

# Curving for Viscous Meshes

Steve L. Karman <sup>1</sup>

<sup>1</sup> Pointwise Inc., Fort Worth, TX 76104, USA  
skarman@pointwise.com

**Abstract.** Finite-element flow solvers can utilize high-order meshes to achieve improved accuracy over traditional linear meshes. High order meshes are generally created by elevating linear meshes. For high Reynold's number viscous flows, the linear mesh is tightly clustered to no-slip surfaces. For curved boundaries the high-order mesh must also curve to match the geometry curvature. An optimization-based node perturbation scheme is described that used a two-component cost function to optimize the high order mesh. The first component uses element Weighted Condition Number (WCN) to enforce element shape. The second component uses a normalized Jacobian to enforce element size and validity. The method is applied to several complex linear meshes with highly curved boundaries and tightly clustered normal spacing.

**Keywords:** Mesh Curving, High Order Meshes, Viscous Meshes.

## 1 Introduction

Significant advances have been made in finite-element (FE) techniques for Computational Fluid Dynamics (CFD) in recent years [1, 2, 3]. Finite-element techniques offer increased accuracy over traditional CFD methods, such as finite-volume, with fewer degrees of freedom and increased efficiency. Two popular approaches are Streamline Upwind/Petrov-Galerkin (SU/PG) and Discontinuous Galerkin (DG). Both approaches achieve stable solution by using upwind methods. SU/PG modifies the Galerkin weighting functions to achieve upwinding. DG applies flux jump conditions at element boundaries to approximate a Riemann solution. Linear meshes (polynomial degree 1) result in 2nd order accurate solutions. Higher order accuracy is achieved by introducing additional vertices (new degrees of freedom) to the element edges, faces and interiors that contribute to the integration of the governing equations. Arbitrary polynomial order is possible, however typical CFD applications tend to use orders of polynomial degrees 2 (P2), 3 (P3) or 4 (P4).

Elevating a linear mesh to a higher degree is relatively straightforward. The new vertices are inserted into the element at predefined locations, such as equally spaced for Lagrangian-based interpolation schemes. SU/PG methods are continuous, so the nodes shared by adjacent elements are not distinct. DG methods duplicate edge and face nodes, so each element has its own copy. This duplication is typically performed by the flow solver, so the task of mesh generation is to create the continuous version of the high-order mesh.

CFD solutions for viscous computations require clustering of the nodes towards no-slip boundaries in the mesh to properly resolve the viscous boundary layer. This creates high aspect ratio elements near the boundary. If the boundary is curved this complicates the mesh elevation process and requires the elements themselves to be curved. Mesh curving has been attempted in a number of ways to alleviate mesh tangling issues associated with curving meshes, with varying degrees of success. Interpolation methods, such as mean value coordinates [4] and radial basis functions [5] have been used. The more successful approaches tend to use solid mechanics analogies where the mesh is treated as an elastic solid that deforms due to forces acting on the boundaries [6, 7]. Manipulation of the stiffness model or thermo-elastic properties helps maintain valid elements in the mesh interior. More severe cases of high curvature and tight viscous clustering still present challenges. Other efforts focus on the solution to the Winslow equations to perform the interior mesh curving [8]. This approach is a natural application of Winslow smoothing techniques as a copy of the unperturbed, elevated mesh serves as the computational mesh. The solution to the Winslow equations then forces the interior of the physical mesh to take on the same character of the computational mesh, i.e. smooth variation of the spacing and element volume across the physical domain. However, more severely curved cases still present challenges requiring strategies such as freezing the original linear nodes.

