

# Automatic Mesh Generation of Hybrid Mesh on Valves in Multiple Positions in Feedline Systems

Douglas H. Ross<sup>1</sup>, Yasushi Ito<sup>2</sup>, Fredric W. Dorothy<sup>3</sup>, Alan M. Shih<sup>4</sup>  
*University of Alabama at Birmingham, Birmingham, Alabama, 35294*

John Peugeot<sup>5</sup>  
*NASA Marshall Space Flight Center, MSFC, Alabama 35812*

## I. Introduction

Fluid flow simulations through a valve often require evaluation of the valve in multiple opening positions. A mesh has to be generated for the valve for each position and compounding. The problem is the fact that the valve is typically part of a larger feedline system. In this paper, we propose to develop a system to create meshes for feedline systems with parametrically controlled valve openings. Herein we outline two approaches to generate the meshes for a valve in a feedline system at multiple positions. There are two issues that must be addressed. The first is the creation of the mesh on the valve for multiple positions. The second is the generation of the mesh for the total feedline system including the valve. For generation of the mesh on the valve, we will describe the use of topology matching and mesh generation parameter transfer<sup>1</sup>. For generation of the total feedline system, we will describe two solutions that we have implemented. In both cases the valve is treated as a component in the feedline system. In the first method the geometry of the valve in the feedline system is replaced with a valve at a different opening position. Geometry is created to connect the valve to the feedline system. Then topology for the valve is created and the portion of the topology for the valve is topology matched to the “standard” valve in a different position. The mesh generation parameters are transferred and then the volume mesh for the whole feedline system is generated. The second method enables the user to generate the volume mesh on the valve in multiple open positions external to the feedline system, to insert it into the volume mesh of the feedline system, and to reduce the amount of computer time required for mesh generation because only two small volume meshes connecting the valve to the feedline mesh need to be updated.

## II. Valve-Only

Typically valve geometries are complex enough to require construction in a computer-aided design program (CAD) or in a mesh generation system that has significant modeling capability. For the purpose of this paper we will consider geometry created externally in a CAD program and requiring multiple meshes to be created on the same valve with the change being the opening position of the valve. Therefore a mesh system must have an ability to create meshes on imported geometry. Most meshing systems require that all the meshing elements and specifications be reattached to each importation of the geometry. A method available in some meshing systems that mitigates this work is a scripting system. However, a script has to be created for each new valve geometry.

The method presented in this paper to generate a mesh on a valve with a modified opening position is mesh generation based on topology matching<sup>1</sup>. Where topology is defined as a boundary representation (b-rep) based on trimmed Non-Uniform Rational B-Splines (NURBS) surfaces including connectivity information<sup>2,3</sup>. The matching portion of the method is to create a one-to-one correspondence between a “standard” topology, which already has a mesh, and a “new” topology, on which a new mesh will be created based on mesh generation parameters used for the mesh on the “standard” topology. The correspondence is created by first matching faces, then unique edges, then vertices and then non-unique edges between the two topologies. The primary match algorithm for faces is a breadth first search with backtracking. A potential match is a face with the same trim structure and the same connection to

---

<sup>1</sup> Programmer II, Dept. of Mechanical Engineering, BEC 257, 1150 10<sup>th</sup> Avenue South, Birmingham, AL 35244.

<sup>2</sup> Research Assistant Professor, Dept. of Mechanical Engineering, BEC 257, 1150 10<sup>th</sup> Avenue South, Birmingham, AL 35244, Senior Member AIAA.

<sup>3</sup> Programmer, Dept. of Mechanical Engineering, BEC 257, 1150 10<sup>th</sup> Avenue South, Birmingham, AL 35244.

