PREPRINT 447

FEBRUARY 2016

Mathematical modeling of ceramic bond bridges in grinding wheels

Michael Rom, Karl-Heinz Brakhage, Sebastian Barth,
Christian Wrobel, Patrick Mattfeld and Fritz Klocke

Institut für Geometrie und Praktische Mathematik
Templergraben 55, 52062 Aachen, Germany

Michael Rom, Karl-Heinz Brakhage
Institute for Geometry and Applied Mathematics (IGPM), RWTH Aachen, Templergraben 55, 52056 Aachen, Germany
{rom,brakhageg}@igpm.rwth-aachen.de
Sebastian Barth, Christian Wrobel, Patrick Mattfeld, Fritz Klocke
Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen, Steinbachstr. 19, 52074 Aachen, Germany
{fs.barth,c.wrobel,p.mattfeld,f.klockeg}@wzl.rwth-aachen.de

Mathematical modeling of ceramic bond bridges in grinding wheels

Michael Rom, Karl-Heinz Brakhage
Institute for Geometry and Applied Mathematics (IGPM),
RWTH Aachen, Templergraben 55, 52056 Aachen, Germany
{*rom,brakhage*}*@igpm.rwth-aachen.de*
Sebastian Barth, Christian Wrobel, Patrick Mattfeld, Fritz Klocke
Laboratory for Machine Tools and Production Engineering (WZL),
RWTH Aachen, Steinbachstr. 19, 52074 Aachen, Germany
{*s.barth,c.wrobel,p.mattfeld,f.klocke*}*@wzl.rwth-aachen.de*

Abstract

Ceramic-bonded grinding wheels with cubic boron nitride (CBN) as grain material belong to the most efficient grinding tools available. They feature a high hardness combined with a high thermal stability and the applicability to grinding ferrous materials. However, the appropriate volumetric composition of grain and bonding material is an expensive and time-consuming process based on experience. Our objective is the mathematical modeling of grinding wheel structures for the prediction of compositions which fulfill given grinding requirements such that using trial and error methods can be avoided. For this purpose, we focus on a three-dimensional element of a grinding wheel which we call volumetric structure element. In this paper, we briefly describe our overall modeling approach and present in detail how we model the ceramic bond. For the latter, we combine analytical and discrete calculations, embedded into an iterative algorithm which ensures to meet bond volume fractions prescribed by grinding wheel specifications.

Keywords: Ceramic Bond Bridges, Vitrified Grinding Wheels, Mathematical Modeling, Cubic Boron Nitride (CBN)

1 Introduction

The working principle of a grinding wheel is illustrated in Fig. 1. The surface of a workpiece is moved along a rotating grinding wheel such that chips are removed from the workpiece by the abrasive grains. Conventional materials for the grains are corundum and silicon carbide. Cubic boron nitride (CBN) and diamond have a higher degree of hardness and are called superabrasive. The grains can be bonded by using vitrified (ceramic), resin (polymer) or metallic bonding systems. The illustration depicts a ceramic-bonded grinding wheel in which the grains are held together by bond bridges. Extensive information on grinding can be found in [16].

There are several advantages of using grinding wheels composed of CBN grains and ceramic bond. CBN is nearly as hard as diamond, but it has a higher thermal stability and it is applicable to grinding ferrous materials [12]. The higher porosity of ceramic-bonded wheels compared with wheels bonded by resin leads to a better transport of cooling lubricant such that thermal stresses on the workpiece are reduced. In addition, the porosity allows for efficient chip removal. On the other hand, a disadvantage of ceramic bond is its brittleness, resulting in a higher risk of bond fracture.

The grinding result, i.e., the final workpiece surface quality, is determined by the grinding wheel topography crucially. The latter depends on the composition of grain and bonding materials. Hence, in our modeling approach we take into consideration a volumetric structure element of a grinding wheel. Our objective is the prediction of the volumetric composition of the grinding wheel components such that grinding requirements can be met without using expensive trial and error methods. Basic concepts and algorithms regarding the implementation of grain shapes obtained from computer tomography scans into the model and the collision-free translation of grains in three-dimensional space can be found in [5]. An overview of further developments and implementations such as grain modeling, grain compaction by rotations and translations, the tetrahedral meshing of a volumetric structure element or ceramic bond modeling is given in [19].

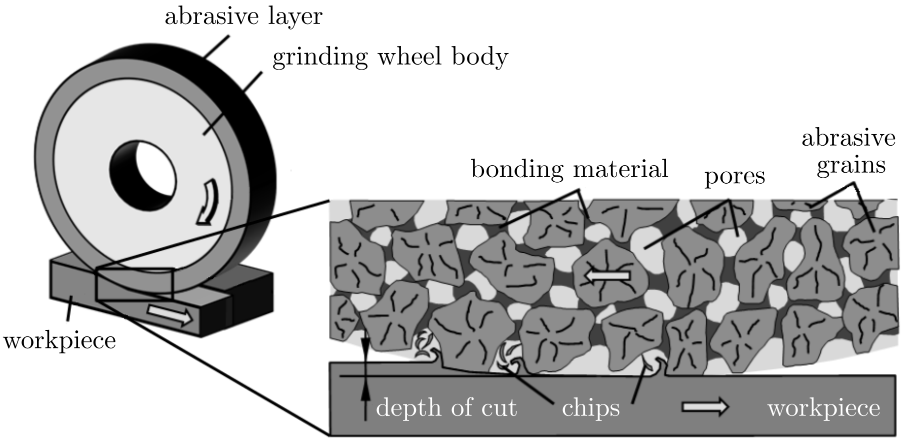

Figure 1: Working principle of a grinding wheel.

Previous works in the context of modeling and simulation of grinding processes mostly concentrate on the grinding wheel topography only, cf. [23, 8, 6]. In a more recent approach, Schumann et al. [20] have developed a more detailed model: CBN-like grains are distributed on the wheel surface randomly such that grains can be intersected during a simulated cutting process to obtain a realistic representation of the topography. However, they do not take into account the grinding wheel structure farther away from the wheel surface.

Models for ceramic bond bridges are scarce and simplified to a large extent. Yamaguchi et al. [24] model grains and bond as spherical particles that stick together. The bond spheres are then replaced by disks of different size to obtain bond bridges. In an approach to model the fracture wear in vitrified CBN grinding wheels with finite elements by Jackson [11], a single grain with an ideal wedge-shaped cutting point connected to two bond bridges is modeled in two dimensions for a stress analysis.

For the modeling of ceramic bond bridges, we have developed an iterative algorithm. It combines analytical and discrete computations and converges to a prescribed bond volume fraction. Based on the grains in the volumetric structure element which are represented by triangle meshes, a tetrahedral mesh is generated for the whole element. Subsequently, adjacent grains are connected by approximations of minimal surfaces which model the bond bridges. These surfaces can be triangulated for visualization purposes easily. The tetrahedral cells which are enclosed by the bond bridge surfaces are determined as bond cells.

We have implemented our algorithms for modeling grinding wheel structures as a plug-in for OpenFlipper [17] which is an open source software framework for geometry processing. With OpenMesh [4] and OpenVolumeMesh [14], it provides efficient halfedge based data structures for polygonal and polyhedral meshes, respectively.

The present paper is structured as follows: Section 2 gives an overview of our overall approach for the modeling of grinding wheel structures. Our bond modeling algorithm is described in Sect. 3 in detail. In Sect. 4, we present volumetric structure elements resulting from our algorithm and a qualitative comparison of a slice through a modeled element with a structure cut of a grinding wheel specimen recorded by light-optical microscopy. Finally, a conclusion and an outlook to future work are given in Sect. 5.

2 Mathematical modeling of grinding wheel structures

In this section, we briefly describe our approach of mathematically modeling grinding wheel structures. The starting point is the choice of the grains. The first option is to use geometries from computer tomography scans of CBN grains, meshed by the *Marching Cubes* algorithm [15] and decimated by a method by Kobbelt et al. [13]. The mesh decimation is necessary to reduce the computation time of geometric calculations such as collision detection. For a simplification of the development of the model and for even faster geometric computations, we use simpler convex polyhedra as a second option for the grains. These polyhedra are based on the platonic solids tetrahedron, octahedron, hexahedron and icosahedron. They are modified by randomly cutting off corners or

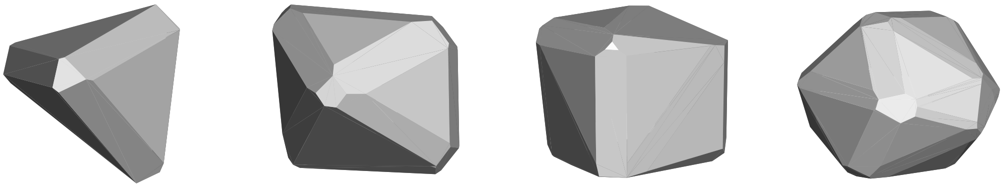

Figure 2: Examples for simplified convex grains based on tetrahedron, octahedron, hexahedron and icosahedron (from left to right).

