Rounding Operations on Shell Meshes for Efficient Analysis of Stamping Tools for Automotive Body Panels

Sang Hun Lee, Won Kyung Lee, Kang-Soo Lee
Graduate School of Automotive Engineering, Kookmin University, Seoul, Korea
shlee@kookmin.ac.kr

In order to reduce trial-and-error in the die design for automotive body panels, CAE systems for analysis of stamping tools have been introduced at the initial design stage recently. The current iterative process for simulation and redesign of stamping tools consists of the following three steps: automatic generation of shell meshes from CAD models for stamping tools, simulation of stamping process on CAE systems for formability evaluation, and modification of process parameters and/or local modification of tool geometry on CAD systems. In this process, however, when only a small part of the model is modified, the whole mesh for a modified CAD model should be regenerated for each iteration. In addition, if an automatic mesh generator does not provide a perfect mesh for the simulation, CAE engineers must correct the mesh interactively for each iteration. To solve these problems, in this paper, we propose local modification capabilities for shell meshes, which give a constant or variable radius of rounding directly to sharp edges of a die shoulder in a tool mesh. Using these capabilities, CAE engineers can modify meshes directly instead of CAD models. As a result, the three steps of the current design-analysis iteration have been reduced to two steps: mesh modification and stamping process simulation, and the manpower and facilities for CAD works can be saved.

Keywords: Automotive Panel, Die Design, Stamping Simulation, Shell Mesh, Rounding, Blending.

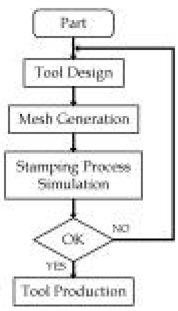
1. INTRODUCTION

In the area of automotive body manufacturing, CAE systems for simulation of stamping process have recently been used at the die design stage in order to reduce the degree of trial-and-error in try-out press. As illustrated in Fig. 1, the current iterative process for simulation and redesign of stamping tools is composed of three steps as described below:

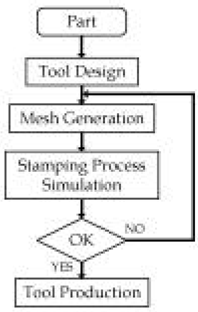
(Step 1) Shell meshes of CAD models for stamping tools including die, punch, and blank-holder are generated by automated mesh generation programs.

(Step 2) Stamping process is simulated for formability evaluation using a CAE system such as PAM-STAMP.

(Step 3) According to the analysis result, if necessary, the process parameters are changed or the CAD models are modified locally using modeling functions of the CAD system such as rounding operations. Then, go to step1 and repeat the cycle until the analysis result is satisfactory.



(a)



(b)

Fig.1 Stamping tool design and analysis process: (a) current process; (b) newly proposed process.

In this process, however, the whole mesh for a modified CAD model should be regenerated for each iteration, although only a small part of the model is modified. In addition, if an automatic mesh generator does not provide a perfect mesh that can be used readily for the simulation, CAE engineers must correct the mesh interactively for each iteration. If the mesh generator is not imbedded in the CAD system, CAE engineers may have to repair the imported CAD data before performing mesh generation, because currently it is almost impossible to transfer CAD data completely from one system to another. Moreover, in order to modify CAD models, CAD systems are occupied, and the labor of CAD operators may be required to support CAE engineers because CAE engineers are usually not skillful in the manipulation of CAD systems. Therefore, f CAE engineers can modify meshes directly instead of CAD models, the design-analysis cycle can be accelerated, and resources for CAD work can be saved.

To achieve this goal, we develop local modification capabilities for shell meshes, which give a constant or variable radius of rounding directly to sharp edges of a die shoulder in a tool mesh. By virtue of these capabilities, as illustrated in Fig. 2, the three steps of the current design-analysis iteration have been reduced to two steps: mesh modification and stamping process simulation. In addition, as CAE engineers can perform the design-analysis cycle without the usage of CAD systems and the cooperation of CAD operators, manpower and facilities can be saved.

RELATED WORK

To date we are unaware of any reported algorithm for rounding operations on a shell mesh model although there has been a lot of research on rounding operations on surface and solid models [1, 2, 3, 4]. Although ESI has implemented the Auto Filleting function, which generates rounding mesh for sharp edges, as a module of PAM-QUICK STAMP that is a quick solver to simulate stamping process in the early design stage, they have not published its detailed algorithms [5].

2. OVERALL PROCEDURE

To facilitate the comprehension, the terminology for mesh rounding is introduced briefly:

- Base mesh: A set of triangular or rectangular elements that are adjacent to the die-shoulder edges. Two base meshes meet at the die-shoulder edges.
- Ball center points: the center points of a virtual sphere that contact on two base meshes.
- Ball-contact points: a pair of points at which a virtual sphere contacts with the base meshes.
- Spine curve: a longitudinal trajectory of a swept surface for rounding. It is obtained from tracing the rolling-ball centers.
- Profile curve: a cross-section of a rounding surface at a point of the spine. It passes through two ball contact points with a center point on the spine curve.

