A new computational method for MAT of injected parts integrated in part modelling stage

Vojislav Petrovic, Pedro Rosado, Rafael Torres

A new computational method for MAT of injected parts integrated in part modelling stage

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2008-IJPR-0868
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	26-Oct-2008
Complete List of Authors:	Petrovic, Vojislav; Research & Technological Development Center AIMME, Product Engineering Rosado, Pedro; Technical University of Valencia, Department of Mechanical Engineering; Polytechnic University of Valencia, Department of Mechanical Engineering Torres, Rafael; Technical University of Valencia, Mechanical Engineering and Materials
Keywords:	DESIGN FOR MANUFACTURE, INJECTION MOULDING, PRODUCT MODELLING
Keywords (user):	DESIGN FOR MANUFACTURE, INJECTION MOULDING

Author's details:

First author's name:	Vojislav Petrović
Affiliation:	Research & Technological Development Center AIMME
Address:	Product Engineering AIMME Avda Leonardo da Vinci, 38, 46980 Paterna (Valencia) SPAIN
Telephone:	+34961318564
Fax:	+34961366145
E-mail:	vpetrovic@aimme.es

Second author's name:	Pedro Rosado Castellano
Affiliation:	Professor of Fabrication Processes at the Department of Mechanical and Materials Engineering
Address:	Area de Procesos de Fabricación Departamento de Ingeniería Mecánica y de Materiales Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 VALENCIA (SPAIN)
Telephone:	+34963877622
Fax:	+34963877629
E-mail:	prosado@mcm.upv.es

Third author's name:	Rafael Torres Carot
Affiliation:	Professor of Fabrication Processes at the Department of Mechanical and Materials Engineering
Address:	Area de Procesos de Fabricación Departamento de Ingeniería Mecánica y de Materiales Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 VALENCIA (SPAIN)
Telephone:	+34963877627
Fax:	+34963877629
E-mail:	rtorres@dimm.upv.es

A new computational method for MAT of injected parts integrated in part modelling stage

Vojislav Petrović†, Pedro Rosado†, Rafael Torres‡

Abstract: In this paper we present a simple and fast approach for MAT generation in discrete form. It is used for manufacturability analysis in part modelling stage of injected parts. The method is a volume thinning method based on straight skeleton computation, modified and applied in 3D on B-rep models in STL. The volume thinning of B-rep model is based on its boundary surfaces offset towards model interior. The surfaces' offset is done with an adequately proposed offset distance which makes some of non adjacent offset model surfaces overlap (they "meet" in mid-surface or MAT). Offset surfaces are used to reconstruct the topology of a new B-rep model (offset model). Overlapping surfaces in offset model are detected, separated and aggregated to MAT. For adequate MAT precision and adequate MAT radius function, we propose to treat B-rep model concave edges (vertices) as cylinders (spheres) of zero-radius and offset them in adequate way. On these bases, we present an iterative algorithm in which MAT is being constructed in incremental way by consecutive volume thinning of obtained offset models. MAT construction is finished when an empty offset model is obtained. An algorithm has been created and implemented in Visual C++. Some of obtained results are presented in this paper.

† Department of Mechanical and Materials Engineering, Technical University of Valencia, 46022 Valencia, Spain

Introduction

Medial Axis (MA) and Medial Axis Transform (MAT) are thoroughly investigated and analyzed geometrical terms with very diverse scientific and engineering use. Many definitions of MAT can be found in scientific literature. Blum (1967) introduced MAT back in 1967, indicating its enormous capacity of shape abstraction. Sherbrooke et al (1996) offers a very precise definition of MAT. For a subset of 2D or 3D Euclidian space, denoted as S, Medial Axis (MA), denoted MA(S), is the locus of points inside S which lie at the centres of all closed discs (balls in 3D) which are maximal in S. The radius value function of MA(S) is a continuous, real-valued function defined on MA(S) whose value at each point on the Medial Axis is equal to the radius of the associated maximal disc (ball in 3D). The Medial Axis Transform (MAT) of S is the MA(S) together with its associated radius function. Though, MAT 2D is a set of lines while MAT 3D is a set of surfaces. MAT 3D is very appropriate for injected parts representation since it has an explicit thickness distribution. MAT 3D has been proposed as a useful shape abstraction tool (Quadros (2001)) in mould and die design, mesh generation, motion planning, etc. MAT 2D can be usefully applied in motion planning, flow analysis (Petrovic (2005)), etc. A model reconstruction based on MAT 3D (Amenta et al (2001)) is very important in visual graphics and animation. Mid-surface as a part of MAT 3D was successfully used in geometry recognition Locket et al (2005).

In this paper, a simple and fast approach for injected parts discrete MAT computation, for the purpose of manufacturability analysis (Petrovic (2008)), is presented. Many existing solutions to MAT generations are available. However, the

contribution of the method developed in this paper is reflected in two aspects: time consumption and a clear geometrical and topological definition of generated MAT. In order to obtain model's discrete MAT, injected parts are represented by also discrete B-rep model exported in STL format. By discrete MAT, we understand a representation in which continuous free-form MAT surfaces are represented by a whole of plane surface patches. The advantages of B-rep model are various. First of all, no matter what CAD modeller was used to create a B-rep model, it has the same definition, which is important when MAT computation tool is integrated with CAD tools. Further more, it is relatively easy to perform offset of plane boundary surfaces which a discrete B-rep model in STL is made of. Finally, although MAT is a geometrical representation without volume, it can be written in STL format. This way we can use the same data format for input (B-rep model) and output (MAT). It is important since the proper design for manufacturability analysis can only be performed over a MAT with fully defined geometry and topology.

The proposed approach is a step-by-step volume thinning method based on straight skeleton computation in 2D (offered by Cacciola (2007)). Some of straight skeleton concepts are modified and applied in 3D on a discrete B-rep model (STL format) of injected parts. In this way, more detailed solution of MAT is generated with concave elements (edges and vertices) properly offset. Volume thinning is very suitable in case of injected parts due to their thin-walled character. In addition, properly designed injected parts have uniform thickness distribution. Hence, there is no need for many volume thinning steps and volume thinning algorithm can be performed in reduced time.

Figure 1 illustrates how straight skeleton is constructed in 2D. A contour is offset towards its interior (Figure 1a) and offset contours are reconstructed so as to obtain straight skeleton (Figure 1b). Accordingly, in 3D we perform volume thinning by offsetting surfaces of discrete B-rep model towards model interior. Hence, a new B-rep model, denoted as offset model, is reconstructed by using obtained offset surfaces. However, for adequate MAT precision and adequate MAT radius function, we propose to treat B-rep model concave edges/vertices as cylinders/spheres of zero-radius. When offset, zero-radius cylinders/spheres are converted to real value radius cylinders/spheres. Radius is then equal to the distance used for offset. This allows us to construct MAT with bigger precision, as it is shown on Figure 2. The level of precision depends on number of plane surfaces that we use to discretize cylinders/spheres. However, computational time is increased if bigger number of plane surfaces is used.

[insert Figure 1 about here]

We choose to perform the offset with distance that is not random and constant like in straight skeleton computation. We opt for an accurately proposed offset distance. As it is explained in following section, this distance is computed in a way that it makes some of non adjacent offset model surfaces overlap (they "meet" in mid-surface or MAT). In this way we reduce the number of necessary offset steps. Overlapping zones are then detected, separated and aggregated to MAT.

[insert Figure 2 about here]

On these bases, we present an iterative algorithm in which MAT is being constructed in incremental way by consecutive volume thinning of obtained offset models. After each step, a new offset model is obtained with some overlapping zones, which are aggregated to MAT. The vertices of overlapping zones are assigned with the offset distance accumulated in previous offset steps. This accumulated distance represents a local thickness in each vertex, which is used to establish a radius function (thickness distribution) when MAT is completed. MAT construction is completed when an empty offset model is obtained.

Related work

As stated before, MAT 3D has many applications. MAT represents a solid model reduced in one dimension which is why it results easier to manage. Many researchers have worked in this area and many useful works are available. A continuous MAT 3D has a very high computational time. Therefore, for engineering applications a discrete form of MAT 3D consisted of plane surfaces is more viable. In continuing lines, we refer to principal MAT computing approaches and representative pieces of work.

One of principal approaches in MAT computation is a tracing approach (Sherbrooke (1996), Turkiyyah (1997)). The algorithm consists in tracing *seams* from model *vertices* which intersect in so-called *junction points*. The junction points limit sheets that medial axis is formed of. The procedure is repeated recursively until all sheets of the medial axis have been traced.

