FEATURE RECOGNITION AND MESH GENERATION FOR POWERTRAIN USING ANSA SCRIPT AND C++ PROGRAMS

[1]Hideki Ishikawa, [2]Koji Otani[*], [2]Hiroaki Ariga
[1]AW Engineering Co., Ltd., Japan, [2]Integral Technology Co., Ltd., Japan

KEYWORDS –
ANSA script, powertrain, feature recognition, mesh generation, pre processing

ABSTRACT –
To ensure certain quality of simulation results, it is essential to create mesh according to specific mesh rules as different FEM mesh can give significant result change.
This paper describes how we generate mesh satisfying the detailed mesh rules for powertrain modeling by recognizing various specific features from the shape of input 3D CAD models and creating appropriate mesh automatically using ANSA script and C++ code.
We used ANSA script to create database, calculate from geometry shapes, call ANSA commands and execute C++ programs. We used C++ programs to solve complicated calculations that do not require geometry data.
Using the feature recognition and mesh generation techniques, we achieved to automate mesh generation procedures for powertrain following the specific mesh rules.

TECHNICAL PAPER -

1. INTRODUCTION

It is essential to create good quality mesh according to mesh rules to assure the quality of simulation result since it may vary depending on the mesh. Mesh rules may cover not only simple features such as fillets, holes, flanges, chamfers, and embosses but also other complex features such as tapered holes, steps, ribs, grooves, and gear teeth. Various approaches to automate feature recognition and mesh generation have been studied. (1-10) ANSA can recognize and control mesh according to mesh rules for various features such as fillets, flanges, chamfers, and so on. However, it does not recognize features and control mesh meeting all mesh rules that a variety of analysis engineer have for each analysis phenomena. Creating ideal FEM mesh according to detailed mesh rules is time-consuming when you cannot get the result that you want instantly and need to modify it manually. (11) To solve this problem, we developed algorithms to recognize a variety of features from the input 3D CAD model shapes and generate mesh automatically according to the detailed mesh rules using ANSA script and C++ programs along with ANSA.

2. SYSTEM OVERVIEW

The system that we developed with the algorithms works on its own Graphical User Interface (GUI) so the users can just drag and drop their input files and click the start button to create mesh. The system configuration diagram is shown in Figure 1. The GUI software that is developed with C++ executes the ANSA script program. This ANSA script program manage all the feature recognition and mesh generation calling ANSA commands and controlling other C++ programs.

