

Visualization of Fracture Progression in Peridynamics

Michael Bußler^{a,*}, Patrick Diehl^b, Dirk Pflüger^d, Steffen Frey^a, Filip Sadlo^c, Thomas Ertl^a, Marc Alexander Schweitzer^{b,e}

^aVisualization Research Center, University of Stuttgart

^bInstitute for Numerical Simulation, University of Bonn

^cInterdisciplinary Center for Scientific Computing, Heidelberg University

^dIPVS/SGS, University of Stuttgart

^eMultiscale Meshfree Methods, Fraunhofer SCAI

Abstract

We present a novel approach for the visualization of fracture processes in peridynamics simulations. In peridynamics simulation, materials are represented by material points linked with bonds, providing complex fracture behavior. Our approach first extracts the cracks from each time step by means of height ridge extraction. To avoid deterioration of the structures, we propose an approach to extract ridges from these data without resampling. The extracted crack geometries are then combined into a spatiotemporal structure, with special focus on temporal coherence and robustness. We then show how this structure can be used for various visualization approaches to reveal fracture dynamics, with a focus on physical mechanisms. We evaluate our approach and demonstrate its utility by means of different data sets.

Keywords: Multifield Visualization, Volume and Flow Visualization, Ridge Extraction, Peridynamics, Crack and Fracture

1. Introduction

In engineering and material science, cracks and fractures in solids are of great interest. One way to model and simulate fractures is the recently developed peridynamics (PD) theory [1], a non-local generalization of continuum mechanics with a focus on discontinuous solutions as they arise in fracture mechanics. Peridynamics simulation provides the process of fracture development over time, from initial cracks to complete fracture growth. In peridynamics simulation, materials are represented by material points linked with bonds. The bonds are defined to break irreversibly if the displacement between the respective material points exceeds a certain limit, providing complex fracture behavior in arbitrarily-shaped objects. In peridynamics data, fractures are implicitly modeled through a damage field, which, for each material points, gives the fraction of broken bonds to the initial number of bonds.

Domain experts in the field of peridynamics are highly interested in gaining insights into the fracturing process to evaluate and refine their material models. Previous to our work, fracture progression in peridynamics had been mostly examined through direct depiction of the damage field [2, 3, 4, 5]. This was achieved by visualizing the damage, typically using a direct rendering of the nodes colored by their damage (the percentage of broken bonds), as shown in Figure 1a. However, this visualization approach does not allow one to understand the internal fracture progression within the material, as only boundary nodes are visible. The clear structure of the fracture cannot

be seen. It therefore requires tedious manual analysis to determine the location of crack tips, and it renders the identification of fragments difficult. It puts focus on the local damage at the material points, while the “global” progression of the cracks is difficult to see. Most importantly, this merely visual representation cannot be used for further analysis tasks, like the estimation of fracture progression velocity.

For domain scientists, it is an important aspect to see the crack surfaces between the material points to understand their growth and bifurcation. Of special interest is therefore the extraction and visualization of the crack surface, its temporal progression, and its growth velocity over time. This allows for a comparison to results of real experiments and to calibrate the model parameters to match real materials [4]. Domain scientists also want to examine how the stress behaves at the surface and near the crack front to obtain a deeper understanding of the fracture process.

To address these demands, we propose to approximate fracture geometry from the damage field by means of height ridge extraction. By extracting fracture geometry for every time step, we would be able to visualize the temporal progression of the fracture, but the ridge extraction process would usually not be temporally stable. We therefore propose a technique to progressively combine ridges into a single reference state to obtain a temporally stable representation of fracture growth. With this technique, we can visualize the temporal progression of the fracture and estimate fracture growth velocity. Furthermore, we can visualize the stress tensor around the growing fractures to obtain a deeper understanding of the fracture process.

Note that standard ridge surface extraction techniques from scalar field analysis have to be significantly modified. With peridynamics, we do not have a traditional mesh, not even an

*Corresponding author

Email address: Michael.Bussler@visus.uni-stuttgart.de
(Michael Bußler)

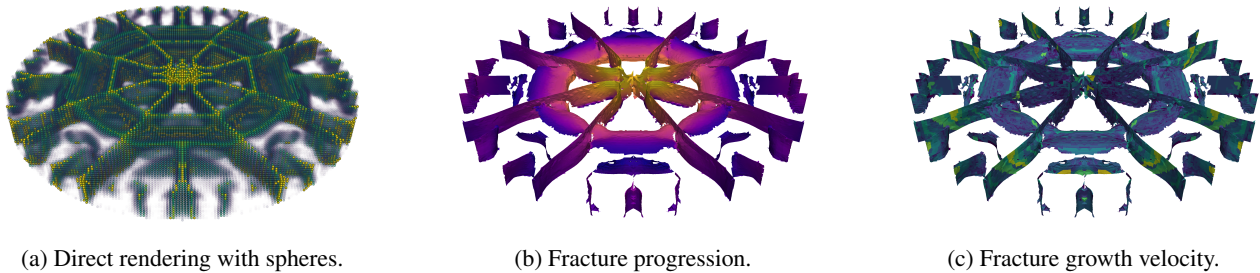


Figure 1: Visualization of fracture dynamics in the Plate data set, a peridynamics simulation of a thin plate impacted at its center by a spherical projectile from above. (a) Direct rendering of the data set using spheres where color indicates damage from blue (no damage) to red (completely broken). (b), (c) Geometry of the fractures calculated using our temporally stable crack extraction technique. Color in (b) indicates progression of the cracks over time (from blue to red), color in (c) shows the growth velocity of the crack front (blue slow, red fast). Note that, throughout the paper, we use the Plasma transfer function to visualize temporal progression, and the Viridis transfer function for rendering growth velocity.

unstructured one. Instead, we have a graph-like setting of nodes that are connected by bonds not only to neighboring ones, but to many more which extend well beyond their nearest neighbors. Taking only a scalar damage value per material point into account is not sufficient. An obvious way out would be a resampling of the damage field, such as an approximation on a structured grid. This can be achieved using Sheppard’s method or radial basis functions, for example. Based on that, approaches to compute ridges, such as the marching ridges approach of Furst et al. [6], which requires a structured grid as input data, could be applied. However, such a resampling step would introduce additional smoothing and a potentially critical loss of detail and accuracy, especially if the resolution is too low compared to the frequencies present in the damage field. Choosing a very high resolution for the resampling would lead to increased processing times and prohibitively large memory consumption. Furthermore, thresholds have to be introduced and adapted to match tedious manual inspection and automatic extraction in any case. But these are much better motivated and justified when working with peridynamics data than with resampled data. We therefore propose to calculate the ridges directly from the material points without resampling.

The peridynamics representation of a material by material points and bonds poses further challenges. Its nature as scattered data makes it hard to obtain stable ridges, as standard approaches work best on smooth and continuous data. Furthermore, it requires further considerations to obtain temporal stability of the ridge growth over time. At the point of impact, scattered material points can change their position significantly, and the breaking of bonds is a discrete event which leads to non-smooth behavior of the damage property.

Our contributions are based on joint interdisciplinary work of researchers from the modeling and the visualization domains. They include:

- The first extraction of fractures on peridynamics data using height ridges directly on node–bond-based input data without resampling.
- Assurance of temporal stability by progressively combining ridges into a reference state over time.

- A spatiotemporal visualization of fracture progression, growth velocity, and correlation with stress.

Our paper is structured as follows: Section 2 gives an overview on related work, in Section 3 a brief introduction to peridynamics is presented. We describe our techniques for height ridge extraction in Section 4, temporal stability in Section 5, and fracture progression visualization in Section 6. Section 7 shows the description of the three data sets, which we used in our visualization, followed by our results. In Section 7.8, our extracted crack growth velocity of the ridges is compared to experimental data. Finally, Section 8 concludes our work.

2. Related Work

Ridge extraction. We employ height ridge extraction to approximate the fracture geometry in peridynamics data. This result provides the basis for analyzing their temporal progression, enabling operations like the computation of distances or the estimation of progression velocity. In other scenarios, in which such a representation is not required but ridges are used for rendering only, also implicit approaches like the raycasting technique of Barakat and Trichoche [7] could be used. Closely related definitions for height ridges are given by Haralick [8], Eberly [9], Lindeberg [10], and Schultz et al. [11]. We base our approach on Eberly’s definition due to its wide use. The marching ridges method by Furst et al. [6] is somewhat related to our work, although we extract 2-dimensional height ridges in 3D by a marching approach on tetrahedral grids based on marching tetrahedra, instead of user-intervened tracing of ridges through hexahedral grids. Thus, the most closely related work is the marching approach by Sadlo et al. [12]. Further, details on height ridge extraction and filtering can also be found in the work by Peikert and Sadlo [13]. Ridge extraction has been employed in various application domains. Kindlmann et al. [14] used ridge extraction to find skeletal structures in magnetic resonance imaging diffusion tensor data. Surface reconstruction from point cloud density using ridge extraction was described by Süßmuth and Greiner [15]. Wu et al. [16] present improvements to extract boundary surfaces from deformable bodies with changing topology (e.g., due to cuts and incisions)

both more accurately and efficiently. While their problem scenarios and goals significantly differ from ours, there are some conceptual similarities with respect to surface reconstruction. However, while they aim to represent boundary surfaces from uniform grids (with linked grid cells) that are accurately aligned with a user-defined cut (and they specifically design their techniques to achieve that), we extract fractures from scattered data values given at the material point positions from a peridynamics simulation, and we have no additional higher-resolution information available to incorporate.

Height ridge surface extraction in space-time has been employed by Bachthaler et al. [17] for analyzing the dynamics within Lagrangian coherent structures, and Heßel et al. [18] for analyzing wave propagation. However, none of the above approaches addressed robust temporal extraction in 3D space.