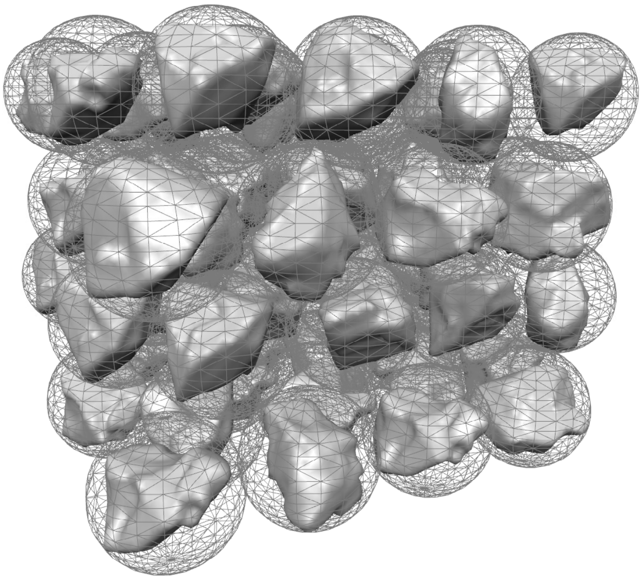

Figure 3: Hexagonal close-packing of equal bounding spheres for the initial grain arrangement.

edges or by randomly changing angles and edge lengths. Figure 2 illustrates examples. The convex grains are meshed by using the convex hull algorithm *Quickhull* [1].

The initial arrangement of grains in three-dimensional space is created by computing the smallest bounding sphere for each grain with an algorithm by Gärtner [10] and distributing the grains such that a hexagonal close-packing of equal spheres regarding the maximum bounding sphere radius is obtained. An example is depicted in Fig. 3. A volumetric structure element can be created by cutting out the contents of an arbitrary bounding box. If the grain volume fraction in such a bounding box is larger than prescribed by a grinding wheel specification, the maximum bounding sphere radius can be increased to match the desired grain volume fraction exactly. This is possible due to the knowledge that space can be filled up to a density of about 74 % with a close-packing of equal spheres, see [7].

For the case of the grain volume fraction being smaller than a prescribed value, we have developed a grain compaction algorithm: the grains are moved toward the center point of the bounding box or to a plane through the origin of the bounding box to get a denser volumetric structure element. For a more efficient compaction, the translations of the grains can be combined with rotations. The effect of the algorithm is demonstrated in Fig. 4: on the left-hand side, an initial arrangement for 27 grains from computer tomography scan data with decimated meshes can be seen together with a bounding box. The grain volume fraction in the bounding box is 26 %. The right-hand side shows the structure element after grain compaction with rotations, followed by the computation of the intersections with the bounding box and the remeshing of the grains. The resulting grain volume fraction has increased to 48 %.

The compaction algorithm can lead to a dense packing of grains around the point or plane toward which the grains have been compacted, while the distances between the grains increase away from the point or plane.

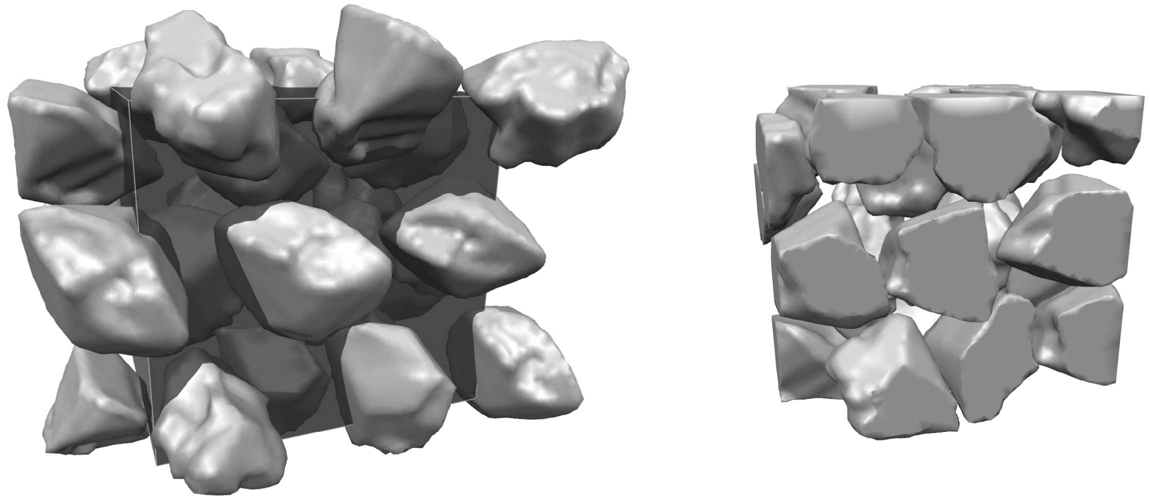

Figure 4: Initial arrangement of grains (left) and volumetric structure element after grain compaction, intersection with bounding box and grain remeshing (right).

Hence, we have developed a relaxation algorithm such that the grains can be distributed more evenly. Each grain inside the volumetric structure element is moved into an arbitrary randomized direction by translation or by rotation plus subsequent translation. Grains which intersect at least one of the six bounding planes are fixed. The algorithm can be used once the desired grain volume fraction has been obtained.

A tetrahedral meshing of a volumetric structure element like the one shown on the right-hand side of Fig. 4 can be applied by using the mesh generator *TetGen* [21]. The tetrahedral mesh is used for the determination of cells belonging to bond bridges as explained in Sect. 3.6.

More information on our modeling concept can be found in [5, 19].

3 Ceramic bond modeling

For the production of ceramic-bonded grinding wheels, abrasive grains and bonding material in powder form are pressed into a mold. In a sintering process, the mixture is heated such that the bonding material liquefies while the grains remain solid. The bonding material flows around the grains and solidifies after the heating process such that bond bridges between the grains develop. Due to the minimization of the surface tension, the bond bridges are shaped as minimal surfaces [18, 22]. Since the construction of minimal surfaces is in general a process with a high computational cost, our objective is the modeling of the bond bridge surfaces by approximating the minimal surfaces by using a fast alternative based on B-splines.

An overview of our iterative algorithm for the generation of bond bridges is provided in Sect. 3.1. The single steps of this algorithm are explained in detail in Sects. 3.2 to 3.7. Section 3.8 describes how we make sure to meet prescribed bond volume fractions up to a tolerance.

3.1 Algorithm for the generation of bond bridges

Our iterative algorithm for the modeling of ceramic bond bridges is summarized as pseudocode in Alg. 1. The first step is to determine for each grain to which of its adjacent grains it should be connected by a bond bridge (lines 1-3). After the initialization of the bond bridges (lines 4 and 19-27), the iterative process (lines 5-18) can be started. A control polygon for a B-spline curve is created between the two grains of each bond bridge and subdivided several times, resulting in a discrete approximation of a B-spline curve (line 7). By rotating such a control polygon around its bond bridge axis, a set of curve approximations is obtained which can be triangulated to visualize the bond bridge by a surface mesh (line 8). With the help of the control polygons, the tetrahedral cells belonging to the bond bridge are determined (line 9). The connection of a bond bridge to its two grains can be adjusted by introducing further tetrahedral bond cells to obtain a more realistic rounded attachment of the bond to the grains (line 10). The iterative process is stopped if the overall bond volume fraction in the volumetric structure element differs from a prescribed value by less than a tolerance. The bond volume fraction is adjusted by changing one single parameter (lines 13-17). Usually, the algorithm needs only three or four iterations to converge when the tolerance is set to 0.001 (0.1 percentage point).