Another approach in MAT computation is a computation of Delaunay Triangulation and its transforming to Voronoi Diagram. As Sherbrooke states (Sherbrooke (1996)), MAT is equivalent to the boundary of Constrained Voronoi Diagram cells. Therefore, if

the Delaunay Triangulation of a set of objects is computed and transformed to the Voronoi Diagram, as its dual graph, MAT can be obtained. This procedure is mostly applied on a discrete solid model represented by a point cloud and a continuous or approximated MAT is computed (Hubbard (1996)). This concept is applicable in 2D also and not only on set of points, but on a set of line segments (Segment Voronoi Diagram) too (Karavelas (2004)).

Another interesting approach is to compute a discrete Voronoi diagram by rendering a three dimensional distance mesh for each member of a set of 2D/3D objects using graphics hardware (OpenGL) (Hoff III (1999)). It is a very interesting idea of using graphics hardware to accelerate the process of Voronoï Diagram computing and it is reported to be fast and efficient in 2D.

Yang et al (Yang (2004)) propose an iterative algorithm for MAT computing that relies on two primitive operations. The first operation identifies an initial point on MAT by tracing a maximal sphere of an arbitrary interior point. The sphere intersects MAT in a number of points which are added to MAT and serve as centres of new spheres. In that way, MAT grows in incremental way. The search of sphere intersection with MAT is based on distance query which is accelerated by using the PQP package. The authors report a very brief computational time and a possibility of accuracy-computational time trade-off.

The experience of the authors regarding the use of existing CAD/CAM systems in early stages of injected parts design is negative. There are two principal reasons for that. The first one is the mathematical definition of MAT. The existing solutions generate MAT as a set of points or triangular elements. Hence, this MAT needs additional steps of data reworking to generate topological and geometrical information of generated

MAT (for instance, a proper moulding analysis can not be done using MAT without completely defined topology of MAT surfaces). Secondly, the time consumption in MAT generation by existing methods is unacceptable for a real time analysis in early design stage. For engineering applications, such as DFM analysis, it is possible to have a precision to time consumption trade-off and yet have applications that are not affected at all by the precision loss. That is why we have designed an algorithm to perform the offset method over an STL format of an injected part model. It enables computation of MAT with less precision but with a proper topology and geometrical definition, completely valid for injected parts DFM analysis (see Amenta (2001)).The generated MAT is then used for a real time DFM analysis includes moldability, mould filling, welding lines and injection pressure (clamping force) analysis.

Proposed approach for MAT computation

As it was mentioned, an iterative algorithm is presented in which MAT is being constructed in incremental way by consecutive volume thinning of obtained offset models. In every step of iteration, we proceed with three sets of operations. The first set includes operations for determining a proper offset distance denoted as maximum offset distance. The second set consists of operations necessary for model offsetting and reconstruction of offset model topology. The final set of operations detects overlapping zones of offset model, separates and aggregates them to MAT structure. The offset model, free of overlapping zones, can be offset again until the final offset model is empty.

Offset distance computing

When the offset is made, model surfaces are displaced towards model interior. Every face has one or more loops of edges with their topological orientation. The orientation of edges is in accordance with surface topological normal which indicates model exterior. Every edge in a B-rep model is shared by two model surfaces. An edge is said to be concave if the vectorial product of those two surfaces' normals is opposite to the edge orientation (Figure 3). When part model surfaces (S) are offset and a new offset model is reconstructed upon offset surfaces (S_{off}), the portion of surface limited by a loop of concave edges will not change its size. However, if a loop contains at least one convex edge, the portion of offset surface after reconstruction will be reduced if it is an exterior loop (EL). Yet, it will expand, if it is an interior loop (IL). The "contraction" of exterior loops and the "expansion" of interior loops imply that, if the offset distance is increasing, there is a moment in which exterior and interior loops of the surface overlap (Figure 4). Any further displacement of model surfaces would cause offset model incongruence due to intersection of reconstructed exterior and interior loops of offset surfaces.

[insert Figure 3 about here]

[insert Figure 4 about here]

Hence, in order to determine the exact value of maximum offset distance and prevent possible model incongruence, we offer following analysis. Figure 5 shows a surface S1 together with its four neighbours N1-4. Each pair of neighbour surfaces defines a bisector plane (B) which contains the edge shared by those two surfaces. Those bisectors intersect and derive directions that we denote as offset directions (A). Offset directions start at loop vertices. Points of every bisector are equally distanced from two surfaces that define that bisector. Therefore, the offset surfaces intersect at the same

bisector. That means that the edges of reconstructed loop, no matter the offset distance magnitude, keep laying at the corresponding bisector. Also, the loop vertices keep laying at corresponding offset directions. The critical moment is when two offset directions intersect (P_{int}). The edge defined by two points (P_{C1N3N4} and P_{C1N3N4}) is being reduced while being offset and finally it is converted to a point, P_{int}. In that moment the loop of S1 is offset in extremis. If we continue offsetting, the points swap their position and the loop overlaps itself (dashed line on Figure 5). Therefore, the distance between P_{int} and the original S1 is the maximum offset distance regarding the edge E1. For each of loop edges, a distance is sought and the shortest of all distances is kept as the proper one for S1. This is done for each of model surfaces and finally a minimum of all maximum distances is kept.

[insert Figure 5 about here]

Each offset direction is created by intersection of two bisectors. If a vertex is shared by three surfaces, corresponding three bisectors intersect at the same offset direction (geometrically it is the axe of a cone tangent to all three surfaces, f.e. N4, N5 and S1 on Figure 6). Nevertheless, if a vertex is shared by more than three surfaces (N2, N3, N4 and S1 on Figure 6), there will be more than one offset direction per vertex. However, for each edge, only one direction per vertex is valid and it results in an offset distance.

[insert Figure 6 about here]

In case of concave vertices, for each of them the shortest distance to model surfaces is sought. We consider only model surfaces for which the vertex is “interior” (the vertex projection on surface is situated at interior side of the surface, regarding the surface topology). One of vertex-to-surface distances of all vertices is the shortest one. Its *half-value* is the maximum offset distance *for concave vertices*.

Offsetting

After the offset distance has been determined, the model is offset. B-rep model plane surfaces result from the grouping of initial triangular plane surfaces of the model in STL. Therefore, B-rep model plane surfaces are limited by exterior and interior loops of edges, connected by vertices. In order to offset a model surface, we establish a vector of the same direction as the surface topological normal, but of opposite orientation (Figure 7). Each loop vertex is displaced along that vector by the offset distance. In that way, we offset any surface when we offset its edges by displacing all vertices.

[insert Figure 7 about here]

The loops of the obtained offset surface are just temporary: the proper loops are reconstructed by intersecting of the offset surface with its also offset neighbours (Figure 8). Provisional loops play an important role in limiting of intersection lines (IL). For the purpose of proper loops reconstruction, an infinite intersection line of the offset surface and a neighbour is restricted by these two surfaces' initial loops (Figure 9a). When the surface and its neighbours are offset, the offset surfaces are intersected, resulting in a number of limited intersection lines. The offset neighbours have been organized in a counter-clockwise order (neig. 1-8), considering also the neighbours created by concave edges and vertices offset. Though, limited intersection lines corresponding to all considered neighbours are organized in counter-clockwise order (IL1-8), too. Each line is then intersected with its previous one in row to obtain a proper initial vertex. Likewise, the line is intersected with its following one to obtain a proper final vertex (Figure 9b). Some of intersection lines overlap partially or totally, so the overlapping part is eliminated from the loop. The initial and final vertices are then used to determine the line orientation vector. Finally, we have a proper reconstructed loop (offset loop)

with properly oriented edges defined by an initial and a final vertex. Reconstructed loops establish the limits of reconstructed offset surface.

[insert Figure 8 about here]

[insert Figure 9 about here]

Hence, all model surfaces are offset and, for each of them, a counter-clockwise organized list of neighbours is determined, one per each of surface loops. Afterwards, the offset surface is intersected with their offset neighbours. A list of intersection lines is then created in the same organized order and used to reconstruct corresponding offset surface loops.

MAT & offset solid determination

Once all model surfaces are offset and their loops are reconstructed, the offset model is completed (Figure 10). According to 'Offset distance computing' section, one or more edges are converted into a point when offset. Note that more than one edge can be characterized by the same maximum offset distance and, after offset, all of them are reduced to a point. They even may belong to the same loop which causes that some of offset surfaces disappear (Figure 11). Therefore, non adjacent surfaces may become neighbours after offset and may even overlap. Hence, the offset model must be checked for overlapping surfaces. Surface overlapping zones are then separated and aggregated to MAT.