A mesh optimization method was presented at IMR25 [9]. The method attempts to optimize a cost related to a distortion measure from an isotropic state, which would be insufficient for viscous mesh curving. An alternate approach for viscous mesh curving was first published in 2016 [10]. This approach relies on a node perturbation/smoothing process first utilized for general isotropic unstructured mesh smoothing [11]. The basic algorithm was generalized for prescribing desired element shapes and extended to the four basic element types [12, 13]. Further enhancements to the original curving technique have been made to integrate with a geometry kernel and enforce positive Jacobians for FE simulations. Surface nodes are projected onto CAD surfaces using a lightweight geometry kernel, Project Geode. The cost function controlling the node perturbations has been expanded to include a normalized Jacobian component that ensures valid positive Jacobians are produced. Details of the mesh curving process are presented. Several examples with a high degree of curvature and tightly clustered viscous meshes are included.

## 2 Mesh Curving Process

The mesh curving process assumes a valid linear mesh has been created, as opposed to actually curving the mesh during mesh creation. In addition, any curved boundaries require the underlying shape be defined in a Computer Aided Design (CAD) model. As the linear mesh is elevated and as the mesh nodes are smoothed, adherence to the geometry is enforced.

## 2.1 Project Geode

Project Geode is a fourth generation, solid modeling and geometry kernel written in C++. It is an integral part of the commercial mesh generation software, Pointwise [14]. A lightweight version is available for queries such as point projections. Use of Project Geode by the mesh curving process is made robust and efficient by segregating the geometric entities into CAD groups and using a mesh linkage that defines the required geometric entities for each mesh node. Project Geode with the mesh linkage are used by the mesh curving program to perform closest point projections for curved surface edge and interior vertices.

### CAD Groups

The geometric entities are sorted into collections of surfaces (NURBS) and curves (B-SPLINES) required by the various mesh surface faces and edge curves. Each group gets stored in a binary split (BSP) tree for fast searches. As a surface mesh interior or edge node is moved it will be projected to the appropriate CAD group in the associated BSP. This provides for efficient projections and robustness. Robustness in the sense that mesh nodes on the top side of a thin geometry, such as a wing trailing edge, are projected to a different CAD group than the mesh nodes on the bottom side of the thin geometry.

### Mesh Linkage

The associativity between the mesh entities (points, curves and surface meshes) and the underlying geometry is typically discarded when the final linear mesh is exported to a file. An open source schema is under development that provides information to re-associate mesh entities with the appropriate geometric entity. The schema recognizes and leverages the hierarchical nature of mesh topology, e.g. edges (and their points) are constrained to CAD curves while faces are constrained to CAD surfaces. Furthermore, by allowing a CAD group to contain many CAD entities, the schema provides mesh linkage for cases where mesh topology does not match CAD topology. This schema will be available for any mesh generator to use to write the mesh linkage file. The mesh linkage can then be used by the mesh adaptive or mesh elevation code with the geometry file (IGES, STEP, etc.) to perform the necessary geometric queries of the CAD geometry.

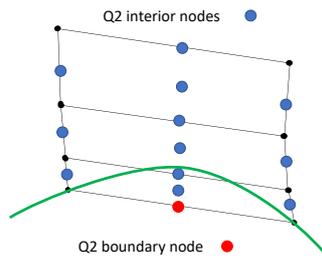
## 2.2 Mesh Elevation and Initial Perturbation Field

High-order meshes are typically obtained by introducing new nodes on edges, faces and in the interior of elements. The number of new nodes added is dependent on the desired order of accuracy. A hybrid linear mesh is usually comprised of tetrahedra with 4 nodes, pyramids with 5 nodes, prisms with 6 nodes and hexahedra with 8 nodes. All these nodes are located at corners of the elements. These are designated P1 elements as they are used with basis functions defined by first order polynomials, i.e. planar solution variation across the element. Adding mid-edge nodes to each linear edge in the P1

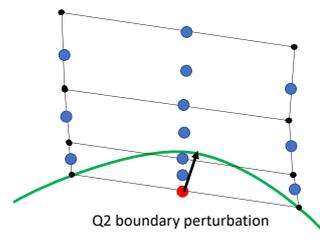
mesh, mid-face nodes to quadrilateral faces in the P1 mesh and a centroidal node for hexahedral P1 elements will create P2 elements. P3 and P4 elements are created in a similar fashion, increasing the number of nodes on each edge, face and element interior. The maximum polynomial order permitted in this endeavor is P4. The CGNS numbering convention is used to identify the ordering of the new nodes in each element [15].