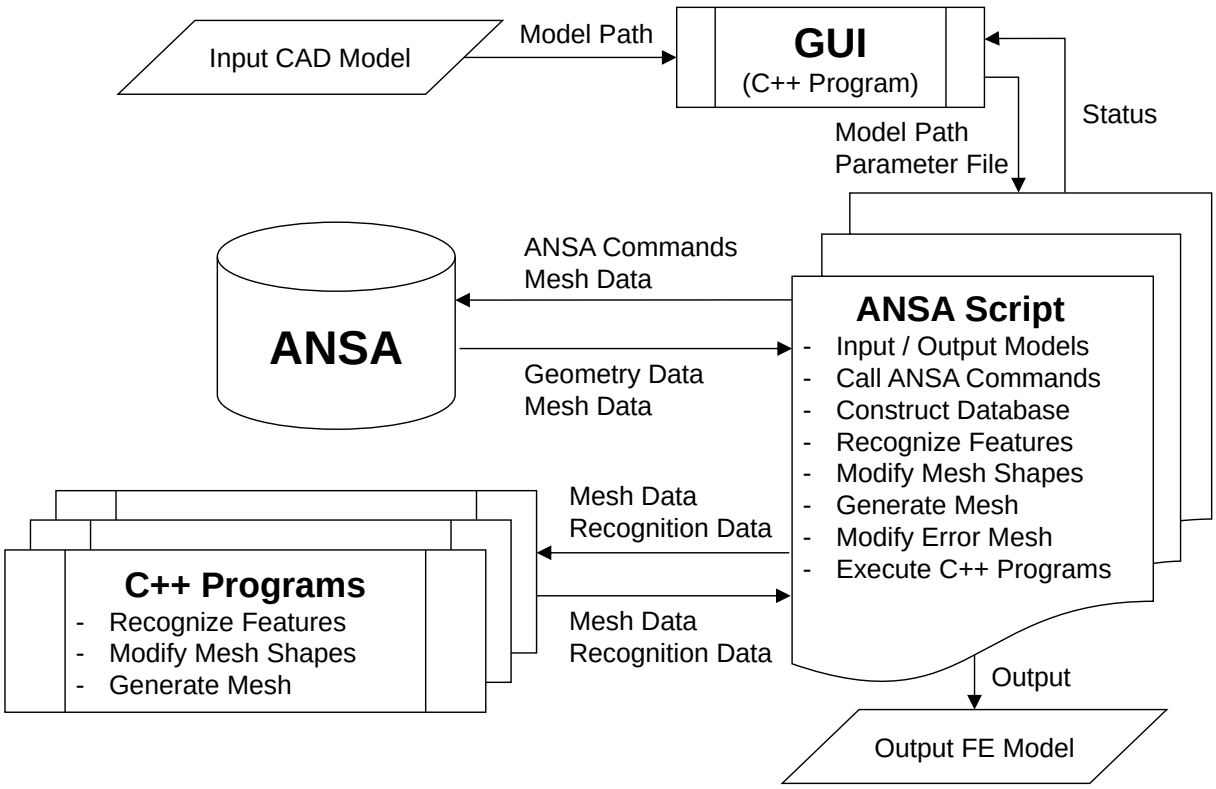

Figure 1 - The system configuration diagram.

3. FEATURE RECOGNITION

The algorithm that we developed to recognize specific features of powertrain parts can be divided into 3 main steps: acquiring information of each face of the CAD models, categorizing each face based on its acquired information and characteristics, and recognizing complex features that consist of a group of faces.

The first step is to acquire information of each face of CAD models. We first create free mesh for each face with ANSA commands and construct mesh database for calculation of characteristics. The characteristics we calculate includes curvature of face, curvature of perimeter, length of perimeter, hot point on perimeter, length ratio of equivalent perimeter pair, circle shape of perimeter, cylinder shape, sphericity, inner angle of hot point, area of face, angle difference between normal vectors within face, surface continuity, contact angle on perimeter, and cross sectional shape. To calculate these characteristics, we mainly refer the mesh database along with additional data that we acquire with ANSA commands like “ProjectAndMarkPoints”.

The second step is to categorize each face based on its acquired information and characteristics. After acquiring data and calculate characteristics for each face, we categorize each face into 5 different face types: plane, fillet, cylinder, sphere, and curved face. We check specific characteristics to categorize depending on the type. For example, to consider a face as a fillet face, we examine curvature of face, curvature of perimeter, length of perimeter, hot point on perimeter, length ratio of equivalent perimeter pair, circle shape of perimeter, inner angle of hot point, area of face, surface continuity, contact angle on perimeter, and cross sectional shape. Fillet, cylinder, sphere, and curved face are all considered as curved face at first. Then we distinguish between them by their different characteristics. Cylinder and sphere can be fillet. We consider them as fillets if they have a continuous curved face next to them in specific way. The characteristics used for categorizing faces are shown in table 1.

The third step is to recognize complex features that consist of a group of faces. We use the face type data and other information to recognize features such as hole, tapered hole,

stepped hole, step on plane, fillet flow, chamfer, corner fillet, thin face between fillets, rib, groove, and gear teeth. For each feature, we defined the combination of face types, positional relationship between faces, feature size, and other factors to recognize.

Table1 – The characteristics used for categorizing faces.

	Face Types				
Characteristics	Plane	Fillet	Cylinder	Sphere	Curved face
Curvature of face	✔	✔	✔	✔	✔
Curvature of perimeter	✔	✔	✔	✔	✔
Length of perimeter		✔	✔		
Hot point on perimeter (position & number)		✔			
Inner angle of hot point		✔			
Length ratio of equivalent perimeter pair		✔	✔		
Circle shape of perimeter		✔	✔	✔	
Cylinder shape			✔		
Sphericity				✔	
Area of face	✔	✔	✔	✔	✔
Angle difference between normal vectors within face	✔				✔
Surface continuity		✔	✔	✔	
Contact angle on perimeter		✔	✔	✔	
Cross sectional shape	✔	✔	✔	✔	✔

Table2 – The face type used for recognizing features.