Algorithm 1 Ceramic bond modeling

```
 1: for each grain do
 2:     Determine the adjacent grains which are to be connected by bond bridges;        ▷ Sect. 3.2
 3: end for
 4: InitializeBondBridges;                                                              ▷ Sect. 3.3
 5: repeat
 6:     for each bond bridge between two grains do
 7:         Compute approximations of B-spline curves based on control polygons containing   ▷ Sect. 3.4
              a point on each of the three circles introduced in InitializeBondBridges;
 8:         Triangulate the surface induced by the B-spline curve approximations; (optional)  ▷ Sect. 3.5
 9:         Mark all tetrahedral cells whose center point lies “inside” the bond bridge surface  ▷ Sect. 3.6
              as bond cells;
10:         Grow the bond bridge at the grain surfaces by introducing new bond cell layers;   ▷ Sect. 3.7
11:     end for
12:     Compute the overall bond volume fraction;
13:     if the overall bond volume fraction is too low then
14:         Increase the middle circle radii;                                           ▷ Sect. 3.8
15:     else
16:         Decrease the middle circle radii;                                           ▷ Sect. 3.8
17:     end if
18: until the overall bond volume fraction differs from the prescribed value by less than a tolerance

19: procedure InitializeBondBridges
20:     for each bond bridge to be constructed between two grains do
21:         Compute the axis through the two grain centers;
22:         for each of the two grains do
23:             Compute an approximation of the maximum inscribed circle of the grain (perpendicular to
                  the axis);
24:         end for
25:         Compute a circle in the middle of the bond bridge with a radius of one half of the minimum of
              the other two radii;
26:     end for
27: end procedure
```

3.2 Detection of grain adjacency

The grains in a volumetric structure element are identified by a continuous numbering. For each grain i, a list of adjacent grains with identification indices larger than i is stored. The list contains the grains which are to be connected by bond bridges. Whether a grain with the index j is detected as adjacent to the grain i (with $i < j$) is determined by the distance $d_{i,j}$ between the center point $\vec{c}_i$ and the center point $\vec{c}_j$ of the potential neighbor. Adjacency is given if

$$d_{i,j} = \|\vec{c}_j - \vec{c}_i\|_2 < R\, r_{\mathrm{bs,max}} \quad \text{with } R \in [1.5, 2.5] \;, \tag{1}$$

where R is a random number and $r_{\mathrm{bs,max}}$ denotes the maximum bounding sphere radius of all grains. Using R as a factor guarantees a random distribution of the bond in the model.

3.3 Initialization of bond bridges

The setup for modeling a bond bridge between two grains i and j is illustrated in a two-dimensional drawing in Fig. 5. For the initialization, the axis $\vec{a}_{i,j}(t)$ of the bond bridge with the two grain centers $\vec{c}_i$ and $\vec{c}_j$ as end points is computed first by

$$\vec{a}_{i,j}(t) = \vec{c}_i + t\,(\vec{c}_j - \vec{c}_i) \quad \text{with } t \in [0,1] \ . \tag{2}$$

The next step is the computation of an approximation of the maximum inscribed circle inside each of the two grains with the circles being perpendicular to the axis $\vec{a}_{i,j}(t)$. This is now shown for grain i. The computations for grain j are done analogously, only the direction of the axis has to be inverted. The point $\vec{x}_i$ at which the axis intersects the grain, see Fig. 5, is needed. To obtain $\vec{x}_i$, triangular faces of the grain mesh are tested to be intersected by the axis. We denote the face points by $\vec{p}_{k,l}$ where $k \in \{1,\dots,N_i\}$ is the index of the face, N_i is the number of triangular faces of the grain mesh i and $l \in \{1,2,3\}$ is the index of the face point. Assume without loss of generality that the three triangle edges $\vec{p}_{k,1}\vec{p}_{k,2}$, $\vec{p}_{k,2}\vec{p}_{k,3}$ and $\vec{p}_{k,3}\vec{p}_{k,1}$ are ordered counterclockwise viewed from $\vec{c}_i$. Then,

$$\begin{aligned} u &= (\vec{c}_j - \vec{c}_i) \cdot ((\vec{p}_{k,3} - \vec{c}_i) \times (\vec{p}_{k,2} - \vec{c}_i)) \ , \\ v &= (\vec{c}_j - \vec{c}_i) \cdot ((\vec{p}_{k,1} - \vec{c}_i) \times (\vec{p}_{k,3} - \vec{c}_i)) \ , \\ w &= (\vec{c}_j - \vec{c}_i) \cdot ((\vec{p}_{k,2} - \vec{c}_i) \times (\vec{p}_{k,1} - \vec{c}_i)) \end{aligned} \tag{3}$$

are nonnegative if the axis passes to the left of each of the three edges. If this case is found for a triangular face, the face is intersected by the axis and the scalars u, v and w are used to compute the intersection point

$$\vec{x}_i = \frac{1}{u+v+w}\,(u\,\vec{p}_{k,1} + v\,\vec{p}_{k,2} + w\,\vec{p}_{k,3}) \ . \tag{4}$$

The intersection tests only have to be conducted if the angle between the axis direction vector and the normal vector of a face is smaller than 90°, i.e., if the scalar product of the direction vector with the face normal is larger than zero. More information on the efficient implementation of the detection of intersections of lines with triangles can be found in [9].

The radius r_i of the approximation of the maximum inscribed circle of grain i, see Fig. 5, is searched by placing the center of the circle on the bond bridge axis at several positions and computing the maximum possible radius such that a circle would not intersect the grain faces or edges. The positioning of the circle centers $\vec{c}_{c,i}$

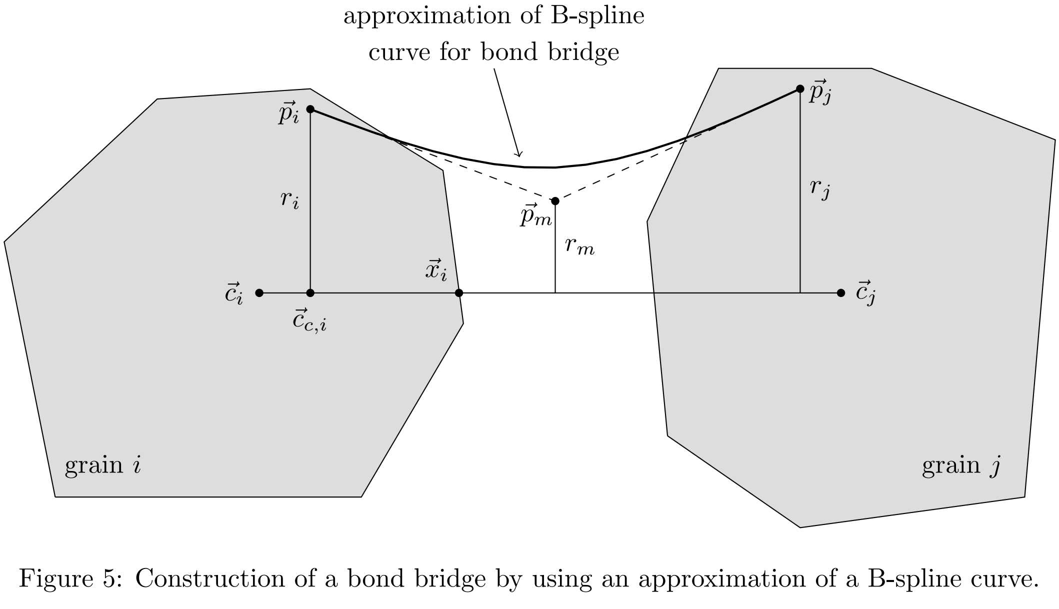

Figure 5: Construction of a bond bridge by using an approximation of a B-spline curve.

is done by choosing a number of iterations n_c and computing the current circle center for $c = 0, \ldots, n_c - 1$ by

$$\vec{c}_{c,i} = \vec{c}_i + \frac{c}{n_c} \left(\vec{x}_i - \vec{c}_i \right) \, . \tag{5}$$

For each position of the circle center $\vec{c}_{c,i}$, the maximum radius is found by testing for intersections between the circle and the grain edges and faces. With the circle plane given by

$$\left(\vec{x} - \vec{c}_{c,i} \right) \cdot \vec{n}_{c,i} = 0 \, , \tag{6}$$

where $\vec{n}_{c,i} = \left(\vec{c}_j - \vec{c}_i \right) / \| \vec{c}_j - \vec{c}_i \|_2$ is the unit normal vector of the circle plane, and an edge with end points $\vec{e}_1$ and $\vec{e}_2$ formulated as the line segment

$$\vec{x}(s) = \vec{e}_1 + s \left(\vec{e}_2 - \vec{e}_1 \right) \quad \text{with } s \in [0, 1] \, , \tag{7}$$

an intersection is found if

$$s = \frac{\left(\vec{c}_{c,i} - \vec{e}_1 \right) \cdot \vec{n}_{c,i}}{\left(\vec{e}_2 - \vec{e}_1 \right) \cdot \vec{n}_{c,i}} \in [0, 1] \, . \tag{8}$$

If the distance between the intersection point and the circle center is smaller than the currently stored maximum possible radius, the radius is updated.