Fracture in Computer Graphics. Physics-based animation of deforming plastic materials and fracture effects in computer graphics was introduced by Terzopoulos et al. [19] and developed further to model brittle fracture by O’Brien et al. [20]. Meshless methods have been introduced to computer graphics by Desbrun et al. [21], who model soft inelastic materials with implicit surfaces. Müller et al. [22] present a method for modeling and animating point-based elastically and plastically deforming objects, which was extended by Pauly et al. [23] to simulate fracturing solids, where fracture surfaces were represented using elliptical splats. Brittle fracture using linear elastic fracture mechanics was recently described by Hahn et al. [24] who model the crack front as a Lagrangian flow to describe fracture dynamics. Peridynamics simulations are somewhat akin to spring–mass models which were used by Hirota et al. [25] to model cracking behavior. Peridynamics simulation for animation of brittle fracture was recently revisited by Levine et al. [26], who also present different strategies for obtaining surface geometry for rendering. They model the surface of pieces falling apart; in contrast, we focus on the crack propagation inside the material before separation. Extraction of surface geometry from peridynamics data was recently described by Watcharopas et al. [27]. They also use a tetrahedral mesh from the given material point set but obtain fracture geometry from the connectivity by splitting the tetrahedra, whereas we propose the extraction of fracture geometry as height ridges of the damage field. For more details we refer to the survey of Wu et al. [28].

Temporal Development. In this paper, we extract ridges as a basis for analyzing the temporal development in the data. Likewise, there is a large body of work relying on different kinds of extracted features. Here, a large body of work is based on analyzing the similarity of time activity curves or similar measures that describes each voxel’s (or voxel a block’s) time series [29, 30, 31]. Schneider et al. [32] compare scalar fields on the basis of the largest contours. Widanagamaachchi et al. [33] employ tracking graphs of features to visualize large-scale time-varying data. Frey et al. [34] present a visualization approach based on similarity matrices that allows for detecting and exploring similarity in the temporal variation of field data.

For the presentation of temporal developments on the basis of features, we employ different visualization techniques on

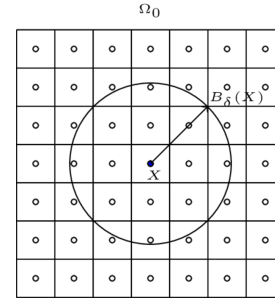


Figure 2: The continuum in the reference configuration Ω_0 with the finite interaction zone of length δ for the material point at \mathbf{X} . All material points inside the interaction zone $B_\delta(\mathbf{X})$ of \mathbf{X} are connected with bonds to exchange forces.

crack surfaces to visualize flow velocities, eigenvectors, etc. Alternative approaches for the visualization of time-varying data include illustration-inspired techniques like speed lines or flow ribbons [35]. Lu and Shen [36] propose interactive storyboards that compose sample volume renderings and descriptive geometric primitives. Bach et al. review temporal data visualization techniques by interpreting them as series of operations performed on a space-time cube [37] (e.g., Woodring and Shen [38] consider time-varying (volumetric) data as a four-dimensional data field, and apply high-dimensional slicing and projection techniques).

In this work, we also employ multi-field visualization, e.g., to jointly visualize fracture progression and stress. Among others, Lee and Shen [39] identify and visualize trend relationships in multivariate time-varying data. Our approach can further be used for comparative visualization and analytics to support the understanding of similarities and differences of a collection of peridynamics data sets. For comparative visualization, VisTrails provides a framework for comparing visualization results with highly customizable parameter changes: e.g., mesh discretization, isovalue, or manually adjusted visualization pipelines [40].

3. Peridynamics

Here, we review fundamentals in peridynamics that we use in the following for the simulation of cracks and fractures in solids. The principle of peridynamics is that material points inside an interaction zone $B_\delta(\mathbf{X})$ are connected with bonds and exchange forces (Figure 2). The bond between two material points is broken after its stretch exceeds the critical value for bond stretch s_c . Thus, the damage $c(t, \mathbf{X})$ of a material point is defined by the ratio between broken bonds at time t and the initial number of bonds in the interaction zone $B_\delta(\mathbf{X})$. Figure 1a shows a simple visualization with spheres at the actual position $\mathbf{x}(t, \mathbf{X})$, colored with the damage $c(t, \mathbf{X})$ from blue (all initial bonds exist) to red (all initial bonds broken). The peridynamic equation of motion (1) delivers the acceleration $\mathbf{a}(t, \mathbf{X})$ for a material point in the reference configuration Ω_0 by the exchange of a pairwise force \mathbf{f} with all material points inside the

finite interaction zone

$$\rho(\mathbf{X})\mathbf{a}(t, \mathbf{X}) = \int_{B_\delta(\mathbf{X})} \mathbf{f}(t, \mathbf{x}(t, \mathbf{X}') - \mathbf{x}(t, \mathbf{X}), \mathbf{X}' - \mathbf{X}) d\mathbf{X}' + b(t, \mathbf{X}). \quad (1)$$

Here, $b(t, \mathbf{X})$ denotes an external volume force, e.g., for modeling the impact of a projectile. Despite the force and the acceleration, Table 1 shows additional attributes available per material point at position \mathbf{X} in the reference configuration Ω_0 . For more details about peridynamics, we refer for theory to Silling [1] and to Silling and Askari [41] for numerical details.

3.1. Importance of Crack Surfaces

The initiation of cracks, without any initial description of the cracks, is one of the benefits of peridynamics. The damage $c(t, \mathbf{X})$ values defined at the material point positions \mathbf{X} indirectly describe the progression of fractures within the object approximated by the node set for consecutive time steps. On the one hand, peridynamics is applicable for the initiation of cracks. On the other hand, the discretization with material points, due to the weak convergence, is computationally expensive. Thus, it is not possible to simulate a large structure, e.g., the wing of an airplane, with pure peridynamics. Therefore, different approaches for coupling PD with other computationally cheaper methods, like finite elements [42, 43, 44, 45] or partition of unity methods [46, 47, 48], exist. In the area of interest, where a crack may occur or an initial crack exists, the computationally expensive peridynamics is simulated and around this area a computationally cheaper method, e.g., finite elements, is applied. In these approaches, the coupling is done with attributes of the material points or extracted information of the displacement field of the material points.

These coupling schemes use the attributes (Table 1) at the material points or extracted information of the displacement field. A novel approach would be to use the crack surface for a “geometrical” coupling. Here, after the fracture has grown and the changes over time are sufficiently small, the meshed ridges can be integrated in the mesh of the computationally cheaper method.

The Kalthoff Winkler experiment is a standard problem in dynamic fractures and a benchmark for the validation of the crack angle from the crack to the initial crack or the crack growth velocity [49, 3]. With the extracted ridges and the resulting crack growth velocity of our technique, we can compare

Table 1: Attributes per material point at location \mathbf{X} in peridynamics simulation.

Attribute	Symbol	Unit
Actual position	$\mathbf{x}(t, \mathbf{X}) \in \mathbb{R}^3$	m
Damage	$c(t, \mathbf{X}) \in \mathbb{R}$	%
Displacement	$\mathbf{d}(t, \mathbf{X}) \in \mathbb{R}^3$	m
Initial position	$\mathbf{X} \in \mathbb{R}^3$	m
Velocity	$\mathbf{v}(t, \mathbf{X}) \in \mathbb{R}^3$	ms^{-1}
Stress	$\boldsymbol{\sigma}(t, \mathbf{X}) \in \mathbb{R}^{3 \times 3}$	Pa

the crack angle or the crack growth velocity with experimental data and other simulation techniques, like smoothed-particle hydrodynamics (SPH) (see Section 7.8).

Obtaining the velocity at the crack front is difficult with the plain material points of the peridynamic simulation. Fineberg and Marder [50] emphasize the comparison for predictions of the linear elastic fracture mechanics (LEFM) theory and experiments. One example is the velocity at the crack front, which often disagrees with the predictions of the LEFM theory [51]. By estimating the fracture growth velocity, the peridynamics simulation can be validated against the LEFM theory and the behavior obtained in the experiments. This would allow us to see more detailed how the crack velocity behaves.

4. Extraction of Fractures

Our goal is the extraction of fracture geometry from peridynamics data. We propose to approximate the crack geometry from the scalar-valued damage field given at the material points using height ridge extraction. The required steps for the extraction of the crack geometry for one point in time are:

1. Height ridge extraction (Section 4.1)
2. Estimation of gradient and Hessian (Section 4.2)
3. Clipping of Delaunay grid (Section 4.3)
4. Ridge extraction based on Marching Tetrahedra (Section 4.4)
5. Filtering (Section 4.5)

4.1. Height Ridge Extraction

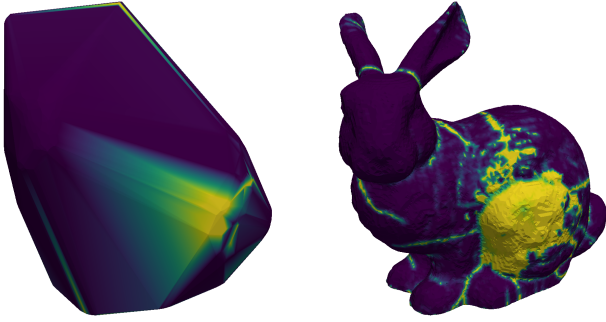
In our implementation, the crack geometry is estimated as the height ridge surfaces of the damage field $c(t, \mathbf{X})$. Height ridges are present where the scalar-valued damage field exhibits a local maximum in at least one direction [9]. Assuming that the damage is given as a scalar field $c : \mathbb{R}^3 \rightarrow \mathbb{R}$, let $\mathbf{g} = \nabla c$ be the gradient of the scalar field and \mathbf{H} be its Hessian with eigenvectors \mathbf{e}_i and respective eigenvalues $\lambda_i, i \in 1, 2, 3$, which are sorted such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Then, a height ridge surface is given by the conditions

$$\mathbf{g} \cdot \mathbf{e}_3 = 0 \quad \wedge \quad \lambda_3 < 0, \quad (2)$$

meaning that the first directional derivative in direction \mathbf{e}_3 is zero, whereas the second directional derivative in \mathbf{e}_3 -direction is negative.

