow from the initial failure point and propagate vertically through the disk. The ends of the disk are pulverized and an internal crack nucleates within a large central fragment. This crack grows and branches, while other cracks grow inward from the outer surface of the disk. Eventually, the entire disk is fragmented and begins to flow outward, with fragments sliding past one another. When the fragments impact the side walls, they are pulverized, and the debris starts to fill the volume and become compressed.

This comminution process at three resolutions ($\Delta x = 0.181$,0.094, and 0.063mm corresponding to $n_p = 14441$, 57457, and 129553 particles) is shown in Figure 16 and Figure 17 using a SF and DFG-partitioning method, respectively. At each resolution, the statistical strength perturbations are the same for the SF and DFG results, but there is no correlation between the strength perturbations between scales. Thus, while it is desirable that the results are qualitatively similar at all resolutions, the specific crack patterns should differ.

The results show that the SF solution is generally stiffer, delaying the onset of crack initiation and restricting the formation of new cracks and crack branches. This is because the stress concentration effect from a single failed particle (potentially the result of an initial 'flaw' represented by a weak particle in the heterogeneous material definition) is suppressed by the single velocity field over the cell containing that particle. Note that the time scales are different in the two figures, to account for the delayed onset of failure in the SF solution. Additionally, grid-aligned crack propagation is favored, which is most evident at low resolutions, and damage localizes to a much larger region rather than forming discrete cracks. In some materials, a dispersed damage solution may be correct, but such behavior should be a result of the constitutive model response, not an artifact of the numerical solution. The maximum principal stress failure criterion used in these simulations should produce brittle failure patterns with sharp localization, as is observed in DFG results.

The velocity-field smearing error in the SF solution restricts crack propagation and causes additional dissipation as the failure region propagates. This results in increased fragment size and decreased fragment velocity. The additional dissipation required to spread the damage across a grid cell decreases with mesh refinement, so this error is reduced with refinement as in the Charpy