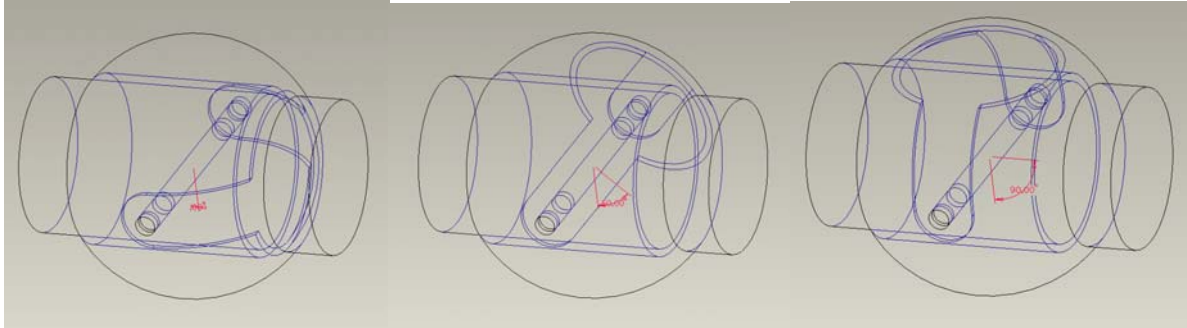
<sup>4</sup> Research Professor, Dept. of Mechanical Engineering, BEC 257, 1150 10<sup>th</sup> Avenue South, Birmingham, AL 35244, Associate Fellow AIAA.

<sup>5</sup> Research Engineer, NASA MSFC ER42, Member, AIAA.

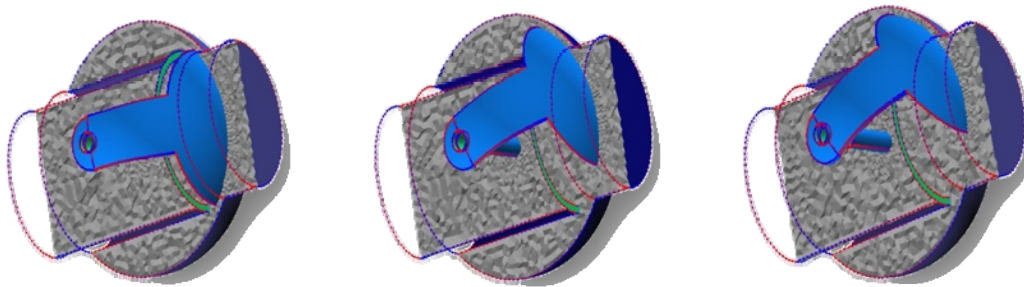
faces which already have a correspondence built. All possible correspondences are built and a best correspondence is chosen. The heuristic to determine the best correspondence is based on the minimum summation of angles between matching faces. To have a viable angle measure a Principal Component Analysis (PCA) is performed on both topologies and a transformation applied to bring the PCA basis to the standard basis. After the topologies have been matched it is a mechanical operation to transfer the grid creation specification information from the “standard” topology to the “new” topology based on the parametric connection values. A mesh is then created on the “new” topology.

If the valve is topology matched without placing it in the feedline system then the valve geometry does not require its position to be transformed to align with the matching topology and principal component analysis is not required. This provides a better match since most of the geometry is at the exact same position. Only a portion of the valve moves so most of the mesh generation parameters do not have to be changed.

While a benefit of the topology matching method for mesh generation is that the geometry can be created in a CAD system that is specifically designed for complex geometry creation. For some valve geometries generated in a CAD system the topology only remains the same over a portion of the valve movement. Of course the valve geometric topology changes when the valve closes but for flow systems only open valve positions are interesting. Three examples of valve geometries are examined: a ball valve, a shield valve and a poppet valve. Figures 1 and 2 show an example of a shield valve with different valve positions and a volume mesh created for it. The ball and shield valves have two ranges of angles over which the topology remains the same. Therefore they each require two “standard” topology models for the matching process. The poppet valve topology remains the same over all opening positions. Shown in Figure 3 is an example of automated mesh generation for a poppet valve of different openings.