4.2. Estimation of Gradient and Hessian

The input for our approach consists of scattered data values given at the material point positions. In order to obtain smooth gradients \mathbf{g} from the damage field, we estimate them using least squares approximation [12]. For the calculation, we collect the neighboring material points for each material point position within a specific range d_r , and use their damage values as weights to fit the gradient in the least squares sense. The same approach is used to calculate the Hessian \mathbf{H} , where in this case we use the estimated gradients at the material points as input. Finally, we calculate the eigenvectors and eigenvalues from the Hessian at each material point.



(a) Geometry after Delaunay triangulation. (b) Geometry after clipping.

Figure 3: Tetrahedral mesh of the Bunny data set at $t = 890 \times 10^{-7}$ s after Delaunay triangulation (a) and after clipping the cells by maximum edge length (b), where color indicates damage.

4.3. Clipping of Delaunay Grid

To avoid resampling, we calculate the ridges directly on the scattered data points. To this end, we create a tetrahedral mesh from the data points using Delaunay triangulation and then calculate the ridges on the unstructured grid based on the marching tetrahedra algorithm [52]. Notice that the Voronoi diagram and thus the Delaunay triangulation is well-defined and robust with respect to perturbations of the data points [53, 54]. Together with the fact that Delaunay triangulation guarantees local properties, it represents a robust means for height ridge extraction from scattered point data. The Delaunay triangulation is calculated according to Si [55], and produces a convex hull from the input data points, which can lead to unexpected ridges in the hull area (cf. Figure 3a). As we are interested in the geometry of the cracks within the object, our solution is to clip the tetrahedral cells at the boundary area. To identify such cells, we exploit the fact that cells within the object have a different shape and size than those at the boundary. Here, we simply use the maximum edge length and remove tetrahedral cells with a cell size larger than a threshold d_e . This is reasonable, as the material points within the object are evenly distributed at the beginning of the simulation and displacement of the material points within the object during the simulation is limited. The effect of this clipping step on the Bunny data set is shown in Figure 3b.

4.4. Ridge extraction based on Marching Tetrahedra

Height ridge extraction according to Eberly [9] is done by evaluating Equation (2) for each grid cell and checking whether the ridge crosses one or more of the cell's edges. This is indicated by a change of the sign of $\mathbf{g} \cdot \mathbf{e}_3$ from one edge node to the other, where the intersection point is given by $\mathbf{g} \cdot \mathbf{e}_3 = 0$ and is determined using linear interpolation. Consistent with the marching tetrahedra algorithm, the edge intersections form up to two triangles within a tetrahedron. As eigenvectors lack an orientation and therefore directional derivatives may not be consistent, we employ principal component analysis to estimate a common orientation and accordingly flip vectors if required (similar to Sadlo et al. [12]).

Algorithm 1 Our approach to achieve stable fracture progression (as discussed in Section 5 and demonstrated in Figure 5).

```

1: function TEMPORALFRACTUREPROGRESSION( $F_{\square}$ )
2:    $\triangleright$  define reference fracture  $F_{\text{ref}}$ 
3:    $F_{\text{ref}} \leftarrow \text{back}(F_{\square})$   $\triangleright$  last fracture of time series is reference
4:    $\triangleright$  stabilize w.r.t.  $F_{\text{ref}}$ 
5:    $F_{-1} \leftarrow \emptyset$ 
6:   for  $t \in \{0 \dots |F_{\square}| - 1\}$  do
7:      $\triangleright$  iterate over time steps (front to back)
8:      $V \leftarrow \emptyset$ 
9:     for all  $v \in \text{vertices}(F_{\text{ref}})$  do  $\triangleright$  loop over vertices (cf. Fig. 5a)
10:      if  $\text{distance}(v, F_t) < d_g$  then
11:         $V \leftarrow V \cup v$ 
12:       $F_* \leftarrow \emptyset$ 
13:      for all  $f \in \text{faces}(F_{\text{ref}}) - F_{t-1}$  do  $\triangleright$  face loop (cf. Fig. 5b)
14:        if  $\text{vertices}(f) \subset V$  then
15:           $F_* \leftarrow F_* \cup f$ 
16:       $F_t \leftarrow F_{t-1} \cup F_*$   $\triangleright$  determine fractures that are both in  $F$  and  $F_{\text{ref}}$ 
17:   return  $F_{\square}$ 

```

4.5. Filtering

While we use least squares approximation to obtain smooth gradients and Hessian, variations in the discrete input data can still lead to unstable ridge detection. To address this, we apply post-operational filters to the resulting ridge geometry (akin to Sadlo et al. [12]). The cracks that are approximated by the ridges can only occur where a minimum damage value is given. Therefore, we remove all triangles of the ridge where the interpolated damage value is below a given threshold for at least one of the ridge nodes.

Another filtering condition concerns the size of the ridges. We expect the resulting fracture geometry to be contiguous such that we can identify fractures by a minimum number of triangles. Ridges below this threshold are most likely not cracks but visual clutter and can therefore be removed. The number of triangles per ridge is estimated by separating the set of all ridge triangles into connected components by connected component labeling.

5. Temporal Fracture Progression

We aim at visualizing the temporal progression of crack growth through the ridges extracted in each individual time step. Unfortunately, height ridge surface extraction is sensitive to noise and small deviations and thus not temporally stable in general. In particular, the filtering step can cause portions of cracks to subsequently appear and disappear in cases in which the filtering values are close to a specified threshold. This results in flickering when browsing through the ridges and is the main cause for temporal instability. As these instabilities are typically small, a simple solution would be to filter w.r.t. size (cf. Figures 4a and 4b), but then flickering still persists and actually desired geometry like newly developing cracks are removed as well.

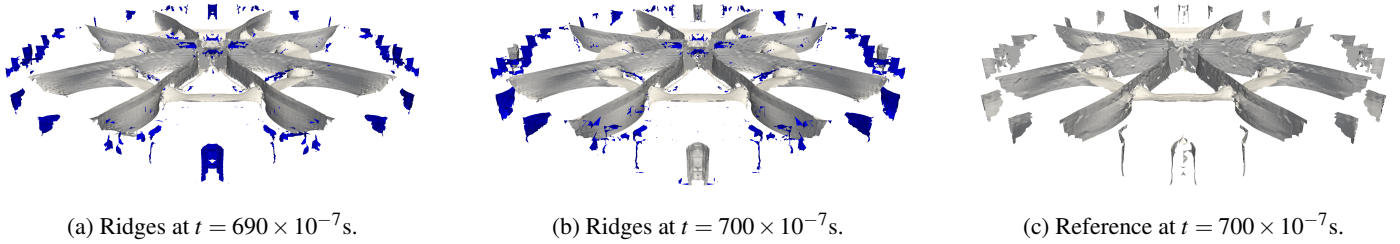
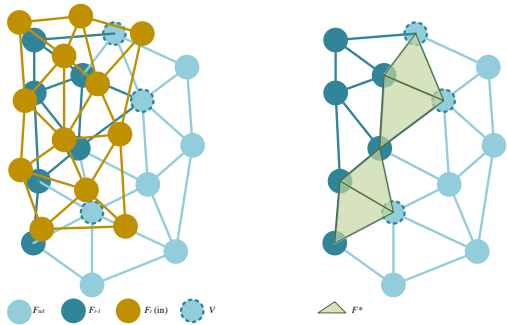


Figure 4: Filtering ridges in each time step (a), (b) may not only remove clutter but also growing cracks (blue). Our approach (c) for temporal stability uses unfiltered ridges but shows no clutter while all important features are captured.



(a) Current and reference ridge. (b) Ridge growing process.

Figure 5: Illustration of the ridge growing process. (a) Geometry in the reference ridges (F_{ref} , blue) is selected based on its distance to the ridges (F_t , yellow) extracted from current time step t . (b) Based on this selection, new geometry (green) is added to F_{t-1} to create the current set F_t .

For this, we propose a technique for stabilizing the ridge extraction over time (Algorithm 1). Our idea is to select a reference step, and use its crack geometry as a temporally stable reference for prior time steps. While this could be done in a streaming fashion as well with minor adjustments, in this paper we concentrate on the a posteriori scenario in which the full simulation is available at once. This allows us to identify the time step where the crack growth has come to a complete stop and then use its crack geometry as a reference. With this approach, we need to apply the ridge size filter only to remove visual clutter in the reference state, as shown in Figure 4c. Geometry from previous time steps is hereby used without applying the size filter, such that no important feature is missed. This is motivated by the fact that the ridges which represent cracks only grow over time, but never shrink again. Therefore, the ridges F_t in each time step t prior to the reference time step can be regarded as a subset $F_t \in F_{\text{ref}}$ of the ridges in the reference. Our approach is based on the following two assumptions (these are typically fulfilled in our domain of application):

1. cracks grow over time, and once they are there, they do not disappear anymore,
2. the change of location of generated fractures is negligible overall (i.e., there is only little “drift”).

Based on these assumptions, in most cases, we can simply use the last time step as reference F_{ref} as it contains all cracks that have been accumulated over time (Algorithm 1, Line 3).