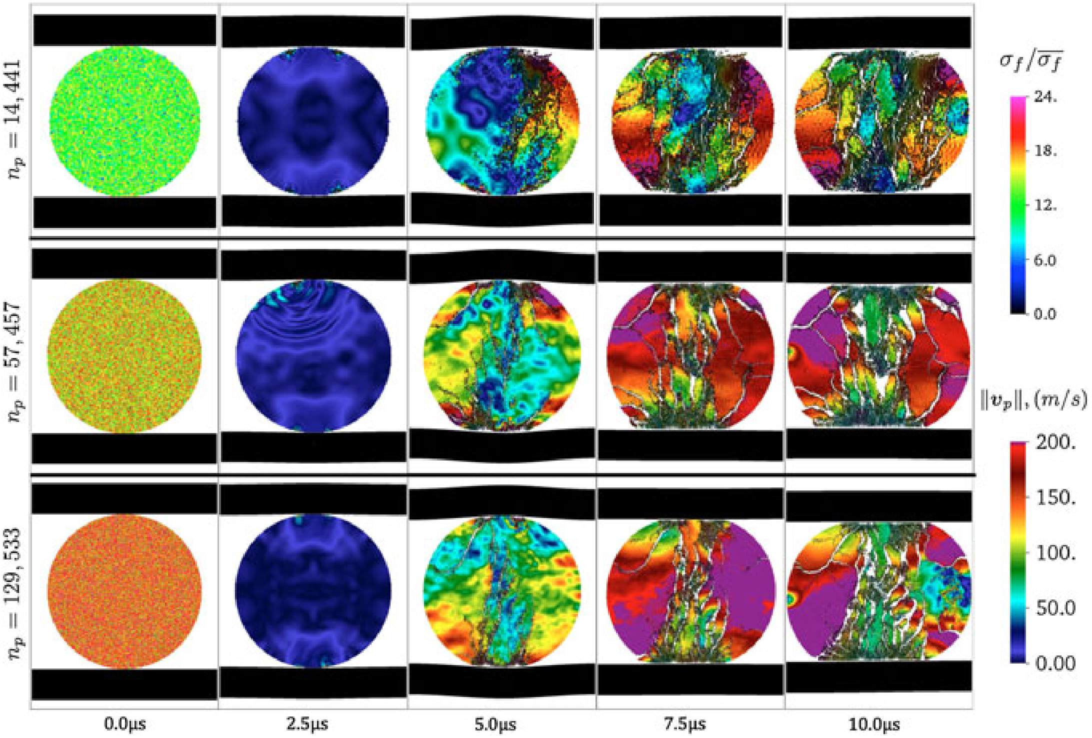

Figure 17. Comparison of dynamic Brazilian simulations of a brittle material using *damage-field gradient partitioning* at three resolutions. The left column shows the Weibull-distributed relative particle strength ($\sigma_f/\overline{\sigma_f}$) in the initial condition; the other frames show particles colored by velocity magnitude ($\|v_p\|$) with damage overlaid in black. damage-field gradient-partitioning produces clean fractures and allows frictional sliding of fragments. Fragment sizes and velocities are similar at all resolutions with small differences being attributed to the different strength realizations at each resolution.

DOI: 10.1002/nme

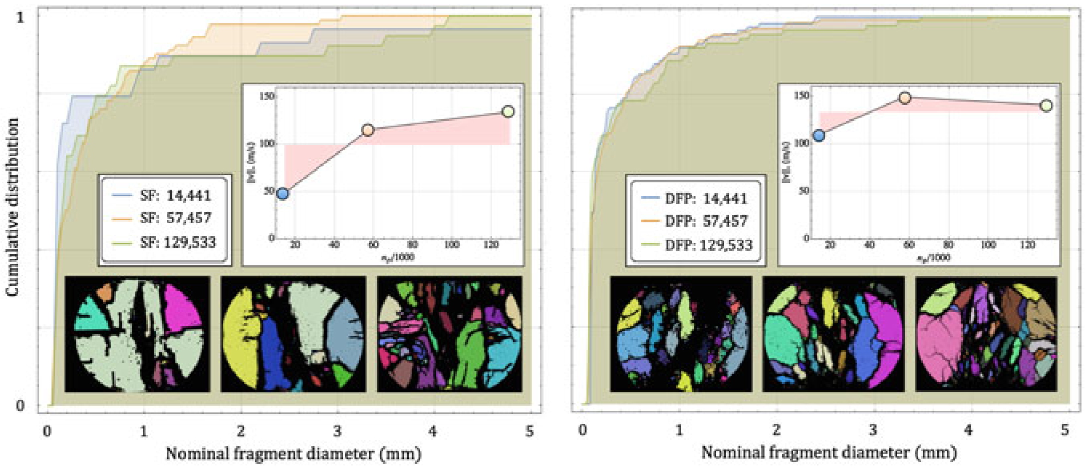

Figure 18. Fragment size distributions for a single-field (SF) solution (left), and damage-field gradient (DFG) partitioning (right) at three levels of spatial refinement. The inset plots shows the average materal velocity magnitude for each result. The DFG approach significantly reduces the mesh dependence in both the velocity and the particle-size distribution of the fragmented disk.

impact example. However, here, this causes a dramatic dependence of the fragmentation response on mesh resolution.

To quantify the mesh resolution effect, we use a connected components algorithm (cf., [71]) to identify fragments of undamaged material and then compute the cummulative particle-size distribution for each simulation. The minimum fragment size resolved in this analysis is approximately 50 μs. To account for the diffence in failure progression rate for the two methods, we analyze the particle size distributions at $t = 10\,\mu\text{s}$ for the DFG-partitioning simulation and at $t = 15\,\mu\text{s}$ for the SF simulation, for which the displacement fields are comparable. The results are shown in Figure 18. The SF solution shows significant changes in fragment sizes with refinement, whereas for the DFG-partitioning result, the fragment sizes and velocities are similar at all resolutions. The mean material velocity magnitude was also computed for each state, which is plotted in the inset frame in each figure. While there is some variability, the DFG-partitioning solution produces a much more consistent velocity field across resolutions.

To fully capture the physical response of fragmentation, brittle materials will likely require a much more sophisticated constitutive model than the simple forms used for this example, and any such development would require significant experimental validation. Nonetheless, these results demonstrates the potential of the DFG-partitioning method to investigate fracture phenomena involving complex crack topologies, branching, frictional 'self'-contact, multiple materials, and large deformation.

3.5. *Computational efficiency*

The field-gradient partitioning method introduces the additional computational expense of: (i) computing and maintaining the neighbor list needed to evaluate the kernel damage field; (ii) evaluating the DFG at the particle; and (iii) additional calculations for the node-wise sorting of material into contact surfaces.