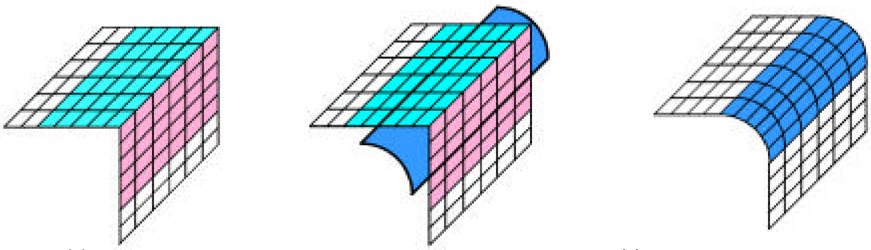
Trim line: a curve for trimming a base mesh. It also constitutes the boundary of a rounding surface or mesh. They are obtained by connecting the ball-contact points on a base mesh.

As illustrated in Fig. 2, the algorithm for the edge rounding operations on a shell mesh consists of the following three main steps:

(Step 1) Detect sharp edges of a die shoulder and their two adjacent face groups, called base meshes, from a given shell mesh.

(Step 2) Generate a rounding surface with a given constant or variable radius, which is in contact with two adjacent face groups. The rolling ball method [6] is adopted for the generation of a rounding surface in this paper.

(Step 3) Cut away the area adjacent to the die-shoulder along the boundary of the rolling-ball surface, then generate a new mesh for the rolling-ball surface, and finally sew it with the original mesh.



(a) (b) (c)

Fig.2 Overall procedure of edge rounding operations: (a) search result of two base meshes; (b) generating a rounding surface; (c) generating a rounding mesh.

The development environment for this work is as follows: An IBM-Compatible Pentium-II PC with 128MB RAM as a hardware platform, Windows NT Workstation 4.0 as an operating system, Visual C++ for compiling and debugging the program, Open Inventor of TGS for graphics user interface, SISL 4.1 of SINTEF as a NURBS geometric library [7], and PAM-STAMP of ESI for conducting stamping process simulation.

An example model has been selected to facilitate readers' comprehension. This model is called as an 's-rail', which was announced as a benchmarking problem in Numisheet 96 Conference [8]. A surface model for the s-rail has been constructed using Pro/Engineer, and its shell mesh was generated using Pro/Mesh, as shown in Fig.3.

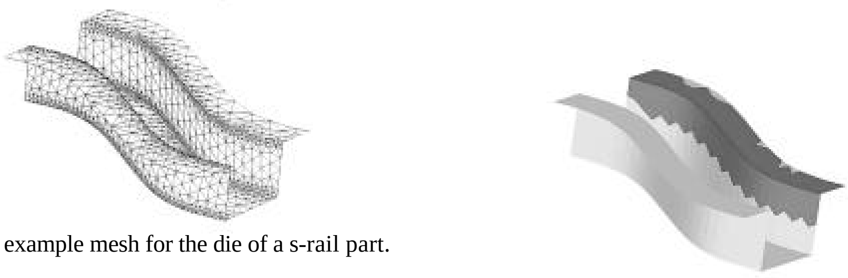


Fig. 3 An example mesh for the die of a s-rail part.

Fig.4 Sharp edges on the die shoulder and a pair of base meshes.

3. RECOGNITION OF DIE-SHOULDER EDGES AND BASE MESHES

In the first step, the system searches for the sharp edges of the die shoulder, on which rounding operations will be applied, and a pair of base meshes adjacent to the die shoulder. If the user selects an edge, the system looks for a series of sharp edges in both the forward and backward directions of the edge. Once the sharp edges has been found successfully, the adjacent faces of the sharp edges within a specified distance (by default, three times of the rounding radius) are searched and grouped by two sets of faces as illustrated in Fig.4. Each set of faces becomes a base mesh that is equivalent to a base surface in surface rounding.

4. GENERATION OF A ROUNDING SURFACE

4.1 OFFSET OF BASE MESHES

Two base meshes are offset by the rounding radius and intersected. A series of the intersection curves become the trajectory of the center of the rolling ball, which is called a spine curve. There exist various offset methods for a mesh according to the treatment of convex edges and vertices. The approach adopted in this paper is to extend the adjacent faces and intersect them to get the edge and vertex geometry, as the shapes of the base meshes are usually wide and smooth and their topology is usually unchanged. Although this method results in an illegal offset mesh, the fault intersection points can be filtered afterward and less computation is required to obtain the intersection between the offset meshes compared to the exact offset method.