Note that, as discussed later in Section 7.2, this last considered time step is not necessarily the last time step of the simulation input data, but may be chosen to be earlier by the user (e.g., to save performance because the fracture progression has visibly come to a stop earlier, or because the structure starts to significantly degrade after a certain point in time). Fracture geometry from the previous time steps is then progressively matched with this reference to obtain stable ridge growth (Lines 8–16). At first, we estimate the subset of the reference geometry that matches the geometry in the current time step. This matching is done by calculating the Hausdorff distance of the reference points to the mesh of the current ridges (Lines 9–11, Figure 5a). Vertices V in the reference state are selected, if their distances are below a certain threshold d_g . The distance threshold d_g is required as the points of the current ridges do not exactly match the reference points. Among others, the damage field from which the ridges are extracted is subject to a little drift due to the displacement of the material points.

Next, the selected vertices determine which faces (triangles) F_* are newly added to F_t (w.r.t. the previous time step F_{t-1} , Lines 12–16). For this, we loop over the faces in F_{ref} that have not yet been selected previously (i.e., $F_{\text{ref}} - F_{t-1}$, Line 13), and add new faces f for which all vertices are contained in V (Lines 14 and 15, Figure 5b). Finally, the new set of faces F_* is added to the previous set of faces F_{t-1} to form the face set F_t of the currently considered time step t (Line 16).

6. Visualization of Fracture Progression

To be able to answer research questions regarding fracture progression, we need to not only provide the crack geometry for a single step in time, but to visualize the overall progression of fractures over time. For this, we propose visualization techniques that are based on our temporally stable ridge extraction technique described in the previous section. These techniques support the understanding of how cracks and fractures develop.

6.1. Temporal Development of Fractures

The temporally-stable ridge extraction is done by selecting parts from reference ridges based on their distance to the current ridges. The selected parts are added to a growing ridge representation, that includes the current and all previous ridge states. Temporal development of fracture growth, as shown in Figure 1b, is visualized by storing for each vertex at which point

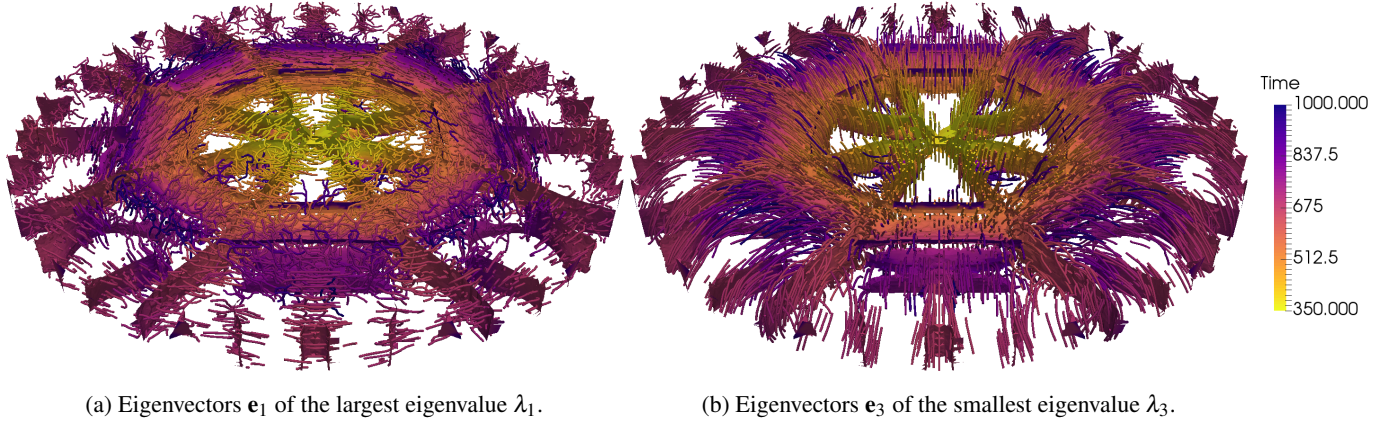


Figure 6: Tensor field lines show the temporal progression of principal stress directions during fracture progression. Time is given in 1×10^{-7} s.

in time it was added to the growing ridge representation. This information is used to visualize the temporal development of the cracks by mapping the obtained time values to color in the reference state, to which the growing process converges. With this, the user can directly see at which point in time a certain ridge fragment was added and can therefore observe the overall temporal development of the cracks for each point in time.

6.2. Fracture Growth Velocity

In order to estimate the fracture growth velocity, we use the information, which parts of the reference ridges were added to the growing ridges. Therefore, we estimate the distances of the crack fronts to the previous fracture geometry of the growing ridges, and divide by the period between temporal states, to obtain the growth velocity of the fracture. As shown in Figure 5b, we first calculate the distance of each new vertex (red) added to the previous growing ridges (green) using the Hausdorff distance measure. The velocity of each crack front is then approximated as the largest distance value per fragment, where the fragments are estimated as the largest connected components, indicated by red triangles in Figure 5b. Special treatment is required for new fragments that are not connected to the previous fracture geometry. Such fragments can appear at relatively large distances to the growing ridge, and lead to unreasonably high velocities. We address this problem by checking whether a new fragment shares at least one vertex with previously added geometry and otherwise define its growth velocity to be zero. Overall, we get an individual velocity approximation for each fragment, as shown in Figure 1c.

In theory, the crack velocity should smoothly increase from rest to the Rayleigh wave speed. In the experiment, the crack accelerates rapidly as it grows, then gets slower and reaches a final velocity lower as the Rayleigh wave speed. These phenomena could be analyzed with our visualization technique and

a domain scientist can get details on how the crack front grows in the simulation.

6.3. Multi-Field Visualization for Fracture Analysis

Domain scientists are interested in obtaining a deeper understanding of the fracture process in peridynamics simulation. This can be achieved by examining physical quantities calculated during the simulation, where a special focus is on how the stress behaves at the surfaces and near the crack front. Therefore, we visualize the stress tensor near the extracted ridge. Figure 6 shows the temporal progression of the principal stress directions near the ridges using tensor field lines. The seed points for the tensor field lines are selected from the set of input data points for the currently processed time step. These data points are filtered by their distance to the new fragments, with an user-defined upper and lower bound. Finally, points that are close to the geometry from the previous step are removed to obtain only those points that lie ahead of the crack front. During the ridge growing process, the tensor field lines seeded at the selected data points in each processed time step are collected to obtain a visualization of the temporal progression of the stress field during the fracture progression.

The ridges of the damage field together with selected levels of the instant von Mises stress are shown in Figure 7. The visualization of stress with the distortion energy theory is one of the most preferred methods for failure analysis in industry. An engineer is validating the design of his object by checking if it withstands a prescribed yield stress value for certain conditions. With the selection of isosurfaces the engineer can focus on the parts of interests and validate the distribution of the von Mises Stress with respect to the prescribed conditions in the specification.

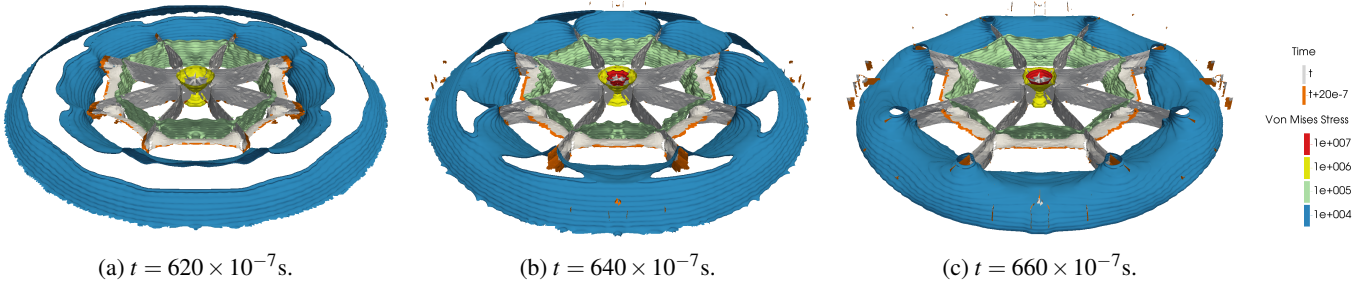


Figure 7: Fracture progression and von Mises stress, indicated through selected isosurfaces in the Plate data set. The highlighted part of the fracture (orange) shows the progression after $\Delta t = 20 \times 10^{-7}$ s.

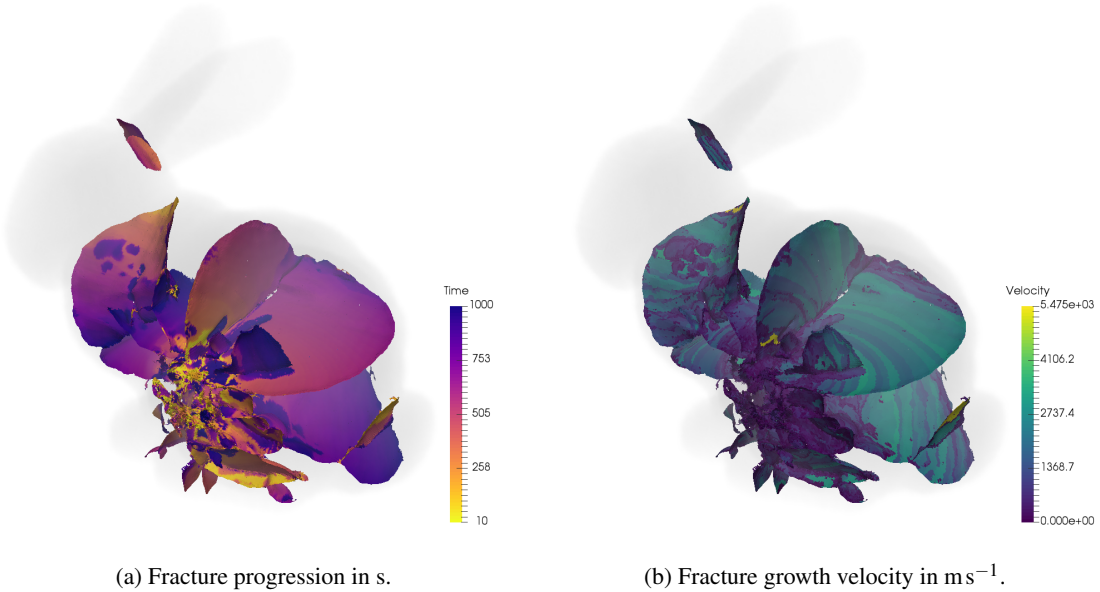


Figure 8: Fracture progression and growth velocity in the Bunny data set at time $t = 1000 \times 10^{-7}$.