[insert Figure 10 about here]

[insert Figure 11 about here]

Each of model surfaces is analyzed for overlapping with other surfaces. So as to filter out unnecessary checking, only surfaces of opposite topological normal and situated in

an infinite plane of the same geometrical parameters are considered. If the surface doesn't overlap with any other surface, it is copied to a new offset model. If it does overlap, its overlapping part is detected, separated and aggregated to MAT model, while the rest of the surface is aggregated to the new offset model. Also for all convex edges, a surface that connects its two vertices with two corresponding offset vertices is constructed and also aggregated to MAT (Figure 12b).

[insert Figure 12 about here]

Algorithm

According to above exposed methodology, we propose an iterative algorithm. It is performed on a solid model with a structure shown on Figure 13. The model structure consists of planes, loops, edges and vertices, organized in hierarchical way. Solid has a direct relation with all its planes, edges and vertices. The algorithm steps consist of following procedures:

1. *Model importation*. A B-rep model of an injected part is designed in any CAD modeller capable of exporting it in STL format. B-rep in STL is consisted of plane triangular surfaces, representing an approximation of part's free-form surfaces. Model data are read form the STL file and organized in a structure that represents a solid model. In this structure, a solid model has a direct relation with all its planes, edges and vertices (full line connectors on Figure 13).

[insert Figure 13 about here]

2. *Model surfaces grouping*. Many triangular surfaces of B-rep in STL can be grouped in a single planar surface. In that way, we obtain one surface with various loops of multiple edges instead of larger number of triangular surfaces

with one loop of three edges (Figure 12a). In that way, we manage fewer surfaces when the model is offset and we reduce MAT computation time.

3. *Inverse elements relating*. In order to perform necessary offset operations, inverse relations of the solid structure elements are formed (as shown by dashed connectors on Figure 13). With the model structure defined, a model plane "knows" which are its loops, edges and vertices. By forming inverse relations, a loop gets to "know" which plane it belongs to. Also, edges/vertices get to know which loops/edges share them. Inverse relations are essential in establishing of offset directions, in determination of surface neighbours and their order, etc.

4. *Evaluation of solid elements convexity*. As stated in 'Offsetting' section, the offset of concave edges are cylinders and the offset of concave points are spheres. The resulting cylinders and spheres are discretized in planar surfaces. Hence, previous to offsetting, we must determine which edges and vertices are concave so as to offset them later. It was mentioned before that an edge is said to be concave if the vectorial product of their two surfaces normals is opposite to the edge orientation. In case of a vertex, it is considered concave if all edges that are starting or ending in that vertex are concave.

5. *Search for offset directions*. For each of *convex* vertices, offset directions are found as exposed in Offset distance computing chapter.

6. *Search for maximum offset distance*. For all solid model edges, a maximum offset distance is sought. If both of its vertices are convex, an intersection is sought for all combination of offset directions (one of initial vertex and the other of final vertex). Finally, there will be one minimum distance per edge and, among all

solid model edges, one minimum distance. The later represents the maximum offset distance *for convex vertices*. For *concave vertices,* a vertex-to-plane shortest distance search is performed, as exposed in previous section. Finally, the distance used for offsetting in this step is the minor value of the maximum offset distance for convex and concave vertices.

7. *Model offsetting*. The model is offset and its topology is reconstructed as exposed in 'Offsetting' section.

8. *MAT aggregation*. After offsetting, an offset model is obtained. An overlapping check is performed, as commented in previous section, and overlapping zones are added to MAT structure, as well as the plane surfaces defined by original and offset convex vertices.

The new offset model, free of overlapping zones, is used to repeat procedures 2-8. This iterative process ends when the obtained offset model is empty (it has zero surfaces).

Results & discussion

The exposed algorithm has been implemented in Visual C++. Classes for each of solid model elements (plane, loop, edge, vertex and vector) have been created. All algorithm procedures are implemented in corresponding functions. The created code has been compiled, linked and executed on PC processor Intel Centrino 1.4GHz with 256 Mb RAM. It has been tested on real industry parts, which B-rep model was exported in STL with up to 1000 planes (a decent model precision, sufficient for manufacturability analysis). We present three examples with the corresponding model, offset steps and final MAT on Figure 14-Figure 16. Note that what is shown for each model corresponds

to mid-surface which is a part of MAT (Petrovic (2008)). It is done for two reasons: mid-surface is used rather than MAT in manufacturability analysis and it can be related visually with corresponding model more clearly than MAT.

[insert Figure 14 about here]

[insert Figure 15 about here]

[insert Figure 16 about here]

In the following Table some basic information about examples and their MAT computing time is offered. The table shows that the key factor of computing time is not the number of B-rep model planes, but the number of necessary offset steps. That is why injected parts with their uniform thickness distribution are suitable for fast volume thinning. Therefore, computational time is relatively low if compared with some other methods (Yang (2004)). However, it guarantees low computational time in thin-walled part design while it may not be so superior in general application.

[insert Table 1 about here]

Conclusions

In this paper, a simple and fast approach for injected parts discrete MAT computation, useful in manufacturability analysis, is presented. In order to obtain its discrete MAT, injected parts are represented by also discrete B-rep model exported in STL format. The proposed approach is a step-by-step volume thinning method, based on straight skeleton computation in 2D. Some of straight skeleton computation concepts are modified and the modified concept is applied in 3D on a discrete B-rep model (STL format) of injected part. Volume thinning is very suitable in case of injected parts due to

their thin-walled character. In addition, properly designed injected parts have a uniform thickness distribution. Hence, there is no need for many volume thinning steps and the volume thinning algorithm can be performed in reduced time.

This MAT generation solution comes as an answer to important shortcomings of existing solutions regarding elevated time consumption and lack of geometrical and topological definition. The principal field of application of the proposed approach is fast mid-surface computation in injected parts for its manufacturability analysis. MAT, as the authors generate it, has some limitations due to its lower precision. Yet, it has much less data than a solid model and it enables design analysis in much less time. Also, generated MAT is capable of offering sufficient data for principal manufacturability aspects analysis (parting directions analysis, fabrication cycle time, uniform thickness analysis) since these aspects can be well analyzed disregarding the lower precision. The 'design for manufacturability' analysis is performed in part modelling stage, which is why reduced time is so important. .

References

Amenta (2001) Amenta N, Choi S and Kolluri R, The power crust, unions of balls, and the medial axis transform. Computational Geometry: Theory and Applications. 2001; 19:(2-3):127-153

Blum (1967) A transformation for extracting new descriptors of form in Models for the Perception of Speech and Visual Form. W. Whaten-Dunn (Ed.). MIT Press: Cambridge, MA. 1967; 362-380.

Cacciola (2007) Cacciola, F. 2D Straight Skeleton and Polygon Offsetting. In CGAL Editorial Board, editor, CGAL User and Reference Manual. 3.3 edition. 2007;

Hoff III (1999) Hoff III K et al. Fast Computation of Generalized Voronoï Diagrams Using Graphics Hardware. Proceedings of SIGGRAPH 99, Los Angeles. 1999.

Hubbard (1996) Hubbard P. Approximating polyhedron with spheres for time critical collision detection. ACM Transactions on Graphics. 1996; 15(3):179–210.

Karavelas (2004) Karavelas M. A robust and efficient implementation for the segment Voronoi diagram. Int Symp on Voronoi Diagrams in Science and Engineering. 2004; 51-62.

Locket (2005) Lockett H, Guenov M. Graph-based feature recognition for injection moulding based on a mid-surface approach. Computer-Aided Design. 2005; 37:251–262

Petrovic (2005) Petrovic V, Rosado P. A geometric approach for injection mould filling pressure estimation in 2D. Proceed of 3rd Int Conf on Comp Aided Des and Manuf, Supetar, Croatia. 2005; 59-61.

Petrovic (2008) Petrovic V, Rosado P. Manufacturability analysis of injected parts based on a mid-surface approach. Journal of Engineering Design. 2008;

Quadros (2001) Quadros WR et al. Skeleton for Representation and Reasoning in Engineering Applications. Engineering with Computers. 2001; 17:186-198

Sherbrooke (1996) Sherbrooke E C, Patrikalakis N M, Brisson E. An algorithm for the medial axis transform of 3d polyhedral solids. IEEE Transactions on Visualization and Computer Graphics. 1996; 2(1):44–61

Turkiyyah (1997) Turkiyyah G M, Storti D W, Ganter M, Chen H and Vimawala M. An accelerated triangulation method for computing the skeletons of free-form solid models. Computer Aided Design. 1997; 29:1:5–19.