While avoiding the need to compute a neighbor list (thus maintaining linear scaling) is frequently cited (cf., [72]) as one of the key benefits of MPM relative to meshless particle methods, efficient sorting algorithms (e.g., an OctTree for a 3-D domain) reduce the sorting cost significantly and the additional computational expense of the sorting step is only a significant factor when the number of particles in a single computational subdomain becomes very large. The cost of sorting can be further reduced by using the particle displacement relative to the last-sorted position to determine whether re-sorting is needed. In our DFG-partitioning implementation, we found that the cost of sorting was less than 0.1% of the total runtime, even for the largest problems, and that evaluation

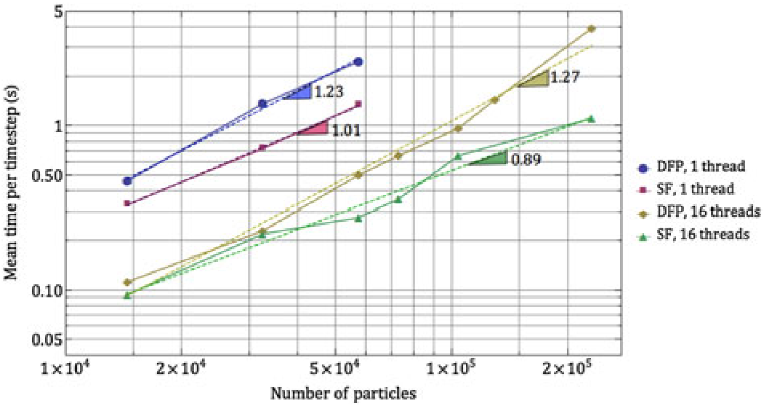

Figure 19. Average runtime for a single time step in a plane-strain dynamic Brazilian simulation versus number of particles for single-thread and 16-thread solutions with single-field (SF) and damage-field gradient (DFG)-partitioning solutions. Slopes of the dashed fitting lines indicate approximate scaling efficiency for each method.

of the DFG at the particles accounted for the majority of the additional computational expense with the DFG-partitioning method. Calculating the DFG at a particle requires a loop over the particle's neighbors. Because we have defined the neighbor radius relative to the grid cell size, and the number of neighbors per particle is unchanged with refinement, it is expected that the expense of this step would scale linearly with n_p.

Figure 19 compares the computational expense per time step of the DFG-partitioning solution with a SF solution for the 2-D plane-strain dynamic Brazilian simulations presented in Section 3.4. Each simulation is run on either a single core or using OpenMP multithreading across a single node comprising two 8-core 2.6 GHz Intel Xeon E5-2670 processors. The simulations are run varying the grid resolution, but with the particles-per-cell fixed at $2 \times 2 \times 1$. The mean computational time per time step is computed as the average over the entire problem runtime, (3,000–18,000 steps, depending on resolution). At lower resolutions ($n_p \leqslant 32,000$) using 16 threads, the multithreading capacity is not fully utilized, which masks the added computational expense of the DFG partitioning. For moderate problem sizes ($32,000 < n_p < 150,000$), the DFG partitioning is shown to increase computational expense by 30–50%, relative to a SF solution. For very large problems, ($n_p > 150,000$), the added expense of the DFG partitioning appears to increase more rapidly; however, even for the largest problem size, timing of individual blocks of code indicate that the expense of the DFG calculations is only 30% of the total expense. This suggests that the observed increase in total time is due to some memory access inefficiencies in the implementation (which may be eliminated with further development), and not due to an inherent algorithmic limitation of the DFG-partitioning approach.

Using domain decomposition with MPI parallelization, the MPM has been used to simulate domains with as many 2.65×10^5 spatial partitions [55]. In such massively parallel implementations, a typical domain for a single processor node is generally not larger than $16 \times 16 \times 16$ grid cells corresponding to 3.2×10^4 particles. While the ideal partitioning scheme is highly dependent on the system architecture, the results in Figure 19 suggest that the additional computational expense of the DFG-partitioning method is tractable for the particles-per-processor-node typical of a massively parallel MPI implementation.

4. SUMMARY AND CONCLUSION

We have demonstrated a new approach for automatic treatment of strong and weak discontinuities in the MPM through node-wise partitioning of material into contact pairs based on the gradient of a scalar field. The scalar field is constructed from particle data using kernel methods. This new approach addresses fundamental limitations of the MPM with regard to localization and frictional contact.