7. Results

With the temporally-stable ridge growing process, we are able to estimate additional quantities in the fracture geometry, like temporal progression and fracture growth velocity. Note, that the fracture growth velocity is the velocity at the crack surface and the velocity $\mathbf{v}(t, \mathbf{X})$ is the velocity at the material point \mathbf{X} . Now, we are able to provide the fracture growth velocity, in addition to the velocity provided by the peridynamic simulation. This information allows for visualizing the temporal progression of the whole fracture, e.g., Figure 8a, or the distribution of growth velocities (Figure 9) and the temporal progression of the crack front, e.g., Figure 8b, within a single image.

In Section 5, we discussed temporal instability when extracting ridges in consecutive time steps. We therefore proposed ridge growing to assure temporal stability, which we evaluated with the given data sets in terms of usability and obtained satisfying results, e.g., shown in Figures 8 for the Bunny and 11 for the Kalthoff Winkler data set. Despite flickering and visual clutter in the unfiltered temporal ridges, we were able to obtain temporally stable fracture progression while all important features in the reference ridges are captured.

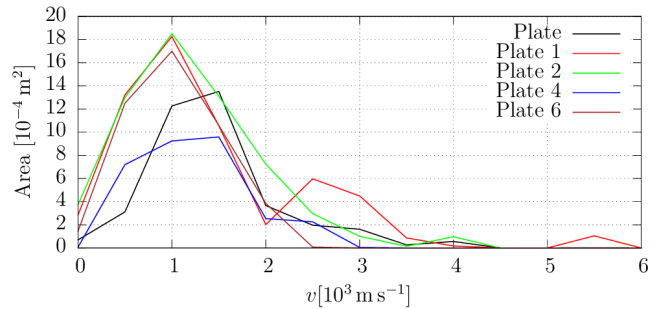


Figure 9: Distribution of growth velocities per area enables comparison of the results for the variations of the Plate data set.

7.1. Data Sets

For the validation of our technique for the visualization of fracture dynamics using height ridge extraction and the stress visualization, we use three example data sets. The first two of them, the plate and the Stanford bunny, were run with the peridynamic package of LAMMPS [56] and the bond-based prototype microelastic brittle (PMB) [41] material model. The Kalthoff Winkler experiment was simulated with Peridigm [57]

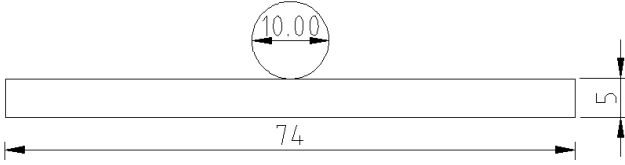


Figure 10: Sketch of the measurements in mm of the plate data set.

and the elastic material model.

7.1.1. Plate Data Set

Figure 10 is a sketch of the geometry of the thin plate, which is hit by a spherical projectile. The thin plate was scattered with 189035 nodes of a regular grid with a spacing of 5×10^{-4} m. The material was modeled with a stiffness constant of $c = 1.6863 \times 10^{22}$, a critical stretch for bond failure of $s_c = 5 \times 10^{-4}$ m and a material density of $\rho = 2200 \text{ kg m}^{-3}$. A spherical projectile hits the plate with a velocity $v = 100 \text{ ms}^{-1}$ and the time step size was set to 5×10^{-7} s. Note, that these material parameters are the initial values for the data set. In Section 7.4, these parameters are changed to study the variation in the fracture progression depending on them.

7.1.2. Stanford Bunny Data Set

For the simulation of the Stanford bunny, we scattered its reconstructed surfaces with 1 787 229 nodes on a regular grid with a spacing of 5×10^{-4} m (Figure 12) and model its material with a bulk modulus K of 210×10^9 Pa, a critical stress intensity factor of K_{Ic} of $2 \times 10^6 \text{ Pa } \sqrt{\text{m}}$, and a material density ρ of 3369 kg m^{-3} . The projectile hits the bunny with a velocity of 100 ms^{-1} and the time step size was set to 10^{-7} s.

7.1.3. Kalthoff Winkler Data Set

The Kalthoff Winkler (KW) experiment is the peridynamics simulation of a cylinder pressing on the upper mid of a plate. Next to the cylinder are two initial cracks to the mid of the height (Figure 13). The plate was scattered with 37962 nodes with an initial spacing of 2.5×10^{-5} m, 5×10^{-5} m, and 1×10^{-4} m in each direction. The material parameters are a bulk modulus K of 14.9×10^9 Pa, a shear modulus μ of 8.94×10^9 Pa, and a material density ρ of 2200 kg m^{-3} . The velocity of the cylinder pressing on the plate is 30 ms^{-1} and 20 ms^{-1} . For the geometry and velocities, we used the same as provided by Raymond et al. [49]. Because of the comparison of our results with the experimental data and the smoothed-particle hydrodynamics simulation (SPH) there in Section 7.8.

7.1.4. Simulation Details

All the simulations were run on the Atacama cluster. Table 2 shows some details of the simulation of all three data sets. All simulations were run on one up to four Dell PowerEdge M620 nodes with 16 Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60 GHz CPU and 64 GB memory per node.

7.2. Parameter Settings and Timings

Table 3 lists the data sets we use for evaluation, along with respective average timings for the different computations steps and parameter settings. For the Plate data set, calculation of the ridges was performed for each time step as described in Section 4, where the gradient and Hessian per material point were calculated using a radius of $d_r = 0.002$ for the least squares fitting. The radius d_r is chosen large enough to obtain stable gradients and as small as possible to not smooth away important information. The tetrahedral grid resulting from the Delaunay triangulation was filtered such that cells with a maximum edge length above $d_e = 0.002$ were removed. This threshold is defined interactively by the user such that the surface of the object is clearly visible and no holes appear at the fractured areas. The possibility of choosing thresholds interactively is requested by application scientists, as an automatic choice might hide important and unexpected information, for example effects caused by the modeling approach and its parameters. During the calculation of the ridge geometry with marching tetrahedra, nodes with a damage value $c(t, \mathbf{X})$ below 0.1 are skipped and no triangles are generated for these nodes. This damage threshold was used for all generated ridges on all data sets, as it is low enough that no ridge that describes a possible fracture is missed but visual clutter is reliably removed. For the variations of the Plate data set shown in Figure 14, the ridges were all calculated with the same parameters except for the maximum edge length which was set to $d_e = 0.0015$ for Plate 1, 4 and 6.

The ridges in the Bunny data set, shown in Figure 8, were also generated with $d_r = 0.002$ and $d_e = 0.0015$, while we used $d_r = 0.004$ and $d_e = 0.005$ for the Kalthoff Winkler data set, shown in Figure 11.

The geometry of each selected reference time step was filtered and small fragments were removed. This step is interactively performed by the user, to remove all fragments that do not describe fractures and to get a clear depiction of the fractures of interest. Note, that this filter is only applied to clean up the reference state, but not the ridges of the previous time steps, as it could potentially remove small fragments of arising fractures that would be missed in the fracture progression calculation, as discussed in Section 5. For the ridge growing process, we used a distance threshold $d_g = 0.0005$ for the Plate and its variations, 0.0001 for the Bunny, and 0.002 for the Kalthoff Winkler case.

As discussed in Section 5, we consider the last time step of a series to be reference time step. In the Bunny and the Kalthoff Winkler data sets, we used the last time step of the simulation

Table 2: Simulations details for all three data sets.

	Plate	Bunny	KW
Material	PMB	PMB	Elastic
Time step size [s]	5×10^{-7}	10^{-6}	1.6×10^{-9}
Time steps	1000	1000	199
Number of nodes	189035	1 787 229	1 616 040
Nodal spacing [m]	5×10^{-4}	5×10^{-4}	2.5×10^{-5}
Computation time [h]	0.18	48	3
Cores	16	64	16
Simulation Tool	LAMMPS	LAMMPS	Peridigm

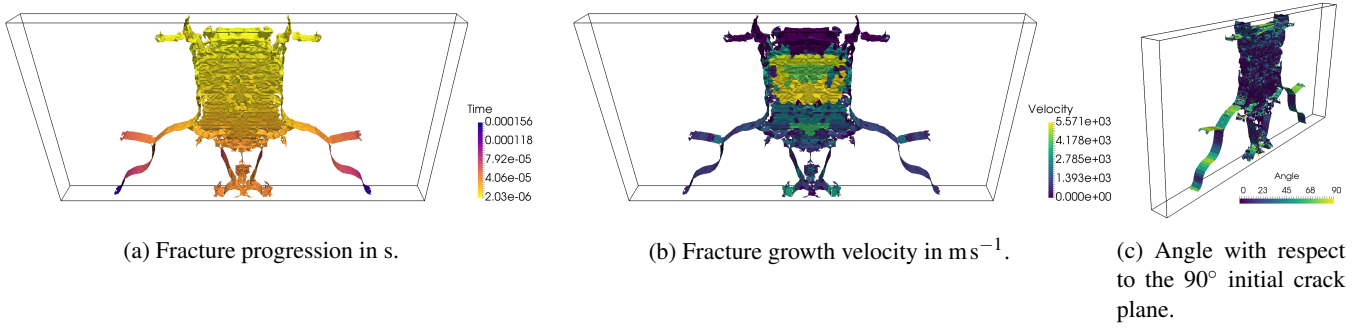


Figure 11: Fractures in the Kalthoff Winkler data set at time $t = 1.58421 \times 10^{-4}$ s. In (c), the camera position was changed to highlight the angle.