Yang (2004) Yang Y., Brock O., Moll R. Efficient and Robust Computation of an Approximated Medial Axis. ACM Symposium on Solid Modelling and Applications. 2004

List of figure captions

Figure 1. Figure offsetting (a) and straight skeleton construction (b) in 2D

Figure 2. Offsetting and MAT 2D construction with modified concave elements treatment

Figure 3. Concave and convex edge

Figure 4. Loops overlapping when offset

Figure 5. Search for offset distance

List of Tables

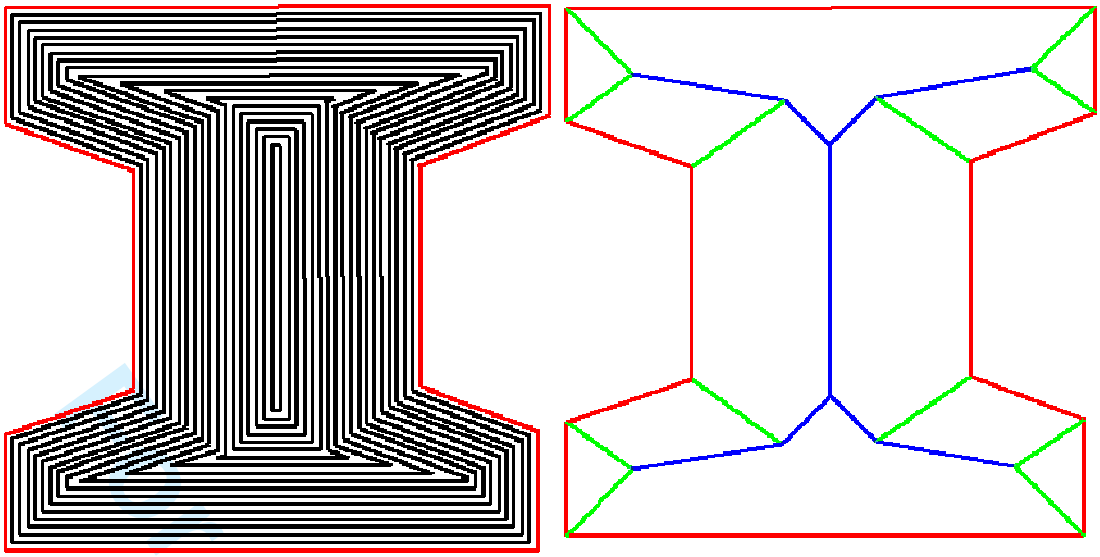


Figure 1. Figure offsetting (a) and straight skeleton construction (b) in 2D

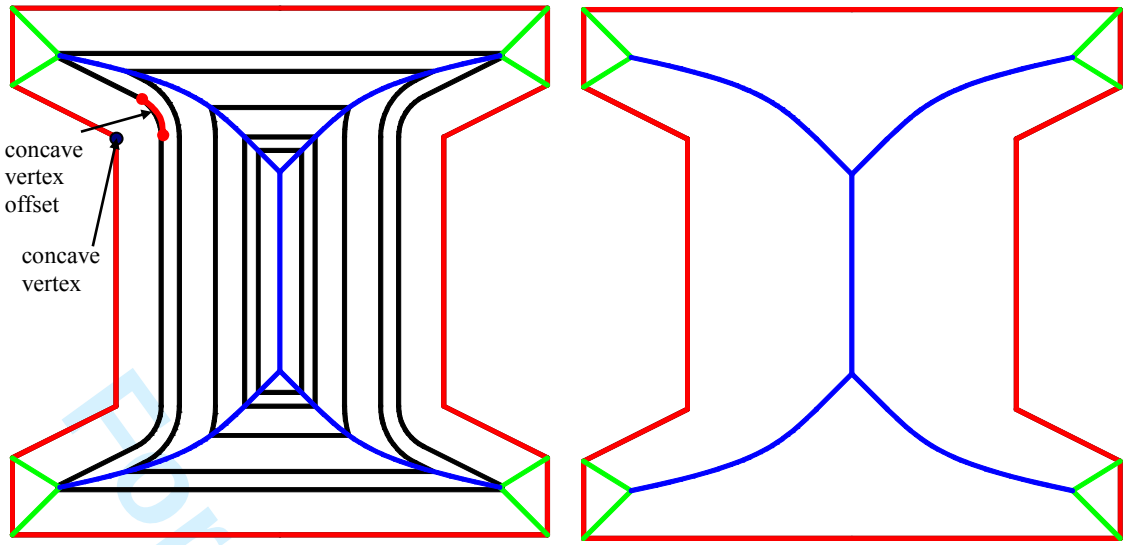


Figure 2. Offsetting and MAT 2D construction with modified concave elements treatment