Each new node associated with a boundary edge is projected to its CAD curve group using Project Geode. Each new node associated with a boundary surface is projected to its CAD surface group using Project Geode. For viscous meshes with tight clustering this can invert elements close to convex curved surfaces requiring some form of smoothing to untangle the mesh. To minimize the tangling and provide a better initialization of the interior, a perturbation field is created based on the boundary node perturbations.

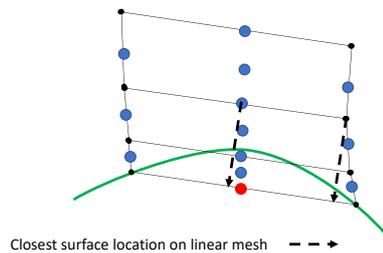
All new high-order nodes are initially placed at locations linearly extrapolated from the boundary node perturbations. The process will be described using a two-dimensional section of a Q2 mesh, shown in **Fig. 1**. The mesh is a collection of quadrilateral elements, but triangles are permitted. In three dimensions all four basic element types are permitted. The high order nodes are depicted in blue. The high order boundary node is red. The boundary geometry is the green curve. The projection of the boundary high order node is shown in **Fig. 2**.



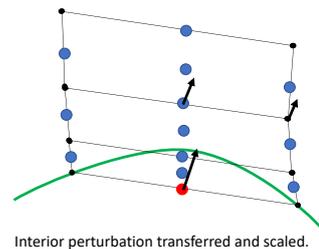
**Fig. 1.** Section of mesh with Q2 interior and boundary nodes.



**Fig. 2.** Perturbation vector for boundary Q2 node.



**Fig. 3.** Closest locations on linear mesh.



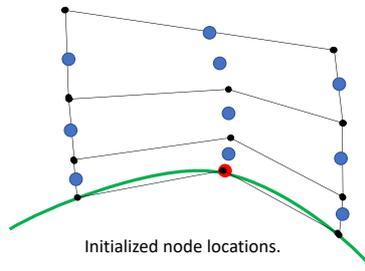
**Fig. 4.** Interior node perturbations.

Each interior node locates the nearest boundary segment or boundary node of the unperturbed mesh, shown in **Fig. 3**. A surface perturbation vector is computed at these

locations by interpolating from the endpoint perturbations. The original linear nodes will have a zero-perturbation vector. This perturbation vector is transferred to the interior node using a linearly decaying function, shown in Equation (1). The distance from the interior node to the boundary location is  $ds$ . The factor decays rapidly away from the boundary.

$$\vec{P}_i = \vec{P}_b * MIN(1.0, \frac{|\vec{P}_b|}{ds}) \quad (1)$$

This perturbation field greatly improves the initial curved mesh in the convex and concave regions. Only an extremely small percentage of elements for more complex configurations are invalid prior to smoothing. The resulting initial mesh for the two-dimensional example case is shown in **Fig. 5**.



**Fig. 5.** Final initialized node locations.

The three-dimensional perturbation field is created in a similar manner using a multi-step process. Each surface high order node has been projected to the appropriate edge curve or geometry surface using Project Geode. This may not be adequate for surfaces that are planar, but the boundary edges are highly curved. So, the first step considers the boundary mesh patches and projects each surface mesh interior point to all other surface patches. If the new perturbation vector is larger than the existing vector, then it is replaced. The volume interior points are then processed by locating the nearest surface node or element. The interpolated perturbation at that location is transferred to the interior location using Equation (1).