Table 3: Calculation timings (in seconds), and employed parameter settings for the shown data sets (“–” denotes “same as above”). Timings for the least squares, tetrahedralization, and ridge calculation are on average per time step, while timings for the ridge growing are the sum over all processed time steps. Measurements were taken on a Core i7-4770 CPU @ 3.40 GHz.

Data set	#Steps	LSq. (\emptyset)	Tet. (\emptyset)	Ridge (\emptyset)	Grow (Σ)	d_e	d_r	d_g	$\langle c(t, \mathbf{X}) \rangle$
Plate	100	2.981	1.207	0.843	412.2	0.002	0.002	0.0005	0.1
Plate 1	60	2.813	1.399	0.921	440.5	0.0015	–	–	–
Plate 2	60	2.801	1.410	0.912	319.3	0.002	–	–	–
Plate 4	80	3.172	1.636	0.925	529.7	0.0015	–	–	–
Plate 6	100	3.126	1.649	0.900	594.9	–	–	–	–
Bunny	100	45.037	57.911	8.255	1785.5	–	–	0.0001	–
KW	100	0.192	0.287	0.136	72.3	0.005	0.004	0.002	–



Figure 12: Plot of the peridynamic material points of the bunny. The size of the scattered bunny is $[-0.04, 0.04] \text{ m} \times [-0.03, 0.04] \text{ m} \times [-0.06, 0] \text{ m}$.

as reference. However, the considered time range is not necessarily defined by the simulation, but a naturally a scientist can define the last time step of interest. This can be useful when the fracture growth has come to a complete stop early, or when after a certain step the structure starts to degrade significantly. For instance, Figures 1b and 1c show the Plate data set at time $t = 800 \times 10^{-7}$ s. For the variations of the Plate (Figure 14) the fracture growth behaves differently, and different final time steps were chosen (cf. Table 3, “#Steps”).

Average timings for the different calculation steps per time step are given in Table 3. The variations in the timings of the ridge and tetrahedral grid calculations are due to the different shapes and numbers of nodes, which leads to varying numbers

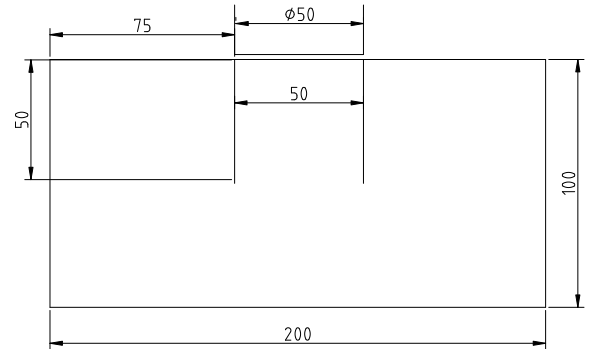


Figure 13: Sketch of the Kalthoff Winkler experiment. Distances are in mm.

of tetrahedral cells. The timing of the gradient and Hessian calculation depends mostly on the number of nodes and the choice of the radius d_r . Calculation timings for the complete ridge growing process over all time steps vary with the selected reference time step and the size of the data sets.

7.3. Thin Plate

Figure 1b shows the temporal development of the fractures in the Plate data set. The projectile hits the plate in the center from above and our visualization shows that the fractures begin at the point of impact (blue color) and then grow quickly toward the opposite site, while the crack fronts move steadily toward the boundary of the plate.

Our visualization of the growth velocity (Figure 1c) shows that the cracks grow steadily in the center area of the plate with only small variations. The velocity of the crack fronts peaks

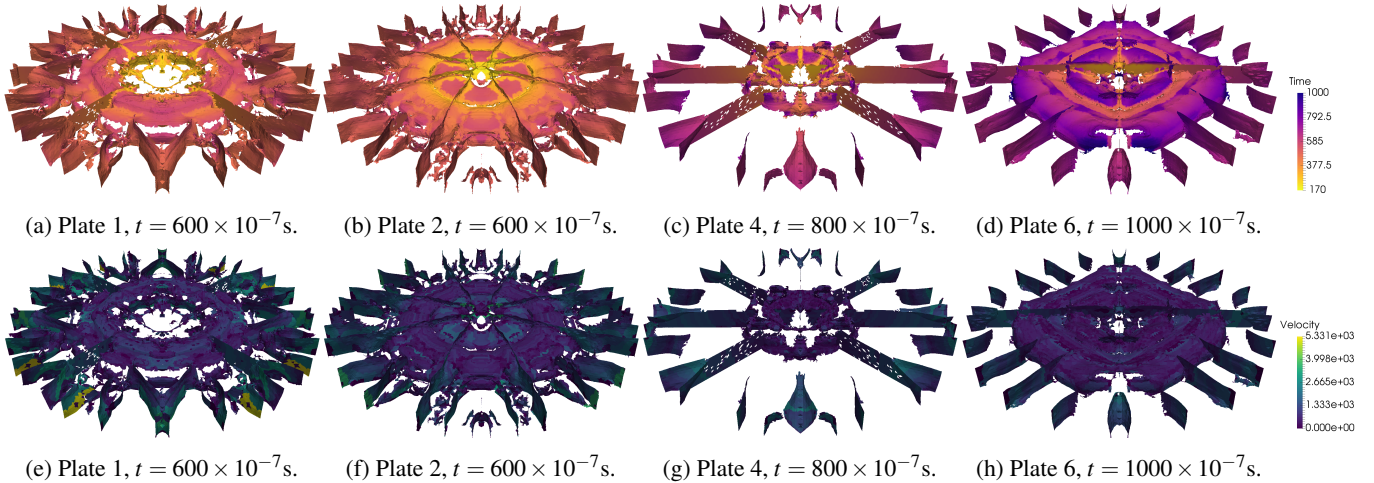


Figure 14: Visualization of fracture dynamics in variations of the Plate data set. Time is given in s and velocity in ms^{-1} .

shortly before they approach the boundary, where velocity decreases again. The figure also shows that the crack front is not straight such that fractures appear earlier at the top of the plate than at the bottom. Just before the crack front reaches the boundary, additional crack fronts appear in this area and grow toward the center. Shortly after the initial crack, a horizontal fracture occurs which grows much slower than the central cracks. This horizontal fracture could not be observed with previous techniques, like the direct rendering of the data set using spheres. This general fracture behavior can also be observed in the variations of the data set as shown in Figure 14. Differences can be observed in the number of fractures as well as the time it takes for them to form. The horizontal fracture is observable in all variations but differs in growth velocity and size.

7.4. Variations of the Thin Plate

Figure 14 shows the fracture dynamics for different variations of the material parameters of the bond-based peridynamic PMB material model for the same geometry of the plate and the spherical projectile (see Section 7.1.1). Table 4 shows the parameter space for the variation of the thin plate for this small study. The parameter $\alpha \in [0, 1)$ controls the elasticity of the material. The influence of the parameter to the simulation is studied with uncertainty quantification by Franzelin et al. [58] with the result that for $\alpha = 0.25$ the simulation is stable with respect to the discretization. From the uncertainty quantification the domain scientist receives a single value or range where his quantity of interest is stable with respect to the discretization. With our new technique the domain scientist can compare

Table 4: Variation of the parameter space for the thin plate in this small study.

Figure	α	Force density in Nm^{-2}
1b	0.25	1×10^{16}
14a	0.50	1×10^{16}
14b	0.75	1×10^{16}
14c	0.25	1×10^{16}
14d	0.25	1×10^{17}

the unstable with stable fracture dynamics visually and investigate where the instability arises. Figure 1b shows the simulation for $\alpha = 0.25$, Figure 14a for a $\alpha = 0.5$, and Figure 14b for $\alpha = 0.75$. For Figure 14a the ridges develop from the center to the boundary (long straight ridges) and get reflected there and then grow back to the center (small ridges left and right to the long ridges). For $\alpha = 0.75$ in Figure 14b, we see the same long ridges from the center to the boundary. In comparison to Figure 1b, some ridges from the center start to bifurcate before hitting the boundary and the reflected ridges at the boundary are longer. The orbital ridge starts closer to the center and is consistent until the point where the reflected ridge hits the orbital ridge. For Figure 14a, we see that it has more similarity with Figure 14a. In comparison, the shape of the reflected ridges is different and the long ridges do not process directly from the center and arise later. For Figure 14c and Figure 14d where the force of the spherical projectile is changed, we see that the reflected ridges are longer and the orbital ridge is additional. For the velocities in Figure 14g and Figure 14h, we see that the velocities are similar. The extracted ridges give more insight in the behavior of the stability as the single scalar value from uncertainty quantification. Thus, our tool provides more complex details of fracture progression, but a detailed discussion with respect to material science would be out of the scope of this paper.

7.5. Kalthoff Winkler Experiment

In the Kalthoff Winkler data set (Figure 11), our ridge extraction technique clearly shows the temporal progression of the two crack fronts that grow from the center toward the lower edges of the block. The growth velocity of the crack fronts is steady with peaks in the center area. The fractures in the Bunny data set (Figure 8) form initially at the impact point and then progress radially through the object. Our visualization technique shows that most fractures grow steadily while some fractures only appear some time after the impact and are even unconnected to previous fractures. The crack fronts are evenly shaped with no preferred growth direction. In all examined data

sets we could observe similar velocities for the fracture growth, which are in average around 1×10^3 to $2 \times 10^3 \text{ ms}^{-1}$ with few high peaks, that may be due to deviations of the temporal ridges to the reference state. The observed velocities are in the same ranges as obtained in experiments [59, 60]. Thus, the values of the fracture growth velocity seem to be physically credible.