**Figure 1. Shield valve geometry created in ProE with parameter value for angle of valve opening**



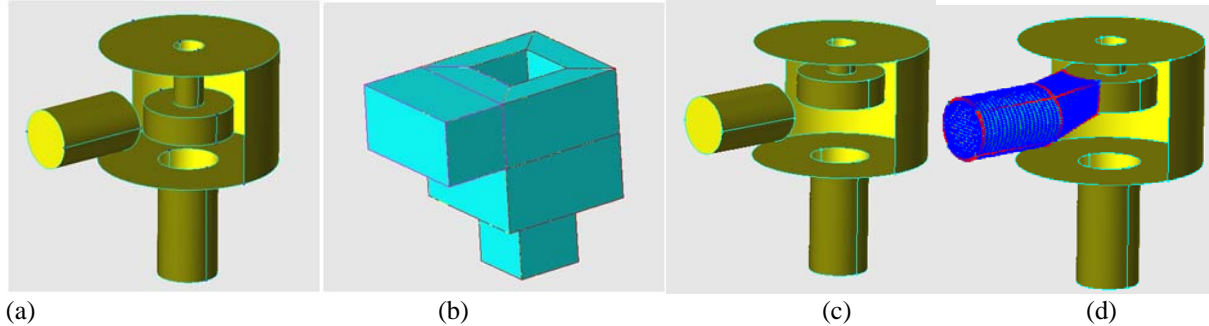
**Figure 2. Shield valve mesh generated by topology matching and mesh generation parameter transfer**

Importing geometry into a watertight topology structure has another benefit. Imported geometry from CAD systems often has gaps between trim surfaces. The topology structure works to close these gaps by providing a single representation at the boundary between surfaces.

The geometry and/or mesh for feedline systems with components can be created in at least as many ways as the mesh for a single valve. The advantages and disadvantages are similar to those for a single valve. The mesh can be created as a custom template by using a software library which is laborious. A general purpose meshing system can be used which may or may not support a parametric valve positioning. The feedline itself is typically straight, curved and branching sections. The components of a feedline system can be simple or complex. Since valves are typically complex the method of adding a valve as a component to a feedline system typically require that the valve

geometry be imported, while the feedline can be generated externally or internal to the meshing system. Two alternatives for generation of the feedline system and two alternatives for replacement of the valve in the feedline system are presented.

The methods for generation of the feedline system are using a CAD package and using the MiniCAD geometry and mesh generation system. But they can be imported from external CAD systems as well.



**Figure 3. Topology associated with the poppet valve: (a) Topology entities; (b) Grid pattern structure was built up out of butterfly and H-grid blocks. The new valve geometry with a more open valve position is imported into MiniCAD and converted to a topology structure. The new topology is matched with the old topology and the grid pattern mesh is transferred from the old topology to (c) the new topology to create (d) a mesh for the more open valve position**

### III. Valves in Feedline

User developed extendibility is essential in modern CAD packages. MiniCAD<sup>2</sup> has been developed from the start with this in mind. All of MiniCAD's functionality is wrapped up in a single library that can be imported into any python script. The user can therefore write short python scripts that can take full advantage of MiniCAD's geometry and grid creation abilities. These scripts can be run from within MiniCAD's interface, and can be designed to have user interfaces that query the user for input. A script was created to generate feedline systems that are run from the script tab available to the user.

That script provides functionality for piecing together feedline systems from a common set of components. The user can build up a feedline system by adding various parts such as straight sections, T-joints, curved sections, and more advanced components, such as valves and tanks. The user can specify the shape and size of all the components, and how they connect together, allowing the user to build up a feedline system similarly to how the user would build the system in real life.