When the scalar field is constructed from surface-flagged particles, the field-gradient partitioning allows frictional contact between surfaces of a body that was initially described by a single velocity field. We have presented results verifying that our method produces the correct frictional response between severely deformed bodies by comparison with the established multi-field approach for contact in MPM. This gives confidence that the post-failure frictional response is accurate in the more complex comminution simulations.

Using the evolving particle damage to generate the scalar field allows for DFG partitioning, which naturally allows fracture propagation, branching, and coalescence in complex 3-D geometries. To establish confidence in the method, we have reproduced experimental results for two simple dynamic fracture test case and have demonstrated that: (i) the method produces comparable results with an XFEM solution using the same constitutive model parameters and without any parameter tuning; and (ii) that the energy dissipation in crack formation is consistent with the fracture energy constitutive model parameter specified in the simulations.

We applied the DFG-partitioning method to simulate comminution in a brittle solid, a challenging problem comprising complex crack interactions, large deformation, large displacement at fracture surfaces, and post-failure frictional contact. For this problem, a traditional SF MPM solution shows significant mesh dependence and requires nonphysical smearing of the damage region to allow frictional sliding and separation at contact surfaces. In contrast, the new DFG-partitioning approach produces sharp fracture features, allows frictional sliding at fracture surfaces, and produces a relatively mesh-independent distribution of fragment sizes and velocities in the simulation.

While the examples in the previous section were all 2-D plane strain, the DFG-partitioning method is inherently 3-D, as is its implementation in the GEOS computational framework. Figure 20 gives an example of a 3-D problem showing a steel sphere impacting a thick ceramic disk. The disk fractures and the fragments separate in subsequent deformation. This result is included simply to demonstrate that the method is not limited to two dimensions and that it can implemented in a massively parallel solution framework.

Computational expense becomes an important factor when solving complex 3-D problems. We have demonstrated that the scaling efficiency of the DFG-partitioning method is sufficient for the type of domain-decomposition that is popular for MPI implementations on current supercomputing systems. There is much effort being devoted towards evolving high-performance computing technology to more power-efficient heterogeneous architectures, combining both CPU and GPU units. In such systems, much of the computational work associated with the DFG-partitioning algorithm can be performed on the GPU. For example, fixed-radius neighbor search algorithms for GPU architectures have demonstrated $O(kn)$ scaling, where k is the number of neighbors per particles and n is the total number of particles (in some computational sub-domain) [73]. Thus, it is expected that the DFG-partitioning method will transition well into next-generation high-performance computing systems.

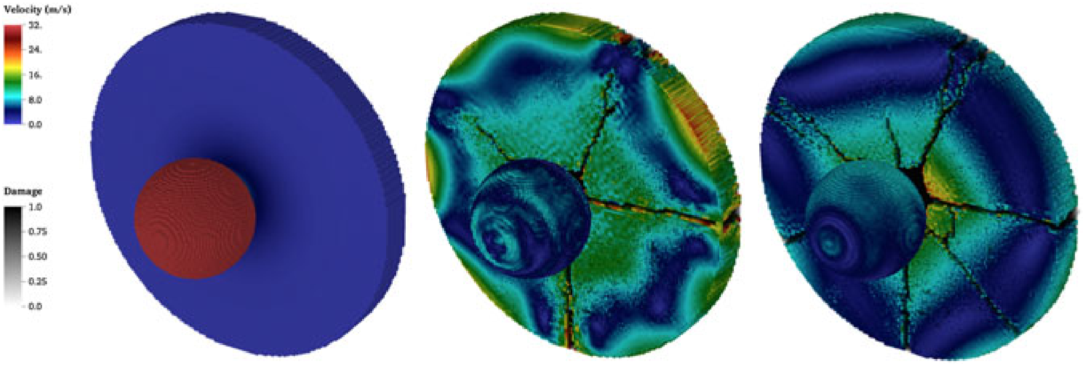

Figure 20. 3-D impact fragmentation example showing contours of velocity. Simulation uses the damage-field gradient-partioning method in a massively parallel computational framework.

4.1. Applications

Much of our focus has been on the application of DFG-partitioning to the challenging task of mesoscale simulation for brittle and porous materials. Such mesoscale models have the potential to enhance our understanding of effects such as damage, softening, induced anisotropy, fluid effects, rate effects, and thermomechanical effects, well beyond what could be obtained with laboratory experimentation. This understanding will lead to a new generation of continuum constitutive models for geomaterials as well as for new classes of exotic metamaterials that have emerged with advent additive manufacturing technology.