For the computations regarding the intersection of a triangular face k of grain i with the circle, we refer to [9] again. The face plane and the circle plane intersect in a line $\vec{l}(r) = \vec{p} + r \, \vec{d}$ if the planes are not parallel to each other. With $\vec{n}_k$ denoting the normal vector of the triangular face k, the direction vector $\vec{d}$ is defined by

$$\vec{d} = \vec{n}_k \times \vec{n}_{c,i} \, . \tag{9}$$

With

$$d_1 = \vec{p}_{k,1} \cdot \vec{n}_k \, , \quad d_2 = \vec{c}_{c,i} \cdot \vec{n}_{c,i} \, , \tag{10}$$

the point $\vec{p}$ is obtained by

$$\vec{p} = \frac{\left(d_1 \vec{n}_{c,i} - d_2 \vec{n}_k \right) \times \vec{d}}{\vec{d} \cdot \vec{d}} \, . \tag{11}$$

The point on the line $\vec{l}(r)$ closest to the circle center is given by $\vec{p} + r_c \, \vec{d}$ with the parameter r_c determined by

$$r_c = \frac{\left(\vec{c}_{c,i} - \vec{p} \right) \cdot \vec{d}}{\vec{d} \cdot \vec{d}} \, . \tag{12}$$

If this point lies inside the currently investigated triangular face and if the distance between the point and the circle center is smaller than the currently stored maximum circle radius, the radius is updated.

When the circle radius r_i has been determined, the point $\vec{p}_i$ on the circle is computed: a vector of length r_i which is perpendicular to the axis $\vec{a}_{i,j}(t)$ is added to $\vec{c}_{c,i}$. By using this axis-perpendicular vector with a length of r_j, the circle point $\vec{p}_j$ is obtained.

The last step of the bond bridge initialization is the construction of a circle in the middle of the axis $\vec{a}_{i,j}(t)$. Its radius r_m is set to

$$r_m = \frac{1}{2} \min \left(r_i, r_j \right) . \tag{13}$$

With the above-mentioned axis-perpendicular vector, the circle point $\vec{p}_m$ is calculated. The radius r_m and, hence, the point $\vec{p}_m$ are variable and changed during the iterative bond bridge generation algorithm to converge to a prescribed overall bond volume fraction, see Sect. 3.8.

3.4 Approximation of B-spline curves

The three points $\vec{p}_i$, $\vec{p}_m$ and $\vec{p}_j$ are used for the construction of a control polygon for a B-spline curve. Extensive information on the theoretical background on B-splines can be found in [3]. By subdividing the control polygon several times, a discrete approximation of a B-spline curve can be obtained. The subdivision rules originate from

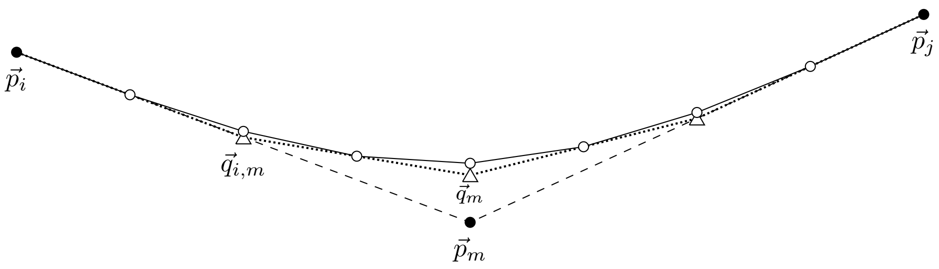

Figure 6: Illustration of the cubic edge split scheme (original points: •, points of first subdivision: △, points of second subdivision: ○).

cubic edge splitting which for a closed polygon in the limit results in a cubic B-spline curve. Each subdivision introduces new control points on the edges of the control polygon and recomputes the already existing control points, except for the two end points $\vec{p}_i$ and $\vec{p}_j$ which in our setting are fixed to their particular circle. This is illustrated in Fig. 6 for two subdivisions of the control polygon $\vec{p}_i$, $\vec{p}_m$, $\vec{p}_j$ of Fig. 5. New control points on the edges are inserted with the ratio 1 : 1 of the adjacent points, whereas existing control points are recomputed with the ratio 1 : 6 : 1. Exemplarily, the points $\vec{q}_{i,m}$ and $\vec{q}_m$ depicted in Fig. 6 are calculated by

$$\vec{q}_{i,m} = \frac{1}{2}\,\vec{p}_i + \frac{1}{2}\,\vec{p}_m\,, \quad \vec{q}_m = \frac{1}{8}\,\vec{p}_i + \frac{3}{4}\,\vec{p}_m + \frac{1}{8}\,\vec{p}_j\,. \tag{14}$$

Usually, we subdivide three times such that we end up with a control polygon with 17 points as a discrete approximation of a B-spline curve. These control points provide enough information to decide which tetrahedral cells in the volumetric structure element belong to a bond bridge. This is explained in Sect. 3.6.

3.5 Generation of triangle meshes for the bond bridges

For the visualization of the bond bridge surfaces, we can generate surface meshes from the control points computed as described in Sect. 3.4. For each bond bridge, its corresponding control points are rotated around the axis $\vec{a}_{i,j}(t)$ by applying the formula for the rotation of a point $\vec{x}$ around an arbitrary axis $\vec{a}$ (with $\vec{a}\cdot\vec{a} = 1$) through the origin, leading to the rotated point

$$\vec{x}_r = (\vec{a}\cdot\vec{x})\,\vec{a} + \cos(\varphi_r)\,((\vec{a}\times\vec{x})\times\vec{a}) + \sin(\varphi_r)\,(\vec{a}\times\vec{x})\ . \tag{15}$$

For this computation, the axis $\vec{a}_{i,j}(t)$ has to be translated such that it passes through the origin. The computed points are translated back later. With a given number of rotations N_r, the rotation angle φ_r is computed for $r = 0, \ldots, N_r - 1$ by

$$\varphi_r = 2\,\pi\,\frac{r}{N_r}\,. \tag{16}$$

We usually use $N_r = 30$ such that we get $17\cdot 30 = 510$ control points around the axis. From these points, a structured triangle mesh can be generated easily, resulting in $2\cdot(17-1)\cdot 30 = 960$ triangular faces. An example is presented in Fig. 7 which shows the meshed bond bridge for two grains from computer tomography scan data in a translucent view.

3.6 Detection of tetrahedral bond bridge cells

As mentioned in Sect. 2, we can generate a tetrahedral mesh for a volumetric structure element by using the mesh generator *TetGen* [21]. The maximum volume of the tetrahedral cells can be prescribed. A cutaway view of an example with grains from computer tomography scan data and with bond bridge meshes can be seen in Fig. 8.

The task is to mark all tetrahedral cells which belong to a bond bridge. A two-dimensional illustration of the setting for a single bond bridge is shown in Fig. 9 which is the result of one subdivision of the polygon given

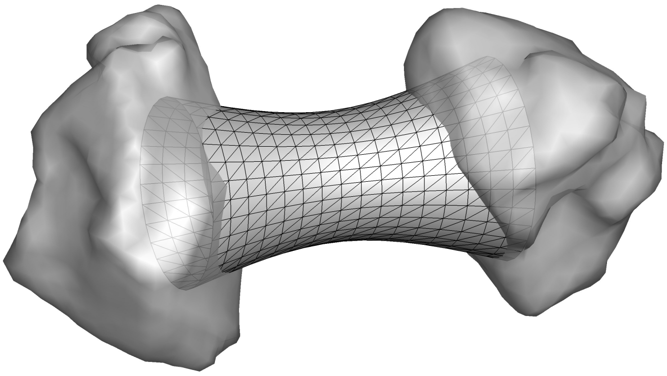

Figure 7: Bond bridge mesh for two grains from computer tomography scan data (translucent view).