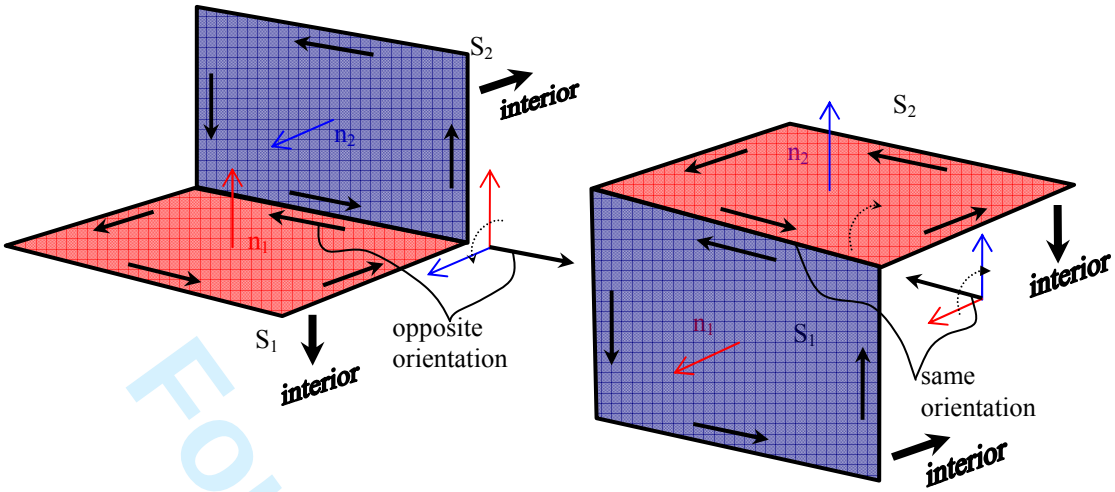


Figure 3. Concave and convex edge

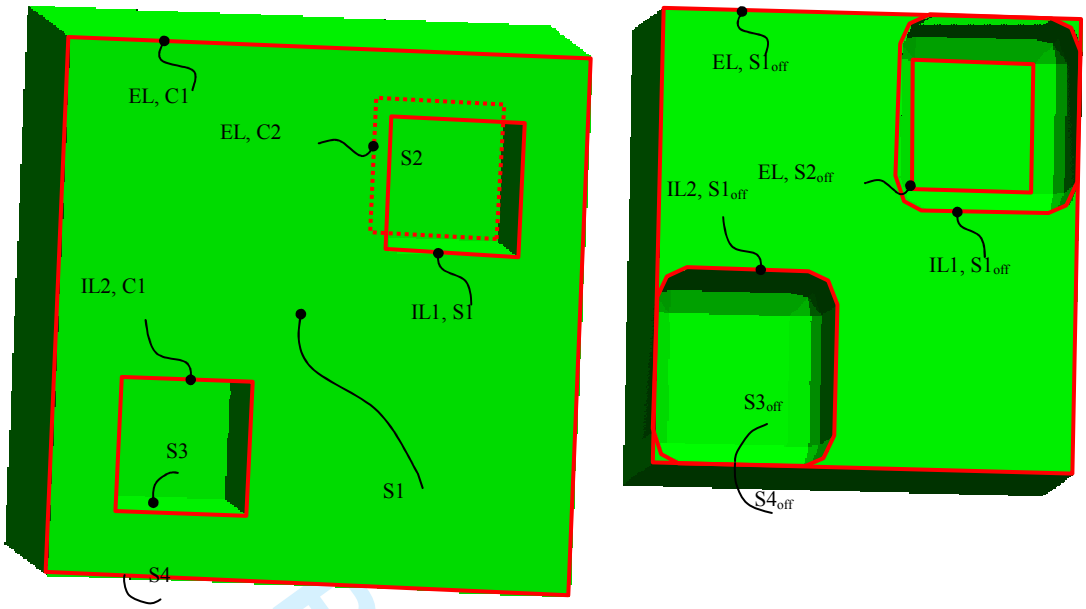


Figure 4. Loops overlapping when offset

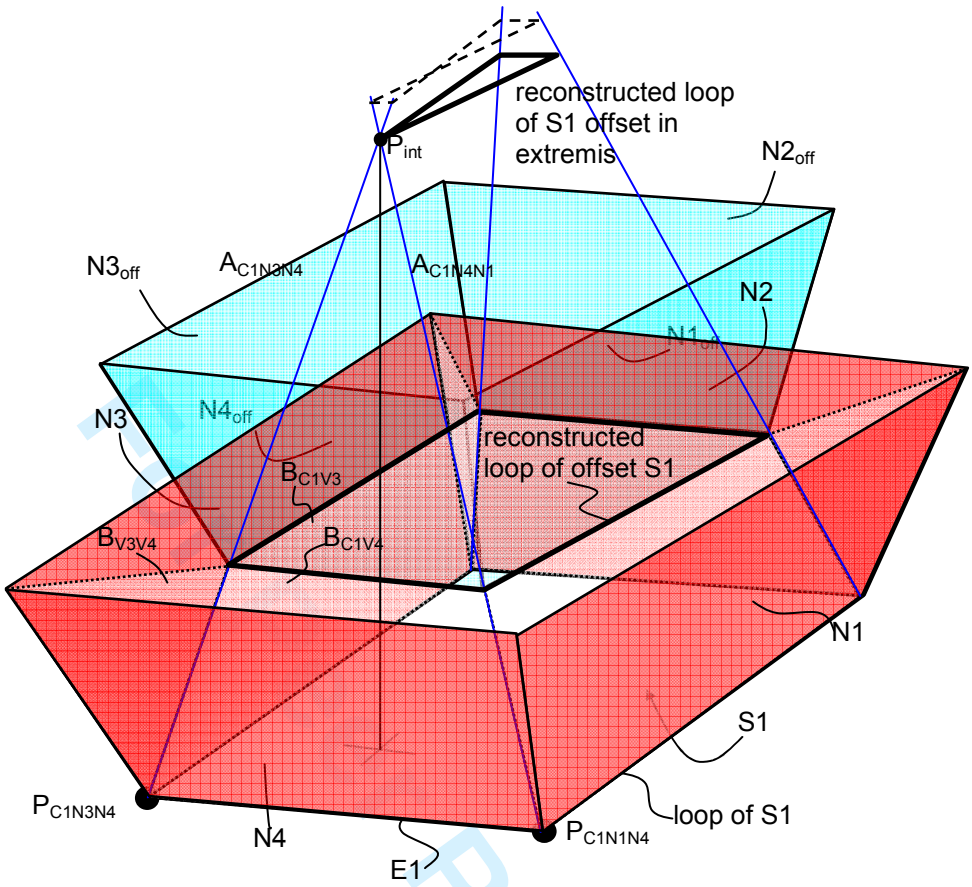


Figure 5. Loops overlapping when offset

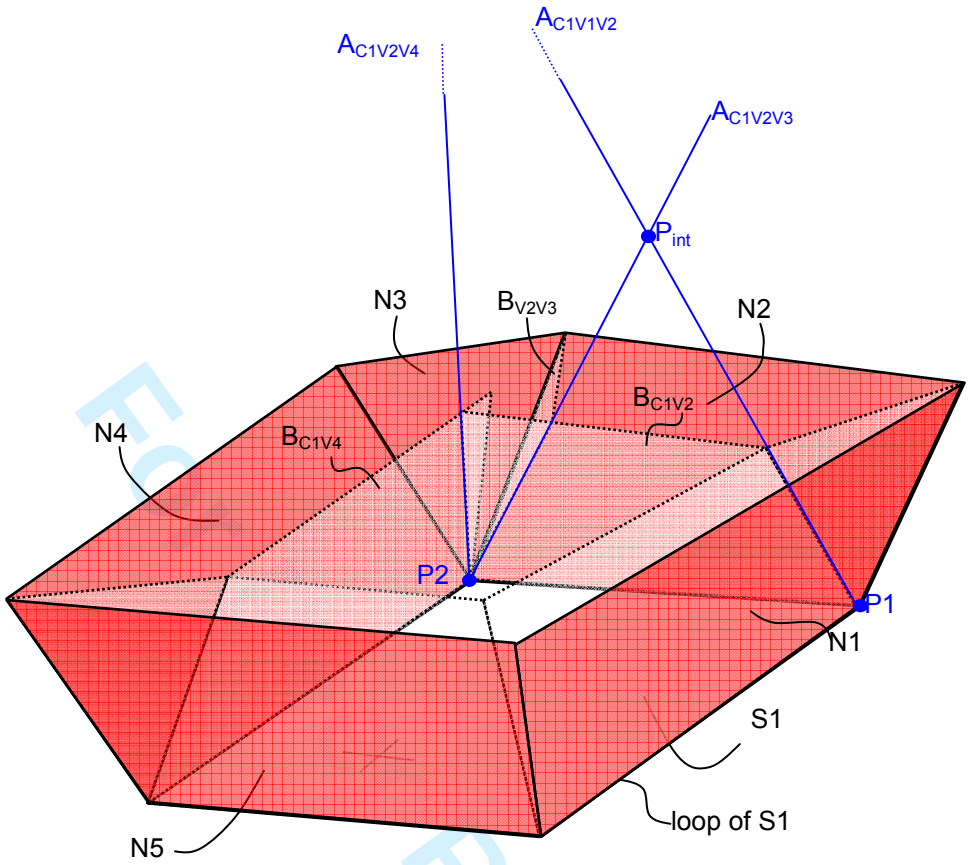


Figure 6. Multiple offset directions

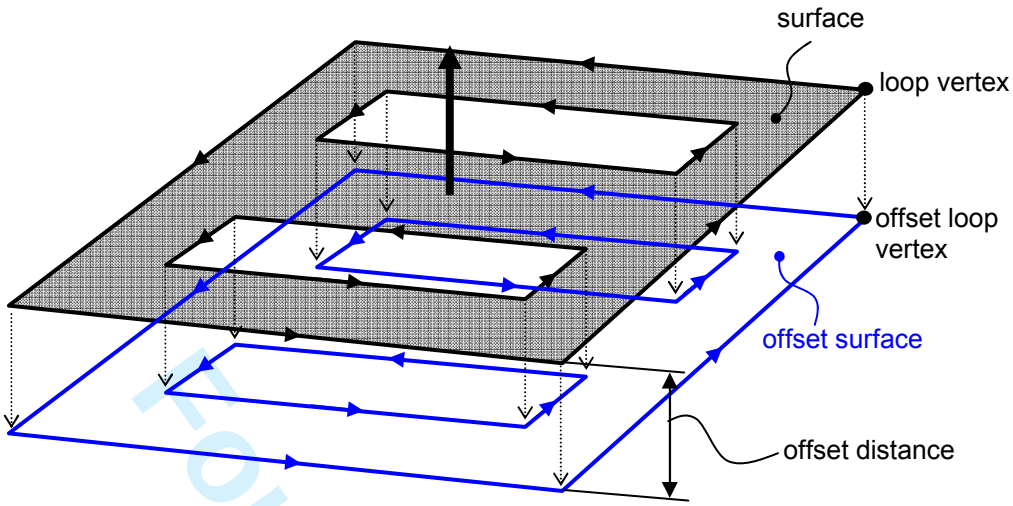


Figure 7. Model surface offset