The MPM has already seen application in the simulation of hypervelocity penetration; the new DFG-partitioning method enhances this capability, allowing for more predictive simulations of penetration into complex materials such as sand, rock, and concrete, as well as providing a tool to assess ballistic penetration and impact fragmentation. The DFG-partitioning method may also be suitable for moderate-rate applications, such as modeling powder processing, grain harvesting, excavation, or natural events such as land slides, avalanches, and ice floes.

In addition to the application space for the enhanced capability to model fracture, the self-contact method itself may have valuable applications. For example, allowing for more accurate simulation of collapse in energy absorbing structures such as the foams used in protective head gear, or the crumple zones in an automobile.

4.2. Limitations

The DFG velocity partitioning method allows for more reasonable post-failure response when using continuum damage models in the MPM. The validity of the method is limited by the ability of the continuum damage model to predict failure of the material. For the test cases presented herein, we have used a very simple brittle failure model, which produces realistic Mode I failure and relatively mesh-independent fracture patterns when combined with spatially heterogeneous and size-dependent material strength. The DFG-partitioning method is not limited to such brittle failure models and could be also be applied to model ductile fracture if a suitable continuum damage measure were defined. In simulating ductile failure or shear band formation, it is generally necessary to employ a non-local constitutive model [74] formulation to avoid mesh dependence of the results; however, this is a constitutive modeling issue, independent of the methods presented herein.

Contact surfaces in the MPM are resolved to within a single grid cell spacing, which not only has a beneficial smoothing effect at stair-step particle boundaries, but can also result in a nonphysical gap at the contact surface between two approaching bodies. This can be seen clearly in Figure 6, but is a general characteristic of contact in MPM, not an effect of the surface detection or DFG-partitioning method. When simulating contact between large bodies, the volume of this boundary gap is generally negligible, but for compaction of granular materials (or porous materials with a high internal surface area), this could lead to a value of the continuum density that is inconsistent with the particle density. This error decreases with mesh refinement but should be accounted for when interpreting mesoscale results.

4.3. Conclusions

The DFG partitioning method in the MPM allows for separation and frictional contact at fracture surfaces with continuum damage/softening models, while avoiding the need to construct and evolve explicit fracture surfaces.

We have identified nonphysical behaviors that arise in the use of traditional SF MPM with constitutive models that include softening or damage, and which manifest as mesh dependence of fracture patterns, or spurious dissipation at fracture surfaces. We have demonstrated that the DFG-partitioning method mitigates this nonphysical behavior in simulation of dynamic fracture, fragmentation, and comminution. Additionally, we have shown that a straightforward extension of the DFG-partitioning method enables frictional contact without the *a priori* identification of contacting bodies, a limitation of traditional approaches for contact in the MPM. We have demonstrated that this new approach produces accurate results in large-deformation frictional contact problems.

The MPM can solve a broad class of problems involving complex constitutive models, large deformations, complex geometries, and deforming contact surfaces, which cannot be solved adequately with traditional Eulerian, Lagrangian, or ALE/remapping schemes. The DFG partitioning significantly improves the localization response of the MPM for constitutive models involving damage or softening and also allows for more realistic post-fracture frictional contact and granular flow.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Joint DoD/DOE Munitions Technology Development Program (JMP) for partial funding to perform this work. The authors are also very grateful for invaluable assistance provided by Randolph Settgast, in the implementation of the field-gradient partitioning method within the GEOS computational framework.

The work was performed under the auspices of the US Department of Energy by University of California, Lawrence Livermore National Laboratory under contract PND-L45911-WFO-DOD. The work was supported by National Nuclear Security Administration release LLNL-JRNL-679544 DE-AC52-07NA27344.