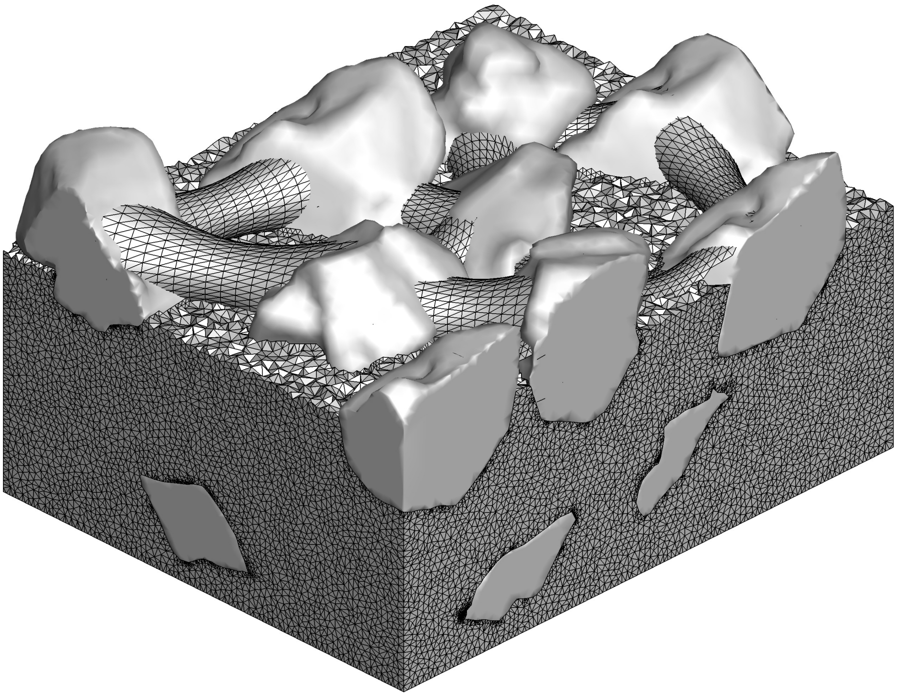

Figure 8: Cutaway view of a volumetric structure element with tetrahedral mesh and bond bridge surface meshes.

by $\vec{p}_i$, $\vec{p}_m$ and $\vec{p}_j$, cf. Fig. 6. The orthogonal projection of the center point $\vec{c}_t$ of a tetrahedral cell onto the axis $\vec{a}_{i,j}(t)$ of the bond bridge determines the parameter

$$t_t = \frac{(\vec{c}_t - \vec{c}_{c,i}) \cdot (\vec{c}_j - \vec{c}_i)}{(\vec{c}_j - \vec{c}_i) \cdot (\vec{c}_j - \vec{c}_i)} \tag{17}$$

for which $t_t \in [0, 1]$ has to be fulfilled. Otherwise, the tetrahedral cell cannot belong to the bond bridge because it exceeds the bounds defined by the end circles of the bridge. In the illustration, the center points of the cells 1 and 2 lie at $t_t = 0.15$, whereas the center points of the cells 3 and 4 are positioned at $t_t = 0.1$.

The first step is to find all cells for which $t_t \in [0, 1]$ and which lie inside the cylinder whose radius r_{cyl} is defined by the minimum distance between the axis and a control point, in the illustration $\vec{q}_m$. For this purpose,

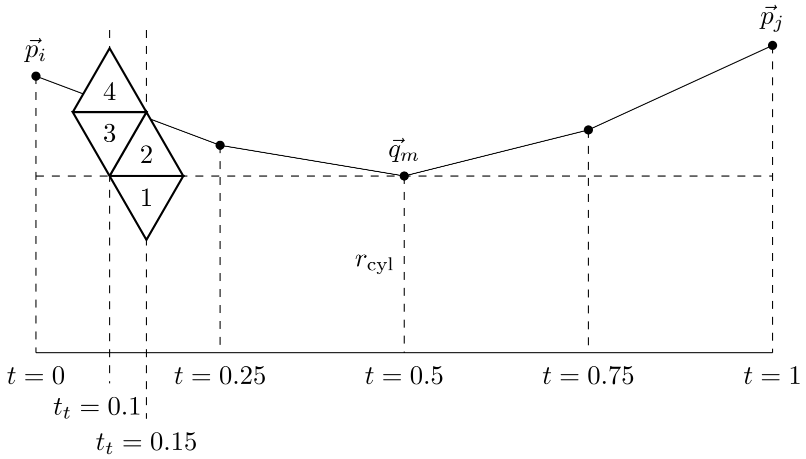

Figure 9: Setting for the determination of tetrahedral cells which belong to a bond bridge.

the distance d_t between a tetrahedral cell center $\vec{c}_t$ and the axis $\vec{a}_{i,j}(t)$ is computed by

$$d_t = \left((\vec{c}_t - \vec{c}_{c,i}) \cdot (\vec{c}_t - \vec{c}_{c,i}) - t_t^2 \, (\vec{c}_j - \vec{c}_i) \cdot (\vec{c}_j - \vec{c}_i) \right)^{\frac{1}{2}} \tag{18}$$

and compared with r_{cyl}.

Subsequently, an iterative process finds the cells whose center point lies between the cylinder and the bond bridge surface. In each iteration, it is verified for each current outer tetrahedral cell of the bond bridge whether its adjacent non-bond cells also have to be marked as bond cells. This is done by computing the distance d_t between the center of the investigated adjacent cell and the axis according to (18). The distances between the B-spline control points, see Sect. 3.4, and the axis are stored dependent on the parameter t. In the example in Fig. 9, these are the distances $d(0)$, $d(0.25)$, $d(0.5)$, $d(0.75)$ and $d(1)$. If d_t is smaller than the two distances at the particular interval ends $t_{t,\text{left}}$ and $t_{t,\text{right}}$, the tested cell is marked as a new outer bond cell. In the example with $t_t = 0.1$ and $t_t = 0.15$, we have $t_{t,\text{left}} = 0$ and $t_{t,\text{right}} = 0.25$. If d_t is smaller than only one of the two interval end distances, the interpolated control polygon distance is calculated by

$$d(t_t) = d(t_{t,\text{left}}) + \frac{d(t_{t,\text{right}}) - d(t_{t,\text{left}})}{t_{t,\text{right}} - t_{t,\text{left}}} \, (t_t - t_{t,\text{left}}) \ . \tag{19}$$

If $d_t < d(t_t)$, the investigated cell is marked as a new outer bond cell.

In the illustration in Fig. 9, cell 1 lies inside the cylinder and, hence, is a bond cell. For cell 2, d_t is smaller than both interval end distances ($d_t < d(t_{t,\text{left}}) = d(0)$ and $d_t < d(t_{t,\text{right}}) = d(0.25)$), whereas $d_t < d(t_t) = d(0.1)$ is fulfilled for cell 3. Hence, these two cells would be found as new outer bond cells in the first and second iteration of the subsequent process, respectively. Due to $d_t > d(t_t) = d(0.1)$, cell 4 would not be added to the bond cells.

For the bond bridge of Fig. 7, the detected bond cells are shown in Fig. 10. The slice in Fig. 11 combines a view of the bond bridge and of the bond cells.

With the stored distances between the control points and the axes and with (19), a faster algorithm could directly compute for each tetrahedral cell whether it belongs to a bond bridge or not. The calculations incorporating the cylinder radii and (19) in an iterative way would not be necessary. The reason why we do not use the faster approach is demonstrated in Fig. 12. Due to the possible non-convexity of the grains, the bond bridge surfaces can intersect the grains at several positions. Hence, it is possible that a cell is detected as a bond cell although it does not belong to the bond bridge as can be seen on the right grain. This is much more unlikely to happen if the bond cells in the bond bridge cylinder are determined first and the remaining bond cells are detected as direct neighbors of these cylinder bond cells. A grain would have to contain a deep dent such that an isle of bond bridge cylinder cells could occur.

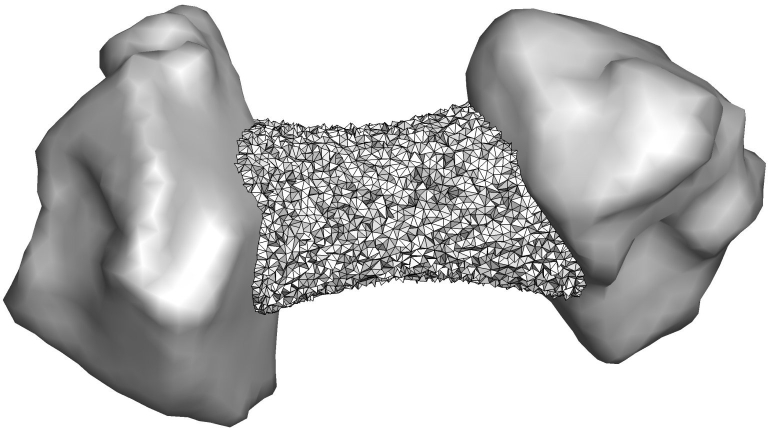

Figure 10: Tetrahedral cells marked as bond cells based on the bond bridge depicted in Fig. 7.

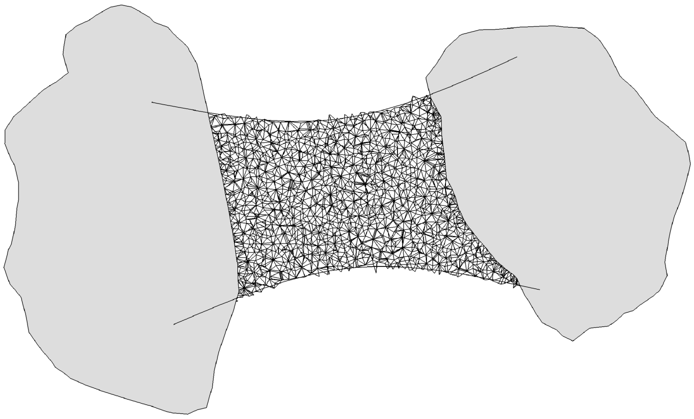

Figure 11: Slice through the grains, the bond bridge mesh and the bond cells depicted in Figs. 7 and 10.