	Features										
Face Types	Hole	Tapered hole	Stepped hole	Step on plane	General face	Fillet flow	Chamfer / corner fillet	Thin face between fillets	Rib	Groove	Gear teeth
Plane			✔	✔	✔		✔	✔	✔	✔	✔
Fillet						✔	✔	✔	✔	✔	
Cylinder	✔	✔	✔								✔
Sphere											
Curved face				✔	✔			✔	✔	✔	✔

4. MESH GENERATION

We mainly focused on modifying input model shapes and moving boundary lines instead of controlling mesh since ANSA has robust meshing algorithm. We used ANSA commands such as “MESH_SHELL_MESH_FREE”, “MESH_SHELL_MESH_RE_MESH “, “MESH_SHELL_MESH_RECONSTRUCT “, and “RunMeshingScenario” to generate and modify mesh after we modify the input model shapes and moving boundary lines.
However, we used our own algorithm to create mesh to follow the mesh rules for features such as cylinder holes, tapered holes, stepped holes, or simplified gear teeth.
To modify input model shapes, we reshaped mesh that is generated for each face of the input models by ANSA Commands at first. We applied this mesh shape modification method for features such as stepped holes, corner fillets, chamfers, corner clearances, ribs, grooves, and gear teeth. To move important feature boundary lines, we calculated and offset imaginary lines and modify mesh to create new mesh boundary lines. We applied this mesh boundary line move method for features such as steps, fillets, and thin faces between fillets.

5. ANSA SCRIPT AND C++ PROGRAMS

We used C++ programs along with ANSA script for functions that does not require geometry data for complicated calculations. We used C++ programs for recognizing steps on planes, gear teeth, and so on. Our C++ programs can also move boundary lines for recognized steps and simplifying gear teeth. The C++ programs require mesh data and recognition data when ANSA script runs them. The recognition data includes face types and characteristics of each face. The C++ programs imports all the data given, construct database, and process the third step of the feature recognition algorithm, which recognizes complex features that consist of a group of faces. After recognizing their targeted features, the programs move boundary lines or simplify shapes by modifying mesh depending on the feature they detect. Then the programs output mesh data and recognition data as result.

6. CONCLUSIONS

We developed algorithms to recognize specific features and create mesh according to the mesh rules using ANSA script and C++ programs. Using the feature recognition and mesh generation algorithms described in this paper, we have achieved to automate the mesh generation process according to specific mesh rules and reduced the manual mesh modifying work by more than 90%.

REFERENCES

(1) Hao Lan Zhang, Van der Velden C., Xinghuo Yu, Bil C., Jones T., and Fieldhouse I., Developing a rule engine for Automated Feature Recognition from CAD models, Industrial Electronics, IECON '09. 35th Annual Conference of IEEE, 2009

(2) S Kumar, K. Shanker, and G.K. Lal, A feature Recognition Methodology for Extrudable Product Shapes, International Journal of Production Research, Volume 37, Issue 11, 1999

(3) Jami J. Shah, David Anderson, Yong Se Kim, and Sanjay Joshi, A Discourse on Geometric Feature Recognition from CAD Models, Journal of Computing and Information Science in Engineering, Vol. 1, 2001

(4) Bojan Babic, Nenad Nesic, and Zoran Miljkovic, A Review of Automated Feature Recognition with Rule-Based Pattern Recognition, Computers in Industry, Volume 59, Isuue 4, 2008

(5) Shuming Gao, Wei Zhao, Hongwei Lin, Fanqin Yang, Xiang Chen, Feature Suppression Based CAD Mesh Model Simplification, Computer=Aided Design, Volume 42, Issue 12, 2010

(6) Y. Lu, R. Gadh, T.J. Tauyges, Feature Based Hex Meshing Methodology: Feature Recognition and Volume Decomposition, Computer=Aided Design, Volume 33, Issue 3, 2001

(7) Dimitri J. Mavriplis, An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness, Journal of Computational Physics, Volume 117, Issue 1, 1995

(8) Joachim Schöberl, NETGEN An Advancing Front 2D/3D-Mesh Generator Based on Abstract Rules, Computing and Visualization in Science, Volume 1, Issue 1, pp 41–52, 1997

(9) Ted D. Blacker, Michael B. Stephenson, Paving. A New Approach to Automated Quadrilateral Mesh Generation, International Journal ofor Numerical Methods in Engineering, Vol. 32, 811-847, 1991

(10) Kyu-Yeul Lee, In-Il Kim, Doo-Yeoun Cho, Tae-wan Kim, An algorithm for Automatic 2D Quadrilateral Mesh Generation with Line Constraints, Computer-Aided Design, Volume 35, Issue 12, 2003

(11) Current Trends and Issues In Automatic Mesh Generation, Kenji Shimada, Carnegie Mellon University, 2006