4.2 INTERSECTION OF TWO OFFSET BASE MESHES

In this step, intersection curves between two offset base meshes are calculated. Each intersection curve is a straight line segment. These curves are called a spine curve and become a trajectory of the center points of a virtual ball of radius *r* contacting two base meshes. From the spine curve, the ball center points are extracted. They are initially selected with the end points of each intersection line segment. However, if the points are too dense, the system samples a subset of points based on the point interval distance and the curvature analysis in order to prevent a rounding surface from being too complicated. A circle can be generated with three adjacent points, and its radius becomes the approximated radius of the curvature. If the curvature is small enough compared with that specified by the user and the point interval is smaller than the critical distance specified by the user, the center point is removed.

4.3 CALCULATION OF BALL CONTACT POINTS

Once the ball center points are extracted, the two ball-contact points on the base meshes are calculated for each ball center point. These three points are used to generate a profile curve of a circular arc for a rounding surface. As we adopted the approximated method, the ball center points do not always lie on the spine curve. Therefore, it is necessary to devise a refining algorithm to calculate the exact ball center points. If the area of the offset face is less than that of the original face, the ball contact point can be obtained exactly by offsetting the intersection point back to the original face. However, if the area has been increased by offsetting the face, it is necessary to devise an algorithm to obtain the exact ball center points. In this paper, we divide an offset face into three regions: a region offset from the face, regions offset from the bounding edges, regions offset from the bounding vertices. Each region stores the pointer to its original topological entity. If the intersection occurs between two face-offset regions, the intersection point is correct. But if it occurs in any edge-offset or vertex-offset region, it is not correct or may not happen. In this case, the correct intersection point is determined after the contact points are calculated. If the intersection point is located in the face-offset region, the contact point is calculated by projecting the point onto the original face in the inverse direction of the face normal. If the point is located in the edge-offset region, the contact point on an edge is calculated. If the intersection point is located in the vertex-offset region, the contact point on a vertex is found. Next, the system constructs two circles whose center are a pair of contact points, and then calculates the intersection between the two circles. The closer intersection point will be the exact center point. If no intersection happens, this point set is discarded.

4.4 CREATION OF PROFILE CURVES OF A ROUNDING SURFACE

Once each pair of a ball center point and two ball-contact points on the base meshes are obtained for each intersection point, a profile curve of a circular arc is generated as a quadratic NURBS curve using SISL NURBS library.

In the case of variable-radius rounding, some additional work is necessary in this step. Initially, two base meshes are offset by the average radius and intersected to get a spine curve and profile curves. Then, a π plane on which a profile curve lies is constructed and intersected with the base meshes to get the cross-sections of the base meshes. Next, the cross-sections are offset by the actual radius and intersected to get the real ball-center point. Using this new center point, the system calculates a pair of new contact points for a certain radius. This idea originates from Choi's paper [6]. Although this center point is an approximation here, we adopted this approach because it is easy to implement and provides a reasonable result in the engineering sense.

4.5 GENERATION OF A ROUNDING SURFACE

Once the profile curves are constructed, a rounding surface can be built by a lofting function. The traditional approach to build a rounding surface is to construct a pair of ball contact curves on the base surfaces first, and then to sweep the profile curves along these ball contact curves. In the system, a lofting function s1538() provided by the SISL library is called to construct a smooth B-spline surface from the given profile curves. The degree of B-spline surface is cubic in the direction of the spine curve and quadratic in the direction of the profile curve.

5. GENERATION OF A ROUNDING MESH

5.1 TRIMMING THE ORIGINAL MESH

Once the rounding surface is constructed, the round area of the mesh is cut off. To cut the mesh, trim lines need to be defined. But the boundary curves of the rounding surface cannot be used directly as trim lines for cutting the mesh because they are not exactly on the mesh. Therefore an algorithm needs to be devised to find the trim lines on the mesh from the rounding surface boundary. First, the nodes of the base meshes are classified into two groups: moving and non-moving nodes. A moving node means a node that should be moved to the rounding surface, and A non-moving node is a node that remains in its own position. Next, trim lines on the mesh are calculated using the node classification data: First, the system search for all the edges whose end vertices are a moving vertex and a non-moving vertex because these edges must intersect with the boundary of the rounding surface. Then, each of the edges is intersected with the rounding surface boundary. As these two curves do not intersect exactly, we calculate the closest point on the edge of the boundary. Finally, new nodes and edges are generated at the intersection position on the mesh. If any new face is not a triangle, we split it into triangles. If an intersection point is close to a node on the edge, the node is only moved to the intersection point.