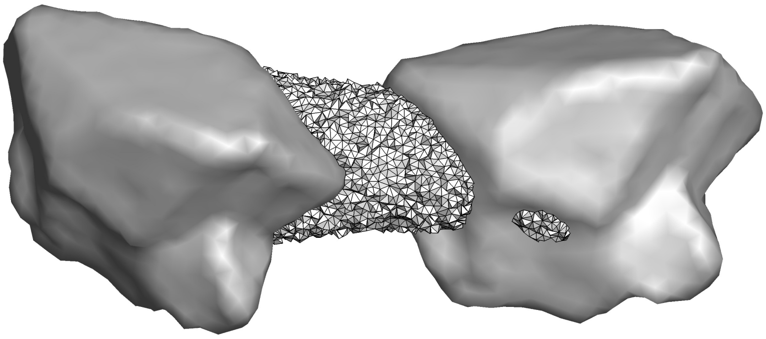

Figure 12: Undesirable behavior of fast bond cell detection: the bond cells in the dent on the right grain should not belong to the bond bridge.

3.7 Bond growth at the grains

Since parts of the bond bridges lie inside the grains, cf. Fig. 11, the connection of a bond bridge to a grain does not necessarily behave as expected from a minimal surface. To improve this, we have developed an algorithm which adds layers of bond cells at the grains to obtain a more realistic rounded attachment.

Algorithm 2 lists the procedure of adding layers of bond cells. It can be divided into the three procedures stated in the lines 2 to 4.

Algorithm 2 Bond growth at the grains in layers

```
1:  **for** each bond bridge $b$ **do**
2:      GrowBondAtGrains($b$);
3:      FillGapsOnGrains($b$);
4:      IntroduceAdditionalLayer($b$);
5:  **end for**
```

We will describe the three steps of our bond growth algorithm in detail now.

1. GrowBondAtGrains: For each bond bridge, the algorithm searches for tetrahedral non-bond cells which are connected to one of the two grains of the bridge by a face and which touch bond cells of the bridge by at least one mesh point. If such a cell is found, the angle between the grain face normal vector and the axis direction vector of the bond bridge decides whether the cell is marked as a new bond cell or not. By prescribing a threshold angle α_1, the growth of the bond bridge at the grains can be regulated. If the bond bridge axis hits a grain face perpendicularly, i.e., the angle is zero, it is more reasonable to add a new bond cell to get a rounded connection than in the case of a large angle where a rounded connection of the bond to the grain already exists. A typical application of the algorithm could comprise three iterations with threshold angles α_1 of 30°, 60° and 90°, respectively.

   An example for bond growth with a threshold angle of 45° is depicted on the left-hand side of Fig. 13. The darker cells at the grain surface have been added by the bond growth algorithm. On the lower right of the bond bridge, where the bond adheres to the grain in a flatter manner, no new cells have been added.

```
1:  **procedure** GrowBondAtGrains($b$)
2:      **for** each cell $i$ (including the ones added in line 11) of the bond bridge $b$ **do**
3:          **if** the cell $i$ is connected to one of the two grain meshes by a face **then**
4:              **for** each mesh point $j$ of the cell $i$ **do**
5:                  Count the number $n_b$ of adjacent bond cells (without cell $i$);
6:                  Count the number $n_n$ of adjacent non-bond cells;
7:                  **if** $n_b > 0$ and $n_n > 0$ **then**
8:                      **for** each non-bond cell $k$ adjacent to the bond bridge cell $i$ **do**
9:                          **if** the cell $k$ is connected to the grain mesh by a face **then**
10:                             **if** the angle between the grain face normal vector and the axis direction
                                vector is smaller than a threshold angle $\alpha_1$ **then**
11:                                 Mark the cell $k$ as a new bond bridge cell;
12:                             **end if**
13:                         **end if**
14:                     **end for**
15:                 **end if**
16:             **end for**
17:         **end if**
18:     **end for**
19: **end procedure**
```

2. FillGapsOnGrains: Since only cells which are connected to grains by a face (and not only by a mesh point) are added as new bond cells, the space between the new bond cells and the already existing ones has to be filled with further bond cells to obtain a smoother transition than visible on the left-hand side of Fig. 13. To fill the gaps, for each mesh point of the two grains of a bond bridge the number of grain faces surrounding the mesh point is compared with the number of surrounding bond cells connected to the particular grain mesh by a face. If these numbers are equal, all other surrounding cells of the mesh point

are marked as bond cells as well. The result of our gap filling algorithm can be seen on the right-hand side of Fig. 13.

```
procedure FillGapsOnGrains($b$)
    for each grain $i$ of the bond bridge $b$ do
        for each mesh point $j$ of the grain $i$ do
            Count the number $n_f$ of adjacent grain mesh faces;
            Count the number $n_b$ of adj. bond cells connected to the grain mesh by a face;
            if $n_f == n_b$ then
                Store the mesh point $j$ in a list;
            end if
        end for
        for each mesh point $j$ of the list do
            Mark all cells adjacent to the mesh point $j$ as new bond bridge cells;
        end for
    end for
end procedure
```

3. IntroduceAdditionalLayer: Depending on a second threshold angle α_2, the bond growth can be extended by introducing an additional layer of bond cells away from the grain surfaces. This is reasonable in regions of small angles between the bond bridge axis and the grain face normals to obtain a better roundness of the bond. For each cell which has been added by the first two procedures of the bond growth algorithm, all cells which are adjacent by at least one mesh point are searched. If in turn such an adjacent cell is adjacent by at least one mesh point to an old bond cell (which has not been added by the bond growth algorithm but before), it is marked as a new bond cell.

   Figure 14 demonstrates the effect of introducing an additional layer of bond cells. The slices show a part of a bond bridge. The bond growth algorithm has been applied with a threshold angle $\alpha_1 = 45°$ for three iterations. The left-hand side is the result for the threshold angle α_1 being larger than α_2 in each iteration of the algorithm, i.e., an additional layer of bond cells has not been introduced at any stage. The right-hand side shows a better roundness of the bond. This is the consequence of adding an additional layer in the first iteration.

```
procedure IntroduceAdditionalLayer($b$)
    if the threshold angle $\alpha_1$ is smaller than or equal to a threshold angle $\alpha_2$ then
        for each new bond cell $i$ added by GrowBondAtGrains or FillGapsOnGrains do
            for each cell $j$ adjacent to the new bond cell $i$ (adjacent by mesh point) do
                for each cell $k$ adjacent to the adjacent cell $j$ (adjacent by mesh point) do
                    if the cell $k$ is an old bond cell (not added by GrowBondAtGrains or FillGapsOnGrains) then
                        Mark the cell $j$ as a new bond bridge cell;
                    end if
                end for
            end for
        end for
    end if
end procedure
```

The bond bridge of Fig. 10 after three bond growth steps with threshold angles α_1 of 30°, 60° and 90°, respectively, and a threshold angle α_2 of 30° in each iteration is depicted in Fig. 15.

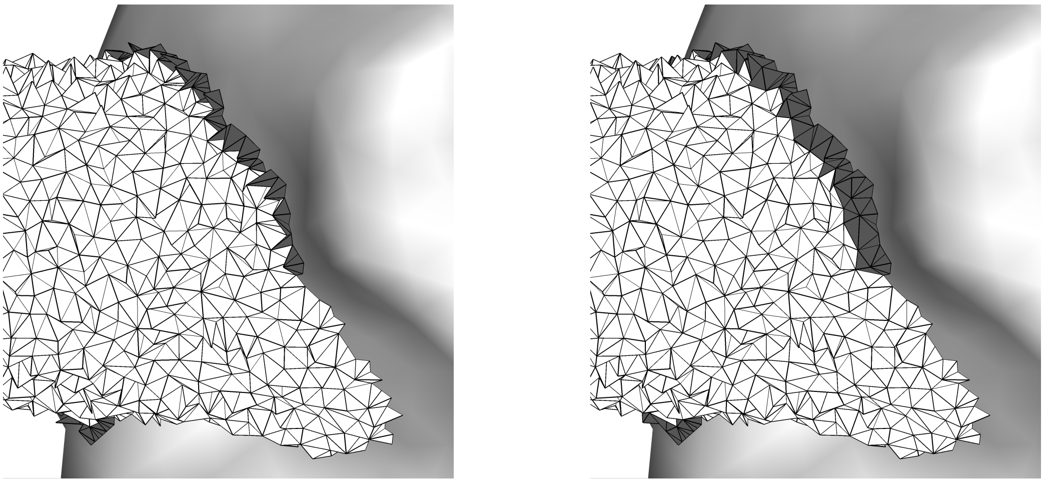

Figure 13: New bond cells (dark grey) added by one iteration of the bond growth algorithm applied for a threshold angle of 45° (left) and bond after gap filling (right).