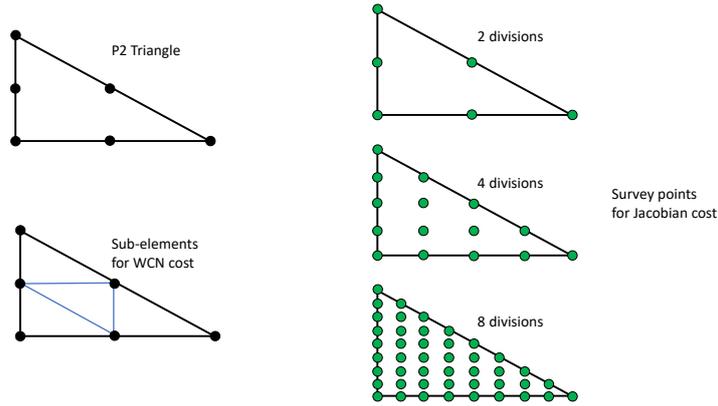
### 2.3 Optimization-based Smoothing

The smoothing method used on the high-order mesh is a node perturbation scheme that attempts to optimize a two-component cost function. The two components are evaluated on the elements and distributed to the nodes. Both components have optimal values at 1 and indicate element inversion at values of 0 and lower. The first component is based on Weighted Condition Number (WCN) [11] and the second component is based on a normalized Jacobian. The linear combination of components, shown in Equation (2), is controlled via the  $\alpha$  parameter that can vary between 0 and 1. The WCN component is required to maintain element shape but cannot ensure positive Jacobians. The Jacobian component does not maintain the shape but is sometimes required to enforce

positive Jacobians. Together they seek to enforce the proper shape and size of the elements.

$$C = \alpha C_{WCN} + (1 - \alpha) C_{JAC} \quad (2)$$

The diagrams in **Fig. 6** are used to illustrate the locations where each component would be calculated for a two-dimensional triangle. The extension to three dimensions is straightforward. For clarity, the triangular element is straight-sided but that is certainly not required. The upper-left image shows a P2 triangle with the high-order nodes located at the mid-edges. The WCN cost component is computed on the linear sub-elements, shown in the lower left image. This cost component only affects the nodes that define the sub-element. The normalized Jacobian cost component is computed at the survey points shown on the right side of the figure. The Jacobian computation involves all real nodes (linear and high-order) in the element, so it affects all nodes of the element. When operating on pyramids, prisms and hexahedra the WCN cost component is computed on a tetrahedral element constructed at each corner of these elements. The normalized Jacobian cost component is computed over the entire element using similar equally spaced survey points.



**Fig. 6.** P2 triangle with the sub-elements and Jacobian survey locations.

In practice the smoothing is applied in one or more stages. In the first stage only the WCN component is used. This is less expensive to compute and provides robustness during mesh untangling at the beginning of the smoothing process. After the first stage the normalized Jacobians are evaluated within each element using a dense survey grid created using number of subdivisions equal to the element polynomial degree times a specified maximum factor, such as 4 for the lower right image. If negative Jacobians are detected, then the second stage is performed with both cost components and the survey grid density corresponds to the polynomial degree, such as 2 for the triangle in the upper left image. If negative Jacobians still exist, then the survey grid density increases for the next stage. Experience has shown the third stage is seldom needed for P2 meshes but needed more for P3 and P4 meshes. The higher polynomial orders permit

inflections in the edges and surfaces which can contribute to negative Jacobians. Increasing the survey grid density identifies more regions of the element that might contain small or negative Jacobian values.

WCN smoothing was originally developed for linear mesh smoothing. Each high order element can be subdivided into a collection of linear sub-elements.  $C_{WCN}$  is computed on these sub-elements where a desired shape is known from the elevated but unperturbed mesh. Enforcing sub-element shapes will require edge curving near surface curvature but will maintain straight elements away from the surface. Cost values computed on the sub-elements contributes to average and minimum values at the nodes of each sub-element.