7.6. Stanford Bunny

With the Stanford bunny we apply our visualization technique to a complex geometry. Figure 3b shows the visualization of the scattered nodes of the Stanford bunny with colored spheres. Here at the mid of the bunny where the spherical projectile impacted, there are plenty of red spheres. At the ears, at the neck, and the throat of the bunny there are small red lines which can be interpreted as cracks. First, the analysis of the cracks in this complex geometry is difficult, because the domain scientist cannot see what is happening in the inside of the bunny. Second, it is hard to decide where the crack surfaces are located between the red line of particles at the ears, or the neck, or the throat of the bunny. With the volume rendering of the node-based data and the visualization of the extracted ridges with our technique shown in Figure 8, a more global view of the development of the crack is provided. Thus, the domain scientists see the shape of the bunny with the volume rendering technique and the fracture progression via the extracted crack surfaces. With the fracture progression and the fracture growth velocity the behavior of crack and fractures, even in complex geometries are feasible. Note, that the variation of the plate was used to study the influence of parameters and the Kalthoff Winkler experiment for the comparison with experimental data. The Bunny data set was used to validate our visualization approach for complex geometries rather than the interpretation of the results with respect to experimental data or other simulation techniques.

7.7. Multi-Field Visualization for Fracture Analysis

Figure 7 shows the combination of the fracture progression and the von Mises stress. With the selected isosurfaces of the von Mises stress we see how the stress at the crack tip behaves. For the blue part of the stress, we see a blue inner and outer isosurface at $t = 620 \times 10^{-7} \text{ s}$. After $20 \times 10^{-7} \text{ s}$, the inner and outer isosurfaces are connected and around the main crack tips we observe a shape looking like a sombrero. After additional $20 \times 10^{-7} \text{ s}$ small inclusions around the main crack tips occur and the isosurface at the boundary of the geometry looks much smoother.

Figure 6 shows the principal stress directions \mathbf{e}_1 and \mathbf{e}_3 for the largest eigenvalue λ_1 and the smallest eigenvalue λ_3 . In Figure 6a, we see that the directions of the eigenvectors of the largest eigenvalue λ_1 (the maximal principle stress) are orthogonal to the extracted ridges of the main crack branches. Figure 6b shows the direction of the eigenvectors of the smallest eigenvalues λ_3 (the minimal principle stress). These are aligned to the extracted surfaces of the ridges. From a physical point of view this behavior of the minimal principal stress and the maximal principal stress is plausible. The visualization of these

values could be applied to understand how cracks bifurcate and to obtain a better understanding of the peridynamics theory.

7.8. Verification with Experimental Data

In Sections 7.3–7.7 we focused on the interpretation of the results with respect to the new features our tool provides. Here, we focus on the comparison to experimental data and other simulation approaches. The reference for the comparison is the Kalthoff Winkler experiment. This experiment is standard problem in dynamic fractures [49, 3]. Raymond et al. [49] simulated the KW experiment with smoothed-particle hydrodynamics and compared with the experimental results by Kalthoff and Winkler [61] (Table 5). The provided geometry, materials parameters, and parameters for the experiment were used to set up the simulations with Peridigm [57]. Note that for the SPH results, we do not know how and where the fracture growth velocity was measured.

For our experiments, we have evaluated the fracture growth velocity based on the extracted values of our new analysis. As it can be observed in Figure 14b for a KW experiment with different parameter settings, there are several types of cracks. The micro-fractures in the completely damaged region of impact in the middle leads to a vast mix of low and high local velocities. The fracture growth velocity, however, should be measured at the main fractures branching off. We therefore excluded the part of the simulation domain below the cylinder that indents the plate and where the plate is completely shattered. We then computed the average fracture growth velocity over the last time steps, more precisely the last 18 % of the overall simulation time. This restricts the simulation to the last major parts of the main cracks until the end of the simulation, where they reach the domain boundary (cf. Figure 14b).

The average velocity that we measure is 1142 m/s, the median 1144 m/s. This matches very well the results obtained both in experiments and with SPH. Figure 15 shows the histogram of the fracture growth velocity of the Kalthoff Winkler experiment. We can observe that we have small cracks (or extensions of the large cracks) with distinct low and high velocities, which indicates that peridynamics might favor certain velocities. But the main velocities can be measured around the peak at $\approx 1140 \text{ m/s}$.

Our new analysis methodology allows us to study such simulation-dependent properties more in depth than possible before. For example, we observe a short peak of crack propagation velocities in the KW experiment of a non-physical velocity of around 5000 m/s after 47 % of the simulation time. There, two distinct vertical cracks evolve which are well-aligned to the

Table 5: Fracture growth velocity obtained by experiment by Kalthoff and Winkler [61], smoothed-particle hydrodynamics (SPH) simulation[49], and our peridynamics (PD) results.

Setting	Fracture growth velocity in ms^{-1}
Experiment	1000
SPH	1200
PD	mean: 1142, median: 1144

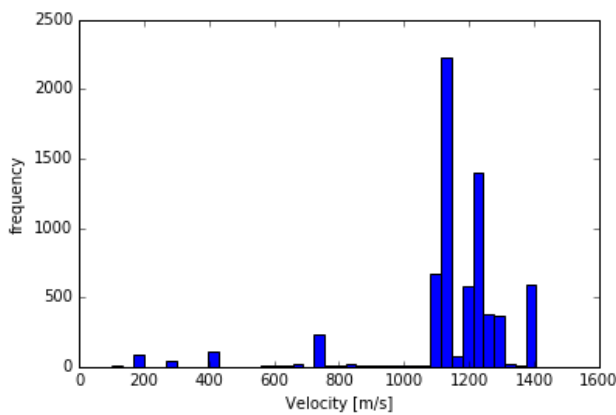


Figure 15: Histogram of the fracture growth velocity over the last 18 % of the simulation time of the Kalthoff Winkler experiment with an initial velocity of 20 ms^{-1} for the cylinder setup. The scenario excludes the inner part of the domain below the cylinder. There, the plate is completely destroyed.

grid structure of the particles, connecting several horizontally-aligned cracks. This gives strong evidence that the initial distribution of the nodes on a Cartesian, crystal-like grid has significant (and, depending on the material, possibly non-physical) influence on the simulation results. A preference for axis-aligned cracks can also be observed visually in most visualizations in this paper. However, a sound analysis will require more study with respect to the peridynamics model in this regard.

8. Conclusion

In this paper, we gave a brief introduction to peridynamics simulation and presented our technique for the extraction of fracture geometry from node-based peridynamics data sets using height ridges. The ridges from each time step are calculated directly on the tetrahedralized input data based on marching tetrahedra. Based on the result of ridge extraction, we proposed a technique for obtaining temporally stable ridge growth by progressively combining temporal ridges into a reference state. We showed how this growth process can be used for the visualization of temporal fracture progression and depiction of growth velocity by combining them into a single spatiotemporal state. We also examined the physical mechanisms of the peridynamics simulations in terms of von Mises stress and principal stress directions to reveal fracture dynamics, and evaluated our approach and demonstrated its utility by means of different data sets. Finally, our new approach enabled us to compare against experiments and simulation results in the literature. It furthermore builds the basis for further studies of the effects of the discrete modeling approach using peridynamics.

For future work, we would like to evaluate our approach on a larger variety of peridynamics data sets. We observed a small drift of the material points in the evaluated data sets, which our temporally-stable ridge extraction approach was able to handle. But this may become an issue if the drift gets larger in later time steps. A possible approach to handle this is to dynamically morph the reference ridge accordingly. We observed

some increased calculation timings for the ridge growing process for large data sets, like the Bunny, due to complex ridge geometry. Therefore, we would like to investigate strategies for performance optimization, such as parallelization.