REFERENCES

1. Walley S. Historical review of high strain rate and shock properties of ceramics relevant to their application in armour. *Advances in Applied Ceramics* 2010; **109**(8):446–466.
2. Huang P, Zhang X, Ma S, Huang X. Contact algorithms for the material point method in impact and penetration simulation. *International Journal for Numerical Methods in Engineering* 2011; **85**(4):498–517.
3. Backman ME, Goldsmith W. The mechanics of penetration of projectiles into targets. *International Journal of Engineering Science* 1978; **16**(1):1–99.
4. Anderson CE, Behner T, Orphal DL, Nicholls AE, Templeton DW. Time-resolved penetration into pre-damaged hot-pressed silicon carbide. *International Journal of Impact Engineering* 2008; **35**(8):661–673.
5. Johnson GR, Holmquist TJ. An improved computational constitutive model for brittle materials. In *High-pressure science and technology 1993*, Vol. 309. AIP Publishing, 1994; 981–984.
6. McCauley J, Strassburger E, Patel P, Paliwal B, Ramesh K. Experimental observations on dynamic response of selected transparent armor materials. *Experimental Mechanics* 2013; **53**(1):3–29.
7. Fortov V, Sultanov V, Shutov A. Chelyabinsk superbolide explosion in the Earth's atmosphere: a common phenomenon or unique coincidence? *Geochemistry International* 2013; **51**(7):549–567.
8. Melosh H. Impact ejection, spallation, and the origin of meteorites. *Icarus* 1984; **59**(2):234–260.
9. Grady DE. Length scales and size distributions in dynamic fragmentation. *International Journal of Fracture* 2010; **163**(1-2):85–99.
10. Movshovitz N, Asphaug E, Korycansky D. Numerical modeling of the disruption of comet d/1993 f2 shoemaker-levy 9 representing the progenitor by a gravitationally bound assemblage of randomly shaped polyhedra. *The Astrophysical Journal* 2012; **759**(2):93.
11. Lomov I, Herbold EB, Antoun TH, Miller P. Influence of mechanical properties relevant to standoff deflection of hazardous asteroids. *Procedia Engineering* 2013; **58**:251–259.
12. Marusich T, Ortiz M. Modelling and simulation of high-speed machining. *International Journal for Numerical Methods in Engineering* 1995; **38**(21):3675–3694.
13. Lewis J, Goldsmith W. The dynamic fracture and prefracture response of compact bone by split Hopkinson bar methods. *Journal of Biomechanics* 1975; **8**(1):27–40.
14. Shockey DA, Curran DR, Seaman L, Rosenberg JT, Petersen CF. Fragmentation of rock under dynamic loads. In *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, Vol. 11. Elsevier, 1974; 303–317.
15. Grady D. Local inertial effects in dynamic fragmentation. *Journal of Applied Physics* 1982; **53**(1):322–325.
16. Parab N D, Claus B, Hudspeth MC, Black JT, Mondal A, Sun J, Fezzaa K, Xiao X, Luo S, Chen W. Experimental assessment of fracture of individual sand particles at different loading rates. *International Journal of Impact Engineering* 2014; **68**:8–14.
17. Allen WA, Mayfield EB, Morrison HL. Dynamics of a projectile penetrating sand. *Journal of Applied Physics* 1957; **28**(3):370–376.
18. Camacho GT, Ortiz M. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures* 1996; **33**(2):2899–2938.
19. Dwivedi SK, Teeter RD, Felice CW, Gupta YM. Two dimensional mesoscale simulations of projectile instability during penetration in dry sand. *Journal of Applied Physics* 2008; **104**(8):083502-1–083502-10.
20. Brannon RM, Fossum AF, Strack OE. Kayenta: theory and user's guide. *Technical Report Report No. SAND2009-2282*, Sandia, 2009.
21. Rubin M, Einav I. A large deformation breakage model of granular materials including porosity and inelastic distortional deformation rate. *International Journal of Engineering Science* 2011; **49**(10):1151–1169.