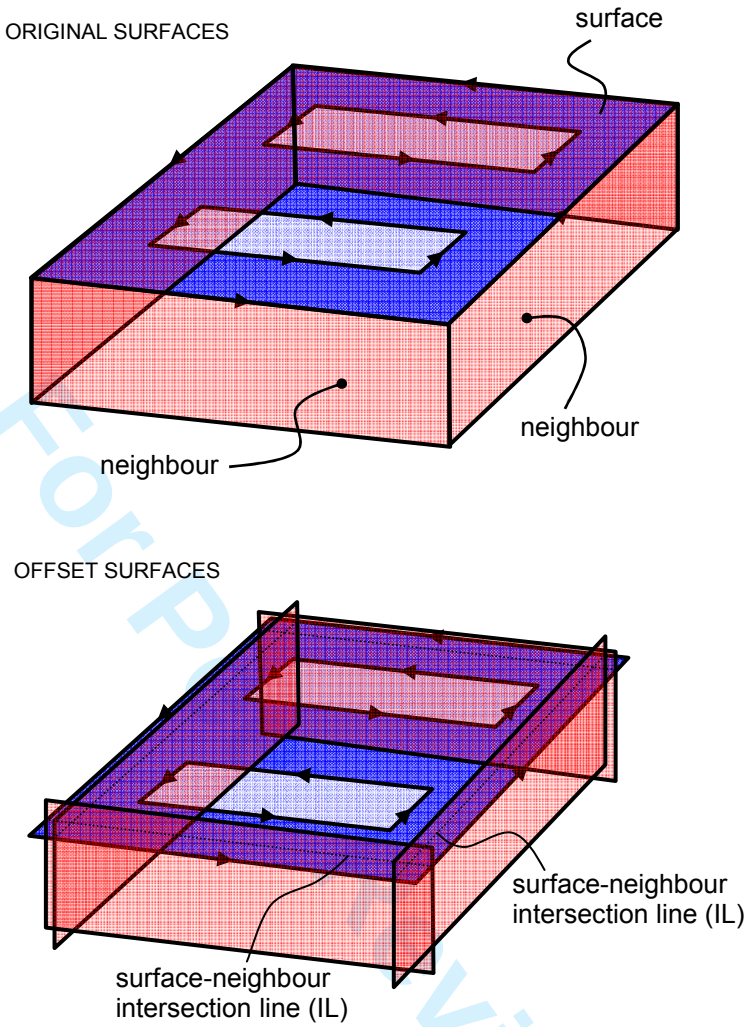


Figure 8. Surface-neighbours intersection

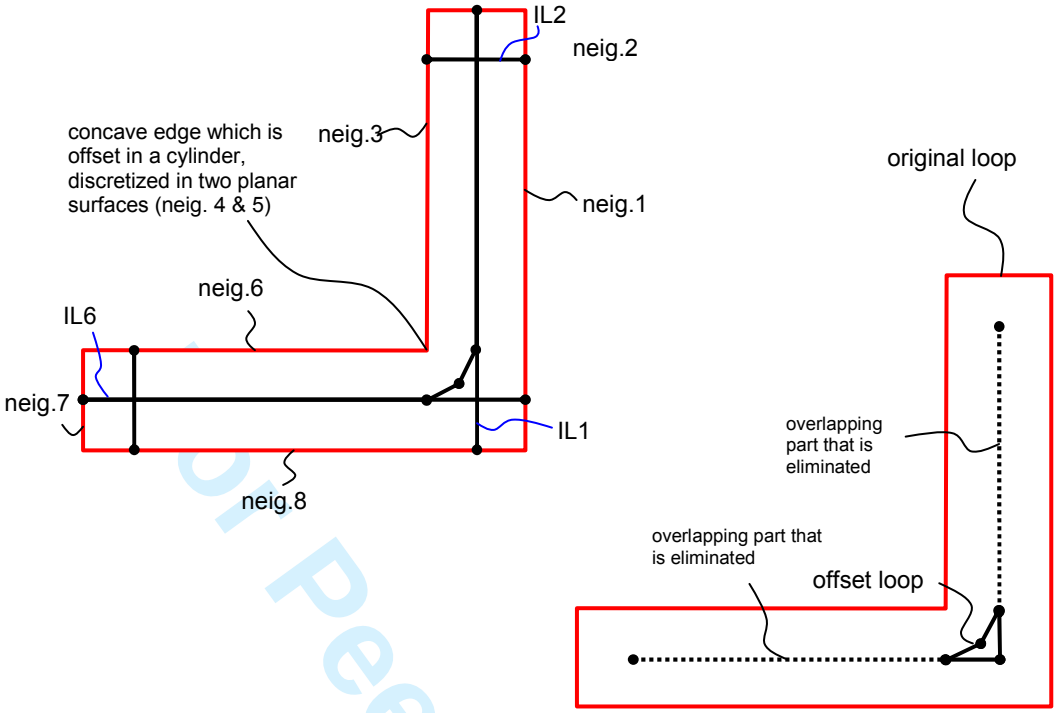


Figure 9. Surface loop reconstruction

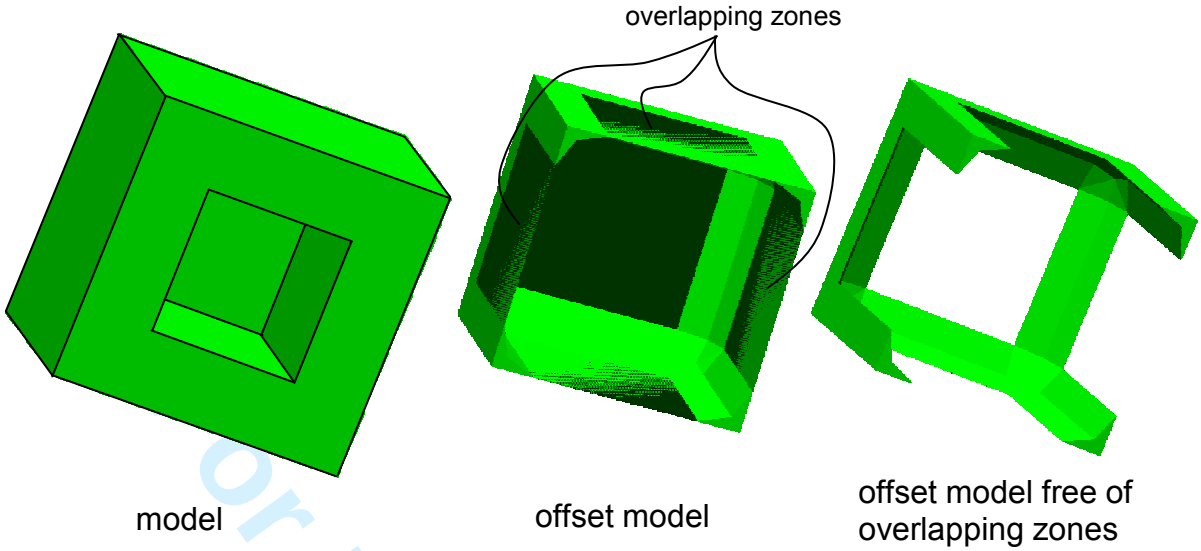


Figure 10. Overlapping zones detection

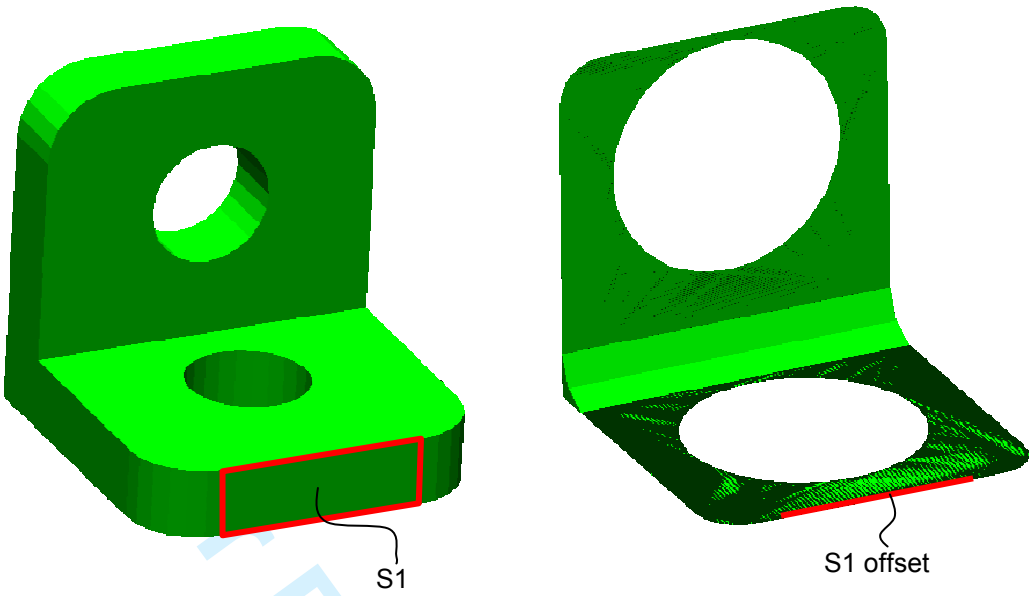


Figure 11. Surface that disappears when offset

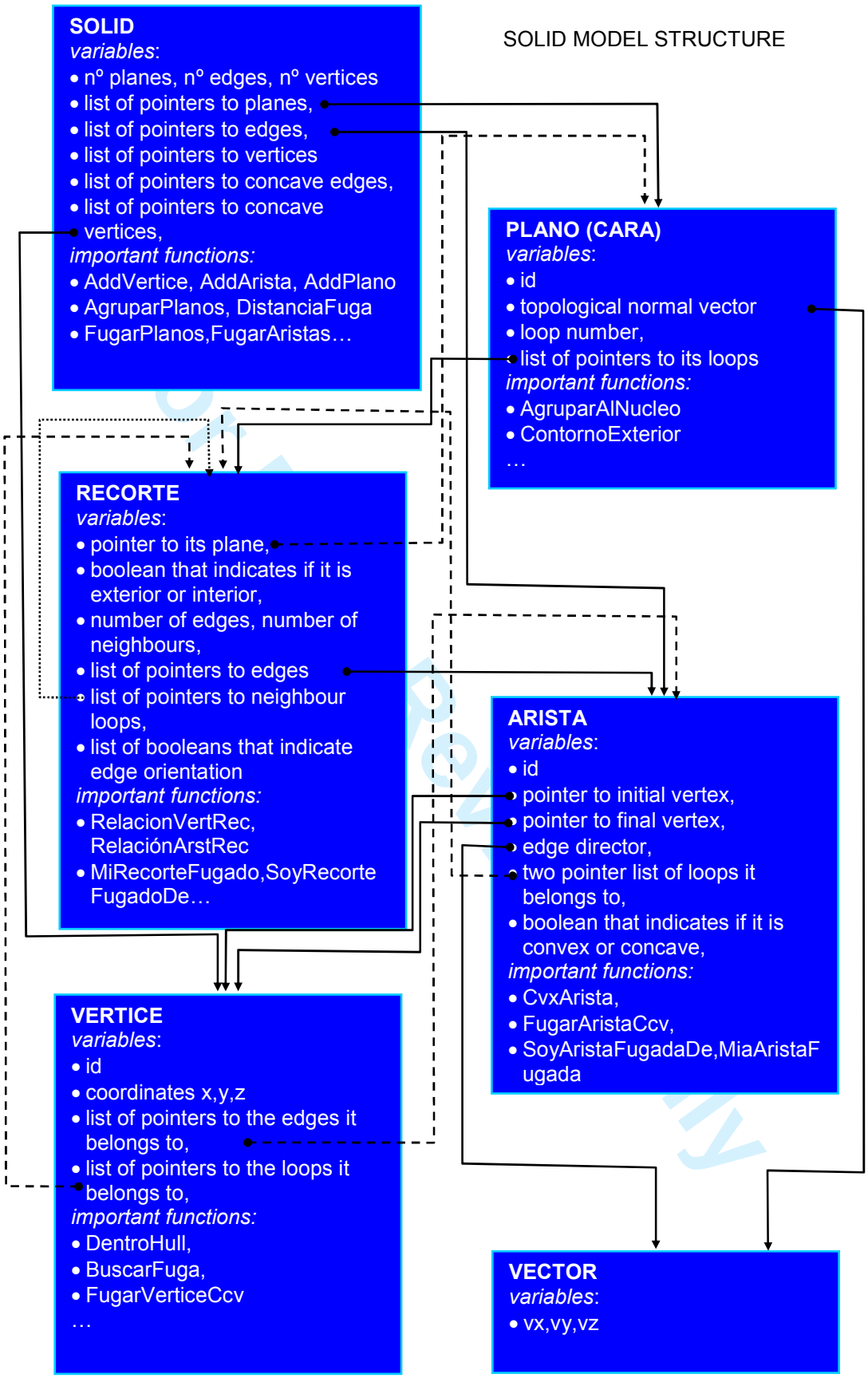