The  $C_{WCN}$  component takes on two forms, depending on the local Jacobian, Equations (3). If the Jacobian is negative the cost is the Jacobian. If the Jacobian is positive the cost is the inverse of the weighted condition number. The delineation at a value of zero is required because the second form does not recognize when the element is inverted. These two forms are C0 continuous at a value of 0.

$$C_{WCN} = J \quad \text{if } J \leq 0.0$$

$$C_{WCN} = \frac{1}{WCN} \quad \text{if } J > 0.0$$
(3)

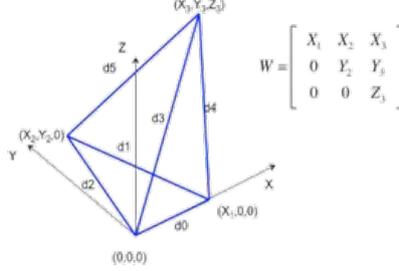
The Jacobian,  $J$ , defined in Equation (4) is the magnitude of the determinant of the Jacobian matrix,  $A$ , formed by inserting the components of the three edge vectors,  $u$ ,  $v$  and  $w$  emanating from the corner of the linear sub-element. A right-hand-rule convention is assumed in the construction of the matrix.

$$J = \begin{vmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{vmatrix}$$
(4)

The second form of the cost function in Equation (3) is based on the weighted condition number,  $WCN$ , given in Equation (5). The bracketed quantities are the Frobenius norms of the matrix products. The  $[A]$  matrix is the actual Jacobian matrix from Equation (4). The  $[W]$  matrix is the weight matrix that transforms the reference corner (a right-angled corner with unit length edges) to the desired corner shape.

$$WCN = \frac{\|AW^{-1}\| \|WA^{-1}\|}{3}$$
(5)

The weight matrix,  $[W]$ , defines the desired shape. It is constructed from the six edges of a corner tetrahedron, shown in Fig. 7 using the unperturbed (computational) coordinates. If the physical coordinates used in  $[A]$  were unperturbed from the computational coordinates the resulting  $C_{WCN}$  would be unity. For smoothing of high order meshes the sub-elements of the high-order element are evaluated using Equations (3). The elevation and projection processes deform elements near curved boundaries. The  $C_{WCN}$  components attempts to recover the original sub-element shapes.



**Fig. 7.** The weight matrix can be formed from the edge lengths of the corner.

The Jacobian component of the cost function,  $C_{JAC}$ , is a normalized Jacobian computed at numerous survey points within each element, Equation (6). The numerator is the value of the Jacobian, computed using Equation (7), at a given survey point in the element using the physical coordinates. The denominator is the value of the Jacobian at the same location using the computational (unperturbed) coordinates. The derivative components of the shape factors,  $N_i$ , are obtained from basis functions for each element type and polynomial order.

$$C_{JAC} = MIN\left(1, \frac{J_p}{J_c}\right) \quad (6)$$

$$J = \begin{vmatrix} \sum \frac{\partial N_i}{\partial \xi} x_i & \sum \frac{\partial N_i}{\partial \xi} y_i & \sum \frac{\partial N_i}{\partial \xi} z_i \\ \sum \frac{\partial N_i}{\partial \eta} x_i & \sum \frac{\partial N_i}{\partial \eta} y_i & \sum \frac{\partial N_i}{\partial \eta} z_i \\ \sum \frac{\partial N_i}{\partial \zeta} x_i & \sum \frac{\partial N_i}{\partial \zeta} y_i & \sum \frac{\partial N_i}{\partial \zeta} z_i \end{vmatrix} \quad (7)$$

The Jacobian component of the cost function,  $C_{JAC}$ , attempts to recover the original Jacobian magnitude of the unperturbed mesh. The possible range of values is negative infinity to 1. The ratio is capped at 1 so larger physical Jacobians are permitted. In essence, this cost component attempts to recover the size of the original element. This form of Jacobian is preferred over other quality measures typically encountered in curved meshes, such as scaled Jacobian. The scaled Jacobian will not respect the viscous clustering. It measures the deviation of the Jacobian within the element. In the optimization process this would drive the mesh towards isotropic element shapes.