22. Paliwal B, Ramesh K. An interacting micro-crack damage model for failure of brittle materials under compression. *Journal of the Mechanics and Physics of Solids* 2008; **56**(3):896–923.
23. Vorobiev O. Generic strength model for dry jointed rock masses. *International Journal of Plasticity* 2008; **24**(12):2221–2247.
24. Wriggers P, Moftah S. Mesoscale models for concrete: homogenisation and damage behaviour. *Finite Elements in Analysis and Design* 2006; **42**(7):623–636.
25. Benkemoun N, Hautefeuille M, Colliat JB, Ibrahimbegovic Adnan. Failure of heterogeneous materials: 3d meso-scale fe models with embedded discontinuities. *International Journal for Numerical Methods in Engineering* 2010; **82**(13):1671–1688.
26. Kim SM, Al-Rub RKA. Meso-scale computational modeling of the plastic-damage response of cementitious composites. *Cement and Concrete Research* 2011; **41**(3):339–358.
27. Ramesh K, Hogan JD, Kimberley J, Stickle A. A review of mechanisms and models for dynamic failure, strength, and fragmentation. *Planetary and Space Science* 2015; **107**:10–23.
28. Sulsky D, Chen A, Schreyer H. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**:179–196.
29. Homel MA, Guilkey J, Brannon RM. Continuum effective-stress approach for high-rate plastic deformation of fluid-saturated geomaterials with application to shaped-charge jet penetration. *Acta Mechanica* 2015; **10**:1007.
30. Kamojjala K, Brannon R, Sadeghirad A, Guilkey J. Verification tests in solid mechanics. *Engineering with Computers* 2013; **31**(2):193–213.
31. Tonge AL, Kimberley J, Ramesh K. A consistent scaling framework for simulating high rate brittle failure problems. *Procedia Engineering* 2013; **58**:692–701.
32. Homel MA, Brannon RM, Guilkey J. Controlling the onset of numerical fracture in parallelized implementations of the material point method (MPM) with convective particle domain interpolation (CPDI) domain scaling. *International Journal for Numerical Methods in Engineering* 2015.
33. Guilkey JE, Bardenhagen S, Roessig K, Brackbill J, Witzel W, Foster J. Improved contact algorithm for the material point method and application to stress propagation in granular material. *Computer Modeling in Engineering & Sciences* 2001; **2.4**:509–522.
34. Nairn J. Material point method calculations with explicit cracks. *Computer Modeling in Engineering and Sciences* 2003; **4**:649–663.
35. Guo Y, Nairn J. Three-dimensional dynamic fracture analysis using the material point method. *Computer Modeling in Engineering and Sciences* 2006; **16**:141–155.
36. Daphalapurkar N, Lu H, Coker D, Komanduri R. Simulation of dynamics crack growth using the generalized interpolation material point (gimp) method. *International Journal of Fracture* 2007; **143**:79–102.
37. Sadeghirad A, Brannon R, Guilkey J. Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces. *International Journal for Numerical Methods in Engineering* 2013; **95**(11):928–952.
38. Bardenhagen S, Kober E. The generalized interpolation material point method. *CMES-Computer Modeling in Engineering and Sciences* 2004; **5**:477–495.
39. Sadeghirad A, Brannon R, Burghardt J. A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations. *International Journal for Numerical Methods in Engineering* 2011; **86**(12):1435–1456.
40. Wallstedt P, Guilkey J. An evaluation of explicit time integration schemes for use with the generalized interpolation material point method. *Journal of Computational Physics* 2008; **227**:9628–9642.
41. Sulsky D, Chen Z, Schreyer HL. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**(1):179–196.
42. Bardenhagen S, Brackbill J, Sulsky D. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering* 2000; **187**(3):529–541.
43. Wriggers P, Laursen TA. *Computational Contact Mechanics*, Vol. 30167. Springer, 2006.
44. Bardenhagen S, Guilkey J, Roessig K, Brackbill J, Witzel W, Foster J. An improved contact algorithm for the material point method and application to stress propagation in granular material. *Computer Modeling in Engineering and Sciences* 2001; **2**:509–522.
45. Appel K, Haken W. Every planar map is four colorable. *Mathematical Solitaires & Games* 1989; **98**:1-741.
46. Chaboche JL. Continuum damage mechanics: part II – damage growth, crack initiation, and crack growth. *Journal of Applied Mechanics* 1988; **55**(1):65–72.
47. Strack O, Leavy R, Brannon R. Aleatory uncertainty and scale effects in computational damage models for failure and fragmentation. *International Journal for Numerical Methods in Engineering* 2014; **102**(2-4):468–495.
48. Song JH, Wang H, Belytschko T. A comparative study on finite element methods for dynamic fracture. *Computational Mechanics* 2008; **42**(2):239–250.
49. Daphalapurkar NP, Lu H, Coker D, Komanduri R. Simulation of dynamic crack growth using the generalized interpolation material point (GIMP) method. *International Journal of Fracture* 2007; **143**(1):79–102.
50. Monaghan JJ. Smoothed particle hydrodynamics. *Reports on Progress in Physics* 2005; **68**(8):1703.
51. Liu WK, Jun S, Zhang YF. *et al.*Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**(8-9):1081–1106.
52. Gyulassy A, Natarajan V. Topology-based simplification for feature extraction from 3d scalar fields. *Vis 05. IEEE*: IEEE, 2005; 535–542.