Figure 12. Solid model structure

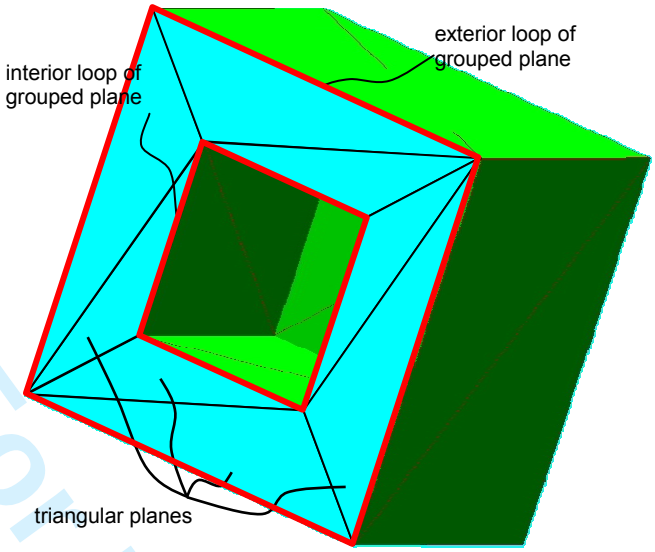


Figure 13. Model surfaces grouping

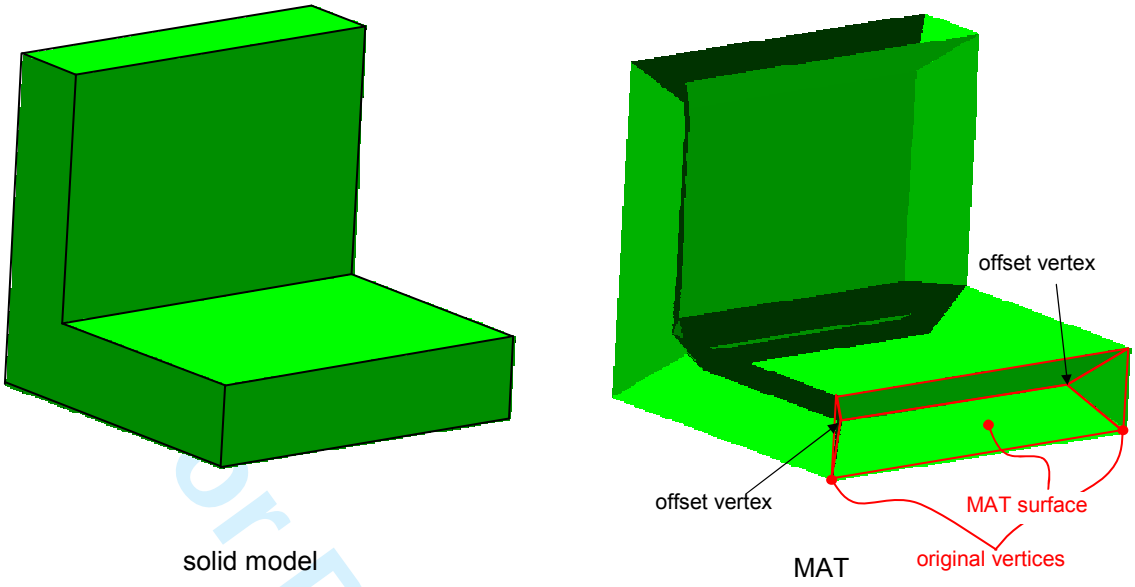


Figure 14. MAT surfaces that unite convex vertices

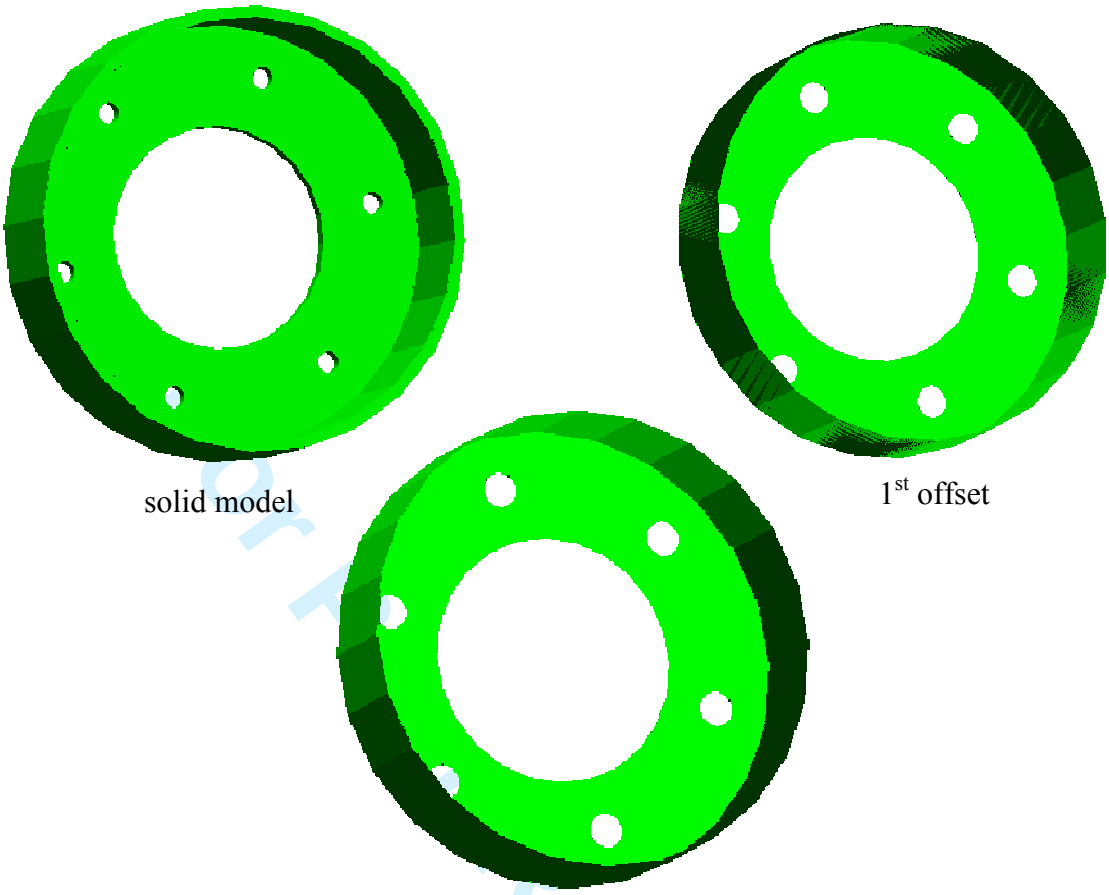


Figure 15. Part 1 – solid model, some offset steps and final mid-surface

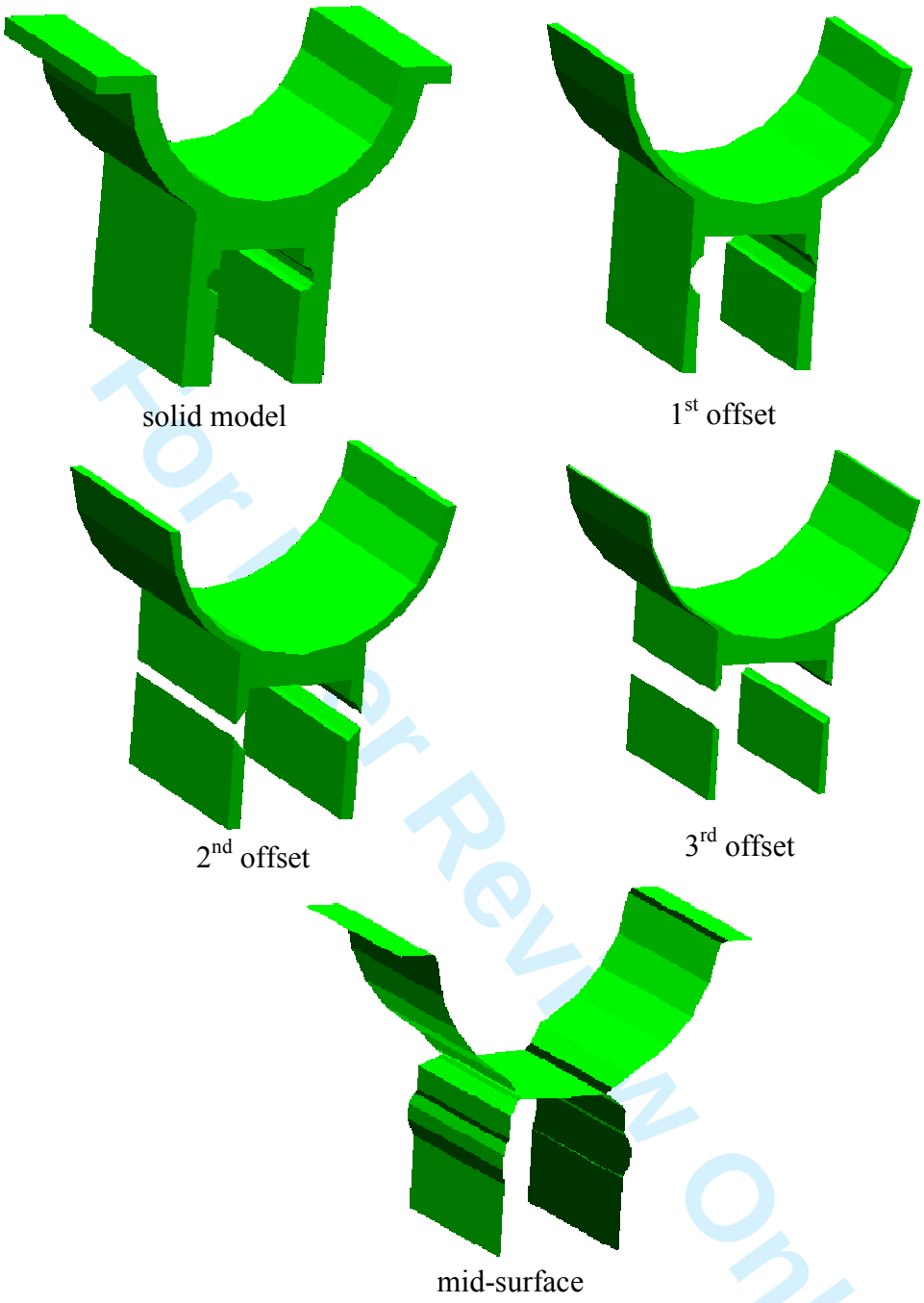