References

- [1] Silling S. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids* 2000;48(1):175–209.
- [2] Bobaru F, Ha Y, Hu W. Damage progression from impact in layered glass modeled with peridynamics. *Open Engineering* 2012;2(4):551–61.
- [3] Silling SA. Peridynamic modeling of the Kalthoff-Winkler experiment. Tech. Rep.; Sandia National Laboratory; 2002.
- [4] Diehl P, Schweitzer MA. Simulation of wave propagation and impact damage in brittle materials using peridynamics. In: *Recent Trends in Computational Engineering – CE2014. Lecture Notes in Computational Science and Engineering*; Springer; 2015.
- [5] Diehl P, Franzelin F, Pflüger D, Ganzenmüller GC. Bond-based peridynamics: a quantitative study of Mode I crack opening. *International Journal of Fracture* 2016;:1–14.
- [6] Furst JD, Pizer SM. Marching ridges. In: *SIP*. 2001, p. 22–6.
- [7] Barakat S, Tricoche X. An image-based approach to interactive crease extraction and rendering. *Procedia Computer Science* 2010;1(1):1709 – 18.
- [8] Haralick RM. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing* 1983;22(1):28–38.
- [9] Eberly D. *Ridges in image and data analysis*; vol. 7. Springer Science & Business Media; 1996.
- [10] Lindeberg T. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision* 1998;30(2):117–56.
- [11] Schultz T, Theisel H, Seidel HP. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Transactions on Visualization and Computer Graphics* 2010;16(1):109–19.
- [12] Sadlo F, Peikert R. Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Transactions on Visualization and Computer Graphics* 2007;13(6):1456–63.
- [13] Peikert R, Sadlo F. Height ridge computation and filtering for visualization. In: *Proceedings of IEEE Pacific Visualization Symposium (PacificVIS)*. 2008, p. 119–26.
- [14] Kindlmann G, Tricoche X, Westin CF. Anisotropy creases delineate white matter structure in diffusion tensor MRI. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*. Springer; 2006, p. 126–33.
- [15] Süßmuth J, Greiner G. Ridge based curve and surface reconstruction. In: *Proceedings of the fifth Eurographics symposium on Geometry processing*. Eurographics Association; 2007, p. 243–51.
- [16] Wu J, Dick C, Westermann R. Interactive high-resolution boundary surfaces for deformable bodies with changing topology. In: *Proceedings of 8th Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)* 2011. 2011, p. 29–38.
- [17] Bachthaler S, Sadlo F, Dachsbacher C, Weiskopf D. Space-time visualization of dynamics in Lagrangian coherent structures of time-dependent 2D vector fields. In: *Proceedings of International Conference on Information Visualization Theory and Applications (IVAPP)*. 2012, p. 573–83.
- [18] Heßel S, Fernandes O, Boblest S, Offenhäuser P, Hoffmann M, Beck A, et al. Visualization of 2D wave propagation by Huygens’ principle. In: *Proceedings of Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)*. 2015, p. 19–28.
- [19] Terzopoulos D, Fleischer K. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In: *ACM SIGGRAPH Computer Graphics*; vol. 22. ACM; 1988, p. 269–78.
- [20] O’Brien JF, Hodgins JK. Graphical modeling and animation of brittle fracture. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH ’99*; New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5; 1999, p. 137–46.
- [21] Desbrun M, Gascuel MP. Animating soft substances with implicit surfaces. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM; 1995, p. 287–90.

- [22] Müller M, Keiser R, Nealen A, Pauly M, Gross M, Alexa M. Point based animation of elastic, plastic and melting objects. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association; 2004, p. 141–51.
- [23] Pauly M, Keiser R, Adams B, Dutré P, Gross M, Guibas LJ. Meshless animation of fracturing solids. In: ACM Transactions on Graphics; vol. 24. ACM; 2005, p. 957–64.
- [24] Hahn D, Wojtan C. High-resolution brittle fracture simulation with boundary elements. ACM Transactions on Graphics 2015;34(4):151:1–151:12.
- [25] Hirota K, Tanoue Y, Kaneko T. Simulation of three-dimensional cracks. The Visual Computer 2000;16(7):371–8.
- [26] Levine JA, Bargteil AW, Corsi C, Tessendorf J, Geist R. A peridynamic perspective on spring-mass fracture. In: Symposium on Computer Animation. Citeseer; 2014, p. 47–55.
- [27] Watcharopas C, Sapra Y, Geist R, Levine JA. Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14–16, 2015, Proceedings, Part I. Cham: Springer International Publishing. ISBN 978-3-319-27857-5; 2015, p. 82–91.
- [28] Wu J, Westermann R, Dick C. Physically-based Simulation of Cuts in Deformable Bodies: A Survey. In: Lefebvre S, Spagnuolo M, editors. Eurographics 2014 - State of the Art Reports. The Eurographics Association; 2014, doi:\bibinfo{doi}{10.2312/egst.20141033}.
- [29] Fang Z, Möller T, Hamarneh G, Celler A. Visualization and exploration of time-varying medical image data sets. In: Proceedings of Graphics Interface 2007. GI '07; New York, NY, USA: ACM. ISBN 978-1-56881-337-0; 2007, p. 281–8.
- [30] Lee TY, Shen HW. Visualizing time-varying features with TAC-based distance fields. In: Visualization Symposium, 2009. PacificVis '09. IEEE Pacific. 2009, p. 1–8.
- [31] Wang C, Yu H, Ma KL. Importance-driven time-varying data visualization. IEEE Transactions on Visualization and Computer Graphics 2008;14(6):1547–54.
- [32] Schneider D, Wiebel A, Carr H, Hlawitschka M, Scheuermann G. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. IEEE Transactions on Visualization and Computer Graphics 2008;14(6):1475–82.
- [33] Widanagamaachchi W, Christensen C, Bremer PT, Pascucci V. Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In: Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on. 2012, p. 9–17.
- [34] Frey S, Sadlo F, Ertl T. Visualization of temporal similarity in field data. IEEE Transactions on Visualization and Computer Graphics 2012;18:2023–32.
- [35] Joshi A, Rheingans P. Evaluation of illustration-inspired techniques for time-varying data visualization. Computer Graphics Forum 2008;27(3):999–1006.
- [36] Lu A, Shen HW. Interactive storyboard for overall time-varying data visualization. In: Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific. 2008, p. 143–50.
- [37] Bach B, Dragicevic P, Archambault D, Hurter C, Carpendale S. A review of temporal data visualizations based on space-time cube operations. In: EuroVis - STARS. The Eurographics Association. ISBN -; 2014.
- [38] Woodring J, Wang C, Shen HW. High dimensional direct rendering of time-varying volumetric data. In: Visualization, 2003. VIS 2003. IEEE. 2003, p. 417–24.
- [39] Lee TY, Shen HW. Visualization and exploration of temporal trend relationships in multivariate time-varying data. IEEE Transactions on Visualization and Computer Graphics 2009;15(6):1359–66.
- [40] Callahan SP, Freire J, Santos E, Scheidegger CE, Silva CT, Vo HT. Vistrails: Visualization meets data management. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. SIGMOD '06; New York, NY, USA: ACM. ISBN 1-59593-434-0; 2006, p. 745–7.
- [41] Silling S, Askari E. A meshfree method based on the peridynamic model of solid mechanics. In: Elsevier, editor. Computer & Structures; vol. 83. 2005, p. 1526–35.
- [42] Littlewood DJ. Simulation of dynamic fracture using peridynamics, finite element modeling, and contact. In: ASME 2010 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers; 2010, p. 209–17.
- [43] Liu W, Hong JW. A coupling approach of discretized peridynamics with finite element method. Computer Methods in Applied Mechanics and Engineering 2012;245:245–246:163–75.
- [44] Madenci E, Oterkus E. Coupling of the Peridynamic Theory and Finite Element Method. New York, NY: Springer New York. ISBN 978-1-4614-8465-3; 2014, p. 191–202.
- [45] Kilic B, Madenci E. Coupling of peridynamic theory and the finite element method. Journal of Mechanics of Materials and Structures 2010;5(5):707–33.
- [46] Schweitzer MA, Wu S. A moving least squares approach to the construction of discontinuous enrichment functions. In: Griebel M, editor. Singular Phenomena and Scaling in Mathematical Models. Springer International Publishing. ISBN 978-3-319-00785-4; 2014, p. 347–60.
- [47] Schweitzer MA, Wu S. Numerical Integration of On-the-Fly-Computed Enrichment Functions in the PUM. Cham: Springer International Publishing. ISBN 978-3-319-06898-5; 2015, p. 247–67.
- [48] Üffinger M, Schweitzer MA, Sadlo F, Ertl T. Direct visualization of particle-partition of unity data. In: Proceedings of International Workshop on Vision, Modeling and Visualization (VMV). 2011, p. 255–62.
- [49] Raymond S, Lemiale V, Ibrahim R, Lau R. A meshfree study of the Kalthoff–Winkler experiment in 3D at room and low temperatures under dynamic loading using viscoplastic modelling. Engineering Analysis with Boundary Elements 2014;42:20–5. Advances on Meshfree and other Mesh reduction methods.
- [50] Fineberg J, Marder M. Instability in dynamic fracture. Physics Reports 1999;313(1–2):1–108.
- [51] Kobayashi A, Ohtani N, Sato T. Phenomenological aspects of viscoelastic crack propagation. Journal of Applied Polymer Science 1974;18(6):1625–38.
- [52] Akio D, Koide A. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. IEICE Transactions on Information and Systems 1991;74(1):214–24.
- [53] Weller F. Stability of voronoi neighborhood under perturbations of the sites. In: CCCG. 1997, URL <http://dblp.uni-trier.de/db/conf/cccg/cccg1997.html#Weller97>.
- [54] Reem D. The geometric stability of voronoi diagrams with respect to small changes of the sites. In: Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry. SoCG '11; New York, NY, USA: ACM. ISBN 978-1-4503-0682-9; 2011, p. 254–63. doi:\bibinfo{doi}{10.1145/1998196.1998234}. URL <http://doi.acm.org/10.1145/1998196.1998234>.
- [55] Si H. TetGen, a Delaunay-based quality tetrahedral mesh generator. ACM Trans Math Softw 2015;41(2):11:1–11:36.
- [56] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. Journal of Computational Physics 1995;117(1):1–19.
- [57] Parks ML, Littlewood D, Mitchell JA, Silling SA. Peridigm users' guide. Tech. Rep. SAND2012-7800; Sandia National Laboratories; 2012.
- [58] Franzelin F, Diehl P, Pflüger D. Non-intrusive uncertainty quantification with sparse grids for multivariate peridynamic simulations. In: Meshfree Methods for Partial Differential Equations VII; vol. 100 of *Lecture Notes in Computational Science and Engineering*. Springer; 2014.
- [59] Bowden FP, Brunton JH, Field JE, Heyes AD. Controlled fracture of brittle solids and interruption of electrical current. Nature 1967;216:38–42.
- [60] Riedel W, Hiermaier S, Thoma K. Transient stress and failure analysis of impact experiments with ceramics. In: Materials Science and Engineering B; vol. 173. Elsevier; 2010, p. 139–47.
- [61] Kalthoff J, Winkler S. Failure mode transition at high rates of shear loading. In: Kunze H, Meyer L, editors. CHiem CY; vol. 1. DGM Informationsgesellschaft Verlag; 1988, p. 185–95.