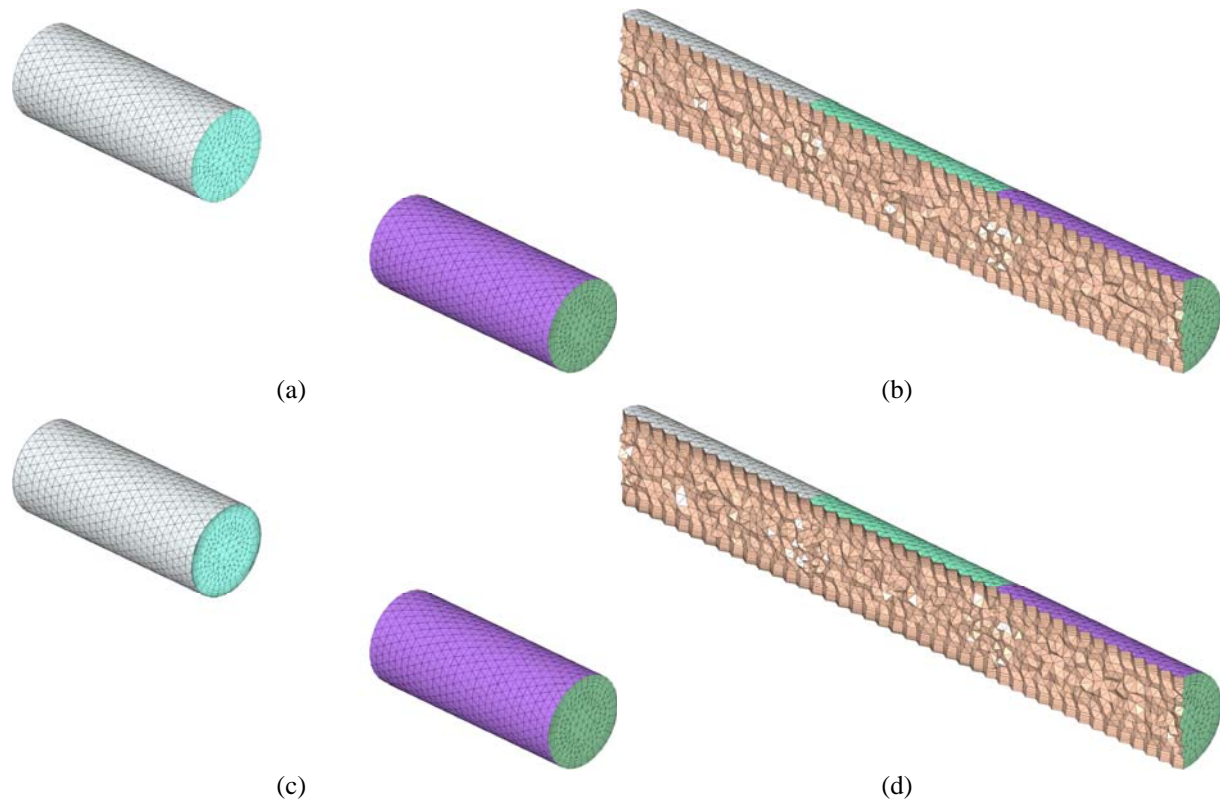
This has a positive impact on the development of feedline systems for two reasons. The first and obvious impact is that the user is able to build feedline systems more quickly. There is less time spent reinventing each part as the feedline system is built. The second impact is that this allows for high-level parameterization of the feedline system. This is because the placement and shape properties of the feedline components are stored as scriptable parameters inside of MiniCAD. The user can build a feedline system, and then automatically produce multiple grids by varying the properties of individual components of the feedline system.

Valve or other component insertion or replacement in the MiniCAD feedline system requires that the valve geometry be prepared by specifying the inlet and outlet curves. The valve can then be inserted with consideration for the number and placement of the trim curves at the pipe element and the valve. Typically the shape and number and start/end points of the trim curves on the valve will not be the same or align with the curves on the feedline element that it connects to. Two NURBS surfaces are created that blend the curves together with tangency considerations and are split in the middle to match the topological requirements of the trim curves on the pipe element and the valve. Using this method the geometry of the valve and the pipe element do not have to be modified. After the connecting geometry and mesh have been created topology matching can be used on the valve to transfer grid parameters and a mesh generated on the total feedline system.