DOI: 10.1002/nme

53. Meza LR, Das S, Greer JR. Strong, lightweight, and recoverable three-dimensional ceramic nanolattices. *Science* 2014; **345**(6202):1322–1326.
54. Marrone S, Colagrossi A, Le Touzé D, Graziani G. Fast free-surface detection and level-set function definition in SPH solvers. *Journal of Computational Physics* 2010; **229**(10):3652–3663.
55. Meng Q, Berzins M. Scalable large-scale fluid–structure interaction solvers in the Uintah framework via hybrid task-based parallelism algorithms. *Concurrency and Computation: Practice and Experience* 2014; **26**(7):1388–1407.
56. Gargantini I. Linear octtrees for fast processing of three-dimensional objects. *Computer Graphics and Image Processing* 1982; **20**(4):365–374.
57. Settgast R, Johnson S, Fu P, Walsh S, Annavarapu C, Hao Y, White J, Ryerson F. GEOS: a framework for massively parallel multi-physics simulations. theory and implementation. *Technical Report 654611*, LLNL, 2014.
58. Settgast R, Fu P, Walsh S, White J, Annavarapu C, Ryerson F. A fully coupled method for massively parallel simulation of hydraulically driven fractures in 3-dimensions. *International Journal of Numerical and Analytical Methods in Geomechanics* 2016. (in press).
59. Childs H, Brugger E, Whitlock B, Meredith J, Ahern S, Pugmire D, Biagas K, Miller M, Harrison C, Weber GH., Krishnan H, Fogal T, Sanderson A, Garth C, Bethel EW, Camp D, Rübel O, Durant M, Favre JM, Navrátil P. VisIt: an end-user tool for visualizing and analyzing very large data. *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, 2012; 357–372.
60. Fu Y, Ogden R. *Nonlinear Elasticity: Theory and Applications*, Lecture note series. Cambridge University Press: Cambridge, United Kingdom, 2001.
61. Cervera M, Chiumenti M. Mesh objective tensile cracking via a local continuum damage model and a crack tracking technique. *Computer Methods in Applied Mechanics and Engineering* 2006; **196**(1):304–320.
62. Sharon E, Fineberg J. Confirming the continuum theory of dynamic brittle fracture for fast cracks. *Nature* 1999; **397**(6717):333–335.
63. Weibull W. Wide applicability. *Journal of Applied Mechanics* 1951; **103**:293–297.
64. Jayatilaka AdS, Trustrum K. Statistical approach to brittle fracture. *Journal of Materials Science* 1977; **12**(7): 1426–1430.
65. Tonge AL, Ramesh K. Multi-scale defect interactions in high-rate failure of brittle materials, part ii: application to design of protection materials. *Journal of the Mechanics and Physics of Solids* 2016; **86**:237–258.
66. Kalthoff J, Winkler S. Failure mode transition at high rates of shear loading. *DGM Informationsgesellschaft mbH, Impact Loading and Dynamic Behavior of Materials* 1988; **1**:185–195.
67. Belytschko T, Chen H, Xu J, Zi G. Dynamic crack propagation based on loss of hyperbolicity and a new discontinuous enrichment. *International Journal for Numerical Methods in Engineering* 2003; **58**(12):1873–1905.
68. Kalthoff J. Transition in the failure behavior of dynamically shear loaded cracks. *Applied Mechanics Reviews* 1990; **43**(5S):S247–S250.
69. Nairn J, Guo Y, Kourkoulis S. 2940-material point method calculations with explicit cracks, fracture parameters, and crack propagation. *Icf11, Italy 2005*, 2013; 649–664.
70. Timoshenko SP, Goodier JN. *Theory of Elasticity, Engineering Societies Monograph*, Vol. 1. MacGraw-HillBook Company: New York City, 1970.
71. Di Stefano L, Bulgarelli A. A simple and efficient connected components labeling algorithm. *Image Analysis and Processing, 1999. Proceedings. International Conference on*: IEEE, 1999; 322–327.
72. Sulsky D, Zhou S, Schreyer H. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**:236–252.
73. Green S. Particle simulation using CUDA. *NVIDIA whitepaper*, Vol. 6, Santa Clara, CA, 2010; 121-128.
74. Burghardt J, Brannon R, Guilkey J. A nonlocal plasticity formulation for the material point method. *Computer Methods in Applied Mechanics and Engineering* 2012; **225**:55–64.