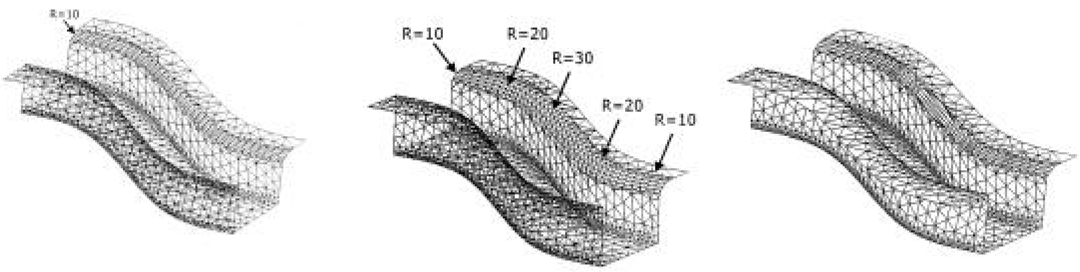
5.2 GENERATING INNER ELEMENTS OF A ROUNDING MESH

In this step, a new mesh is generated on the u, v parametric domain of the rounding surface. Because the nodes on the round mesh boundary should be matched to the nodes on the base mesh boundary, only a mesh for the inner area of the rounding surface is generated first, and then connects the boundary of this mesh with that of the base mesh. The inner mesh is easily generated in equal u, v intervals on the u, v parametric domains of the rounding surface.

5.3 SEWING A ROUNDING MESH WITH THE ORIGINAL MESH

Once the inner mesh is constructed, triangle faces are put into the gap between the rounding mesh and the original mesh. In order to connect the nodes of the two meshes, we use the following method: For each edge of the rounding mesh, the system finds the closest node of the base mesh from the mid-point of the edge, and then creates a new face with two nodes of the edge and the closest node found. If the connection process for the boundary of the rounding mesh is completed, the same process proceeds for the boundary of the base mesh.

Fig.5 shows the results of constant and variable radius rounding operations on the S-rail model. The nodes were generated in equal u, v intervals. If we use the same u, v interval over the rounding surface to generate the round mesh, the mesh result can be too dense in some areas and too sparse in other areas. To overcome these shortages, the mesh generation process starts from the large interval in the direction of the spine curve, and it is examined if the distance from the mid-point of the triangle to the rounding surface is less than the given error allowance. If the error is larger than the allowance, the interval is divided into two. Then the above process is repeated until the error allowance is satisfied. The adaptive constant-radius rounding mesh for the S-rail model is shown in Fig.5(c).



(a) (b) (c)

Fig.5 Rounding results: (a) constant-radius rounding; (b) variable-radius rounding; (c) an adaptive rounding mesh.

6. CONCLUSIONS

In this paper, we propose rounding operations on shell meshes, which give a constant or variable radius of rounding directly to sharp edges of a mesh, and applied them to stamping tool design and analysis process in a computational environment for automotive body production. Using these capabilities, three steps of the current design-analysis iteration have been reduced to two steps: mesh modification and stamping process simulation, and the manpower and facilities for CAD works can be saved as CAE engineers perform analysis and its preprocessing work independently on their own computers. In addition, these operations avoid data exchange problems between different CAD/CAE systems and the regeneration of whole meshes for modified tools. However, the current mesh rounding method has the following limitations, and more research is necessary to overcome the drawbacks. First, the current method only gives rounding to a spine curve. It is necessary to extend this approach to rounding several spine curves joining at a vertex in the future. Second, currently, CAE engineer changes rounding radii manually. If this program is associated with an optimization program for rounding radii, design productivity will increase.

REFERENCES

[1] Vida, J., Martin, R.R. and Varady,T. (1994), ? survey of blending methods that use parametric surfaces , *Computer Aided Design*, vol. 26/5, 341-364.

[2] Choi, B.K. and Ju, S.Y. (1989), "Constant-radius blending in surface modeling", *Computer Aided Design,* vol. 21/4, 213-220.

[3] Varady, T (1989), "Rolling ball blends in solid modeling", In F. Kimura and Rolstadas,A., editors, *Computer Applications in Production and Engineering*, North-Holland, Amsterdam, 295-308.

[4] Chiyokura, H. (1987), "An extended rounding operation for modeling solids with freeform surfaces", *IEEE Computer Graphics and Applications*, vol. 7/12, 27-36.

[5] Khaldi, E. (1998), "Industrial Validation Cases for New Rapid Stamping Simulation Solution", *Proceedings of HAN PAM '98*, 409~443.

[6] Choi, B.K. (1991), *Surface Modeling for CAD/CAM*, Elsevier Science Publish.

[7] SINTEF (1998), SISL Reference Manual, SINTEF.

[8] NUMISHEET 96 - Problem Definition: B2, http://roblix.eng.ohiostate.edu/~sridhar/numisheet/B2-S_Rail.html

<< Back