Using a CAD system and in particular Pro/E for generation of the geometry for feedline systems is relatively straightforward but some issues require consideration. Since the valve geometry is considered to be complex the valve would typically be created as a separate part. Therefore the feedline system would be created as an assembly rather than as a single part. If the valve was added into a single part the volume of the feedline system would well defined. However when a part is added to an assembly each part is its own volume and does not merge into the assembly to create a single volume. Some options to deal with this would be that the meshing system be able to attach meshing information to a geometry structure that is not a single volume and does not have a good topological

structure. Alternatively the meshing system could perform all the required Boolean operations to create a two-manifold geometry structure. The method that we use is to stop the feedline short of the valves to be replaced and create connecting geometry as is done in a similar manner to the MiniCAD self contained feedline system. A feedline geometry created in an external CAD system the feedline is generated with an appropriate amount of gap between the end of the pipes and the valve inlet and outlet. The pipes are imported and mesh is generated on them. The valve has mesh generated for multiple opening using topology matching. The mesh for the valve is placed into position using rigid body transformations to fill its spot in the feedline system. Connector geometry is created based on curves generated from the grid edges surrounding the faces of the pipe and component faces to be joined. A NURBS surface with good tangential properties is created to connect the two curves and a surface mesh is generated on the connecting surface.

A hybrid volume mesh can be generated using the Mixed-Element Grid Generator in 3 Dimensions (MEGG3D)<sup>4</sup> based on the connecting surface mesh and the extracted surface meshes from the pipe outlet and valve inlet. The valve outlet and pipe inlet have both triangular and quadrilateral elements that were generated when the hybrid mesh was generated on the feedline system and valve component. Figure 4 shows a simple example for pipe geometries. A good-quality hybrid mesh can be created from a surface mesh with pre-specified interfaces. The white cylinder at the upper left in each sub-figure imitates the valve component and the purple cylinder represents the feedline element. The green cylinder shown in Figures 4b and d represents the connector created between the feedline and valve component. The replacement of the valve component with different volume mesh parameters can still be connected to the feedline with a high-quality mesh.



**Figure 4. New hybrid meshes created between existing hybrid meshes: (a) Two hybrid meshes created from the same parameters for the prismatic layers; (b) Hybrid mesh between the hybrid meshes shown in a; (c) Two hybrid meshes created from different parameters for the prismatic layers (tighter spacing for the upper left pipe); (d) Hybrid mesh between the hybrid meshes shown in c**

#### IV. Conclusion

A method for automatically creating mesh on feedline systems including valves with multiple opening positions has been presented. The second method for generation of the volume mesh by generation of mesh on the valve and then insertion in the previously meshed feedline system is the preferred method of operation from both a time and

simplicity standpoint. Future work will include the application of multi-block structured mesh. This paper will include more valve examples to demonstrate the innovation in the topology matching approach to achieve automated mesh generation for topologically-equivalent geometries. More technical details will also be included in the final paper and presentation to provide detailed information about the approach and its technical foundation.

### Acknowledgments

This work was supported by the Space Vehicle Technology Institute under grant NCC3-989 jointly funded by NASA and DOD within the NASA Constellation University Institutes Project, with Claudia Meyer as the project manager.

### References

- <sup>1</sup> Ross, D. H., Dorothy, F. W., Ito, Y., Shih, A. M. and Soni, B. K., "Mesh Generation Transfer Based on Topology Matching," AIAA Paper 2008-0926, 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2008.
- <sup>2</sup> Gopalsamy, S., Ross, D.H. and Shih, A.M., "API for Grid Generation over Topological Models," Proceedings of the 13<sup>th</sup> International Meshing Roundtable, Williamsburg, VA, 2004, pp. 221-230.
- <sup>3</sup> Shih, A.M., Yu, T-Y, Gopalsamy, S., Ito, Y., and Soni, B.K., "Geometry and Mesh Generation for High Fidelity Computational Simulations Using Non-Uniform Rational B-Splines," *Applied Numerical Mathematics*, Vol. 55, Issue 3, 2005, pp. 368-381.
- <sup>4</sup> Ito, Y., Murayama, M., Yamamoto, K., Shih, A. M. and Soni, B. K., "Efficient CFD Evaluation of Small Device Locations with Automatic Local Remeshing," AIAA Paper 2008-7180, 26<sup>th</sup> AIAA Applied Aerodynamics Conference, Honolulu, HI, 2008.