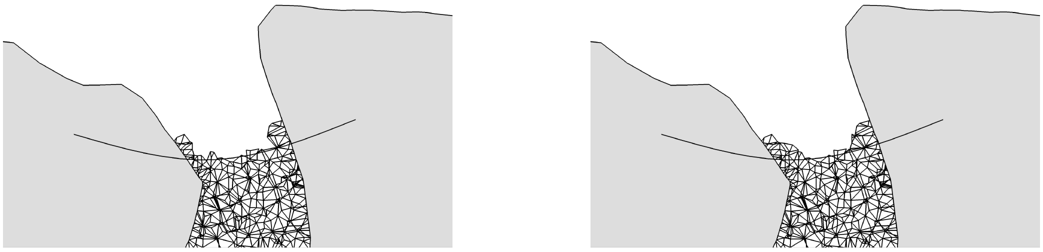

Figure 14: Slices through a part of a bond bridge after three bond growth iterations: without introducing an additional layer of bond cells (left) and with introducing an additional layer in the first iteration (right).

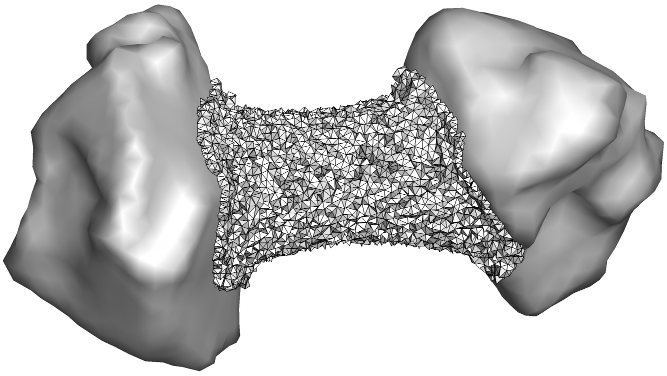

Figure 15: Tetrahedral bond cells after three iterations of bond growth at the grains, based on the bond bridge depicted in Fig. 10.

3.8 Iterative adaptation of the bond volume fraction

The parts of our ceramic bond modeling approach described in Sects. 3.4 to 3.7 are embedded into an iterative algorithm which ensures that a prescribed overall bond volume fraction is obtained. The only parameter that is changed from one iteration to the next is the radius r_m of the middle circles of the bond bridges, cf. Fig. 5 and (13).

The middle circle radius $r_{m,n}$ for a bond bridge of iteration n is dependent on a factor f_n and computed by

$$r_{m,n} = f_n \min(r_i, r_j) , \tag{20}$$

where r_i and r_j are the radii of the two end circles of the bond bridge, cf. Fig. 5. The initial factor f_1 is set to 0.5, i.e., $r_{m,1}$ is given by (13). If the overall bond volume fraction is too low after the first iteration of bond modeling, the factor is increased to $f_2 = 0.75$ such that $r_{m,2} = 0.75 \min(r_i, r_j)$ for the second iteration. If instead the bond volume fraction is too large, the factor is decreased to $f_2 = 0.25$. Subsequently, we use the secant method to determine the factor for the respective next iteration $n + 1$ by

$$f_{n+1} = f_n - \frac{f_n - f_{n-1}}{V_n - V_{n-1}} (V_n - V_{\text{target}}) \quad \text{with } n \geq 2 , \tag{21}$$

where V denotes the bond volume fraction. The bond modeling algorithm is terminated if

$$|V_n - V_{\text{target}}| < \varepsilon \tag{22}$$

for a prescribed tolerance ε or if a maximum number of iterations is reached.

4 Results

In Fig. 16, we present two modeled volumetric structure elements. The element on the left-hand side contains simplified convex grains based on randomized tetrahedra. It was generated with 125 grains with an average grain size of 46 *μm*. After compacting up to a grain volume fraction of 40 %, the ceramic bond was generated with a bond volume fraction of 20 %. The final result was obtained by cutting out the contents of a cubic bounding box with an edge length of 160 *μm* such that 111 grains were left.

For the element on the right-hand side of Fig. 16, we used 125 grains obtained from computer tomography scan data. The average grain size is 91 *μm*. Here, we compacted up to a grain volume fraction of 35 % and generated ceramic bond bridges with a bond volume fraction of 15 %. Cutting with a cubic bounding box with an edge length of 320 *μm* reduced the number of grains to 103.

The left-hand side of Fig. 17 shows a part of a real structure cut of a specimen with CBN grains and ceramic bond. The original structure cut, presented in [2], has a size of 4 *mm* × 4 *mm*. We have taken a part of this

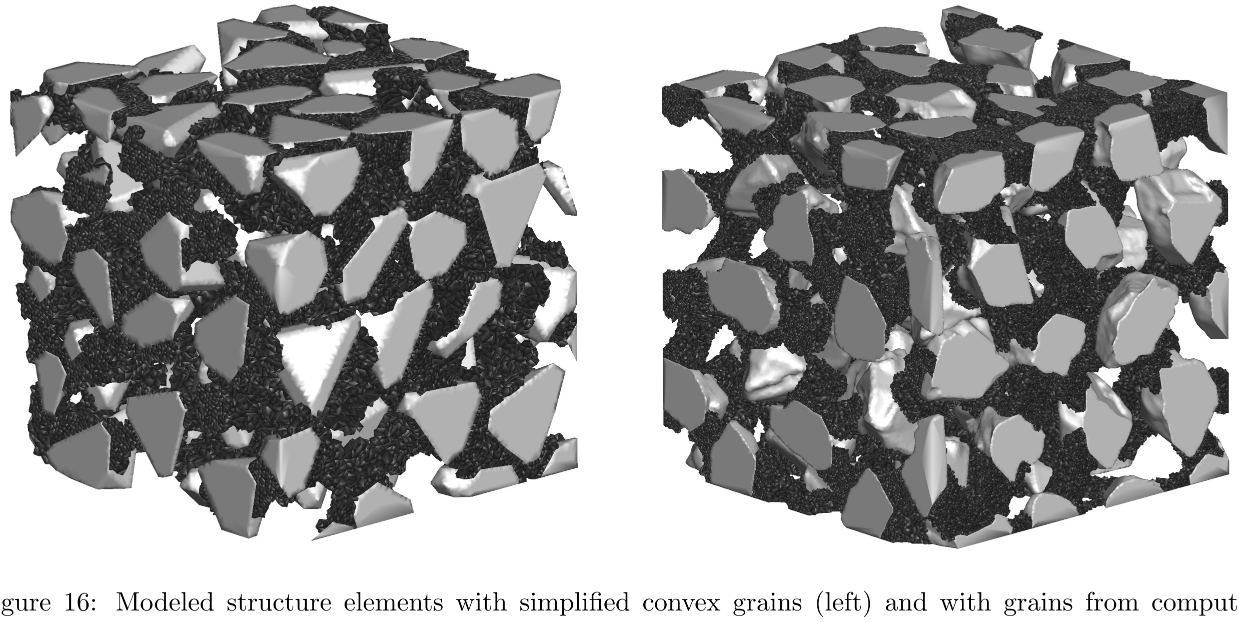

Figure 16: Modeled structure elements with simplified convex grains (left) and with grains from computer tomography scan data (right).