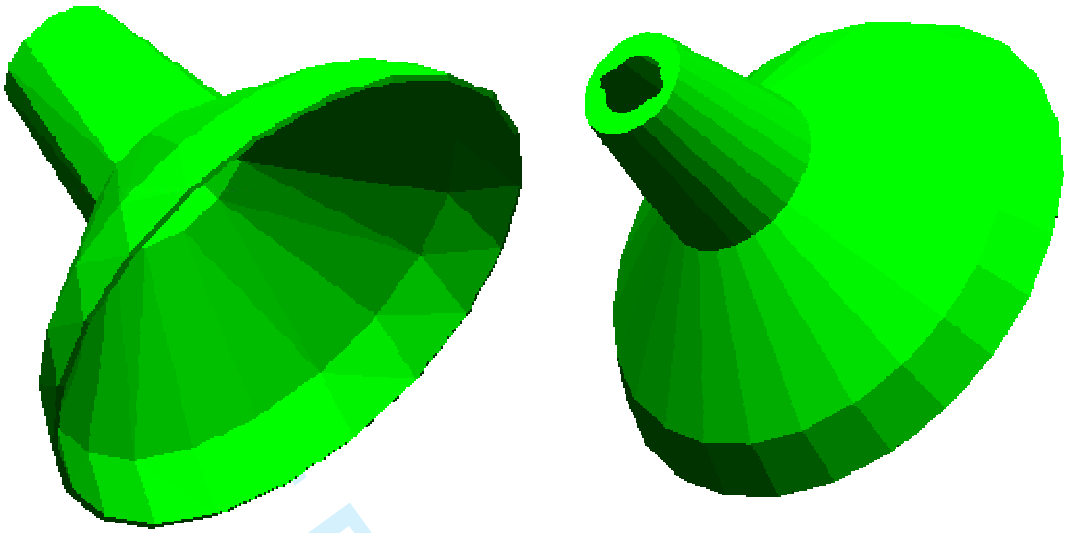


Figure 16. Part 2 – solid model, some offset steps and final mid-surface



solid model mid-surface

Figure 17. Part 3 – solid model and final mid-surface

	# of planes (model)	# of planes (mid-surface)	# of offset steps	total computation time [s]
Part 1	887	587	1	11.757
Part 2	167	83	6	16.836
Part 3	357	259	1	5.683

Table 1. Basic info related to MAT computation.