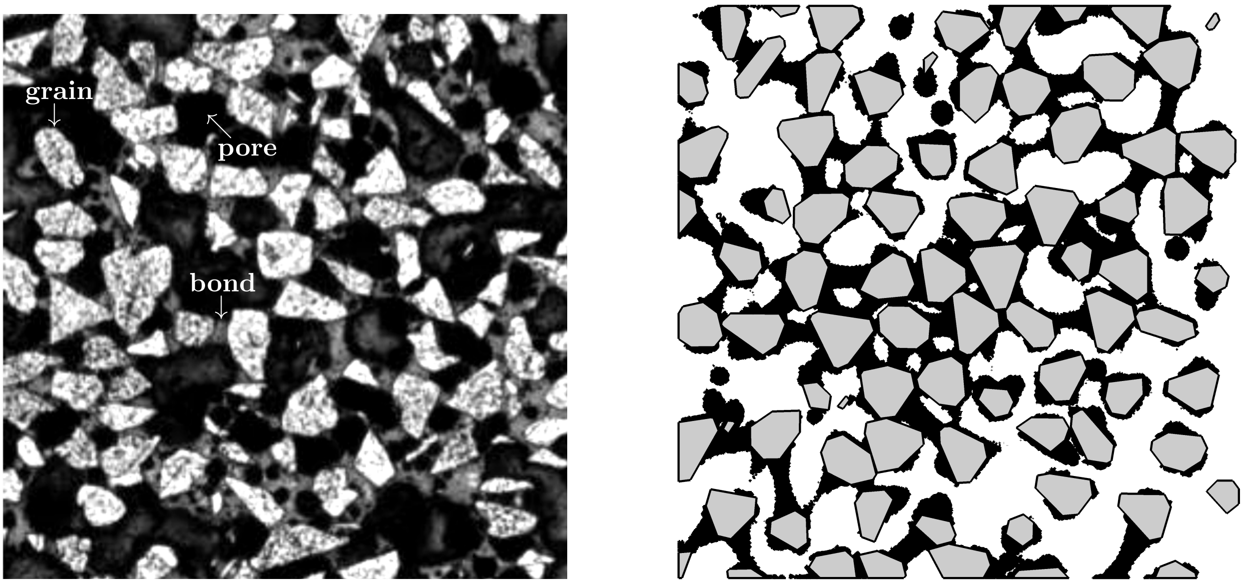

Figure 17: Comparison of a real structure cut of a specimen with CBN grains and ceramic bond (left) and a modeled structure cut with simplified convex grains (gray) and bond bridges (black) (right).

picture with an approximate size of $0.91\,mm \times 0.91\,mm$ here. The grain volume fraction of 40 % and the bond volume fraction of 20 % of the real structure were used for a modeled element with randomized tetrahedra as grains for which a structure cut can be seen on the right-hand side of Fig. 17. From a qualitative comparison, we can conclude that both pictures are similar regarding grain sizes, grain distribution and bond bridges.

5 Conclusion and future work

In this paper, we have presented algorithms for the modeling of ceramic bond bridges in grinding wheels. The starting point is a volumetric structure element composed of simplified convex grains or grains obtained from computer tomography scan data. The grains are arranged according to a close-packing of their bounding spheres and can be compacted and relaxed by rotations and translations to achieve prescribed grain volume fractions. The ceramic bond bridges are then modeled based on the computation of control points for B-spline curves, where the control polygons can be considered as discrete approximations of B-spline curves. The end points lie inside the two grains to which the particular bond bridge is attached. By rotating such a control polygon around the axis connecting the two involved grain centers and meshing the surface induced by the curves, a discrete representation of a bond bridge can be constructed for visualization purposes. We use the control polygons to decide which cells of the tetrahedral mesh of a volumetric structure element belong to a bond bridge. Further tetrahedral cells can be added to bond bridges by our bond growth algorithm to obtain more rounded attachments of the bond to the grains. For converging to a prescribed bond volume fraction in a volumetric structure element, we have implemented our bond modeling algorithm as an iterative process. Only one parameter is needed to regulate the bond volume fraction, namely the radius of the bond bridges at the narrowest point, i.e., in the middle of the bond bridges.

A validation of our grinding wheel modeling approach in general and the ceramic bond modeling in particular has to be done for different grinding wheel specifications. Real structure cuts like the one shown in Fig. 17 have to be compared with the modeled structure cuts quantitatively. Parameters for the comparison could be the number of grains per area, the average grain distance or grain distance distributions.

In a next step, we want to model the dressing process which is necessary to restore the grinding performance of a wheel by recreating the concentricity, cleaning the wheel, removing blunt grains and truing the wheel. For this purpose, grain fracturing, grain break-out and bond fracturing have to be taken into account by considering statistical distribution functions determined by experiments or by modeling material properties.

Acknowledgment

The authors would like to thank the German Research Foundation (Deutsche Forschungsgemeinschaft) for the support of the depicted research within the project DA117/24-2, KL500/80-2 “Mathematical Modeling of Grinding Wheel Structures”.

References

[1] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, The Quickhull algorithm for convex hulls, ACM Trans. Math. Softw. 22 (4) (1996) 469-483.

[2] S. Barth, Ermittlung des Einflusses der Spezifikation von CBN-Schleifscheiben auf das Schleifprozessverhalten (Master’s Thesis), Werkzeugmaschinenlabor der RWTH Aachen, 2014.

[3] C. de Boor, A Practical Guide to Splines, Springer, New York, 1978.

[4] M. Botsch, S. Steinberg, S. Bischoff, L. Kobbelt, OpenMesh - a generic and efficient polygon mesh data structure, in: OpenSG Symposium, 2002.

[5] K.-H. Brakhage, M. Makowski, F. Klocke, M. Weiss, Grinding wheel modeling: development of a mathematical model, in: Proceedings of MASCOT11-IMACS/ISGG Workshop, IMACS Series in Computational and Applied Mathematics, 17, 2013, pp. 31-40.

[6] E. Brinksmeier, J.C. Aurich, E. Govekar, C. Heinzel, H.-W. Hoffmeister, F. Klocke, J. Peters, R. Rentsch, D.J. Stephenson, E. Uhlmann, K. Weinert, M. Wittmann, Advances in modeling and simulation of grinding processes, CIRP Ann. Manuf. Technol. 55 (2) (2006) 667-696.

[7] J. Conway, N.J.A. Sloane, Sphere Packings, Lattices and Groups, third ed., Springer, New York, 1999.

[8] D.A. Doman, A. Warkentin, R. Bauer, A survey of recent grinding wheel topography models, Int. J. Mach. Tools Manuf. 46 (3-4) (2006) 343-352.

[9] C. Ericson, Real-Time Collision Detection, Morgan Kaufmann Publishers Inc., San Francisco, 2004.

[10] B. Gärtner, Fast and robust smallest enclosing balls, in: Proceedings of the 7th Annual European Symposium on Algorithms, 1999, pp. 325-338.

[11] M.J. Jackson, Modelling of fracture wear in vitrified cBN grinding wheels, J. of Achiev. in Mater. and Manuf. Eng. 24 (1) (2007) 230-236.

[12] F. Klocke, Manufacturing Processes 2 - Grinding, Honing, Lapping, Springer, Berlin Heidelberg, 2009.

[13] L. Kobbelt, S. Campagna, H.-P. Seidel, A general framework for mesh decimation, in: Proceedings of Graphics Interface, 1998, pp. 43-50.

[14] M. Kremer, D. Bommes, L. Kobbelt, OpenVolumeMesh - a versatile index-based data structure for 3D polytopal complexes, in: Proceedings of the 21st International Meshing Roundtable, 2013, pp. 531-548.

[15] W.E. Lorensen, H.E. Cline, Marching Cubes: a high resolution 3D surface construction algorithm, in: Proceedings of SIGGRAPH, 1987, pp. 163-169.

[16] I.D. Marinescu, M.P. Hitchiner, E. Uhlmann, W.B. Rowe, I. Inasaki, Handbook of Machining with Grinding Wheels, CRC Press, Boca Raton, 2006.

[17] J. Möbius, L. Kobbelt, OpenFlipper: an open source geometry processing and rendering framework, in: Proceedings of the 7th International Conference on Curves and Surfaces, 2010, pp. 488-500.

[18] W. Pietsch, H. Rumpf, Haftkraft, Kapillardruck, Flüssigkeitsvolumen und Grenzwinkel einer Flüssigkeitsbrücke zwischen zwei Kugeln, Chem. Ing. Tech. 39 (15) (1967) 885-893.

[19] M. Rom, K.-H. Brakhage, S. Barth, C. Wrobel, P. Mattfeld, F. Klocke, Mathematical modeling of ceramic-bonded grinding wheel structures, IGPM Preprint 437, 2015.

[20] S. Schumann, T. Siebrecht, P. Kersting, D. Biermann, R. Holtermann, A. Menzel, Determination of the thermal load distribution in internal traverse grinding using a geometric-kinematic simulation, in: Proceedings of the 15th CIRP Conference on Modelling of Machining Operations, 2015, pp. 322-327.

[21] H. Si, TetGen, a Delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw. 41 (2) (2015) 11:1-11:36.

[22] R. Stabenow, Härtewirksame Effekte bei Schleifscheiben, in: Tagungsband: European Conference on Grinding, 2003, pp. 5/1-5/19.

[23] H.K. Tönshoff, J. Peters, I. Inasaki, T. Paul, Modelling and simulation of grinding processes, CIRP Ann. Manuf. Technol. 41 (2) (1992) 677-688.

[24] T. Yamaguchi, M. Higuchi, S. Shimada, N. Matsumori, I. Yoshizawa, H. Ogura, Fractal modeling method for 3D structure of vitrified-bonded wheel, Precis. Eng. 31 (1) (2007) 40-46.