$C_{JAC}$  is computed at the survey points within each element. Each value computed contributes to the average and minimum value recorded for each node in the element. The number of survey points within an element can vary. For the smoothing operation it is an equally spaced array of locations derived by specifying the number of divisions on an edge. For instance, if the number of divisions is 2 for a P2 element then the locations will be the corner nodes plus the high-order nodes. Three divisions for P3 elements would also result in calculations at the corners and the high-order nodes. Number of divisions higher than the polynomial order are possible, but generally not required to achieve a valid mesh and would increase the computational cost significantly.

Nodal values of the cost,  $C$ , are computed using a blending of the cost for the elements surrounding each node, Equation (8). The blending combines the worst cost value,  $C_w$ , with the arithmetic average of the element cost values,  $C_{Avg}$ . This ensures the worst value has significant weight as a surrounding element is inverted or nearly inverted.

$$\beta = 1 - C_w$$

$$C = \beta C_w + (1 - \beta) C_{Avg} \quad (8)$$

Smoothing proceeds in an iterative manner where each node is perturbed in the direction of increasing cost, driving towards the ideal value of 1. The direction is determined using operator overloaded math functions which compute values and sensitivities in the same operation. The sensitivity vector is used in a gradient based algorithm to advance the mesh. The step size is limited to a fraction of the local inscribed radius for each element. A user-specified threshold cost value, such as 0.8, is used to activate nodes in the mesh for optimization. Only nodes whose cost is below this threshold and their immediate neighbors are computed and moved during any given iteration. Each element connected to an activated node is also activated during the smoothing iteration. This dramatically improves the performance of the scheme as most of the mesh will have a unit cost value. Only nodes in the vicinity of curved boundaries will have lower cost values. Smoothing stops when all nodal cost values exceed the threshold, or the specified number of iterations are completed.

The cost threshold can greatly influence the curving process run time and resulting quality. Lower values may permit excessive kinks in the edges oriented in the normal direction from curved boundaries, especially very close to the surface. Higher threshold values will increase the run times for curving but will produce higher quality meshes. The default threshold value is 0.8 but higher values approaching 0.95 are commonly used. The  $\alpha$  parameter is automatically calculated to be the polynomial degree minus one over the polynomial degree. This would produce values of  $1/2$ ,  $2/3$  and  $3/4$  for Q2, Q3 and Q4 meshes. In extreme cases this parameter has been lowered to 0.3 – 0.4 to ensure valid Jacobians are produced. These meshes included some highly curved surface and interior edges as a result.

### 3 Results

A collection of example cases is shown with multiple curved boundaries and viscous clustering toward these boundaries. The cases increase in size from a small generic shape to a complex air vehicle. All curved boundary shapes are defined using CAD geometry. Surface evaluations are performed using the CAD geometry and Project Geode projection calls. Images were created using the Gmsh application [16] as a mesh viewer for P2, P3 and P4 meshes for the first two cases and ParaView [17] for the final case P2 meshes.

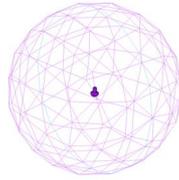
In each case the resulting mesh is valid with no negative Jacobians resulting at the densest survey grid. The minimum combined cost values are greater than 0.5 and approach or exceed the threshold values of 0.9 – 0.95. For those cases where the combined cost is lower the normalized Jacobian minimum value is in the range from 0.1 – 0.3.

### 3.1 Weeble

The first case is a small mesh created on the geometry shown in Fig. 8. It is a surface with convex and concave features separated by sharp edges. The outer boundary is a simple sphere. The linear mesh, shown in Fig. 9, has 4,876 points and 28,821 tetrahedra. The mesh on the weeble, shown in Fig. 10, is very coarse. The volume mesh is clustered towards the surface with a normal spacing equal to  $1e-04$ . The maximum element aspect ratio is 5,246 for an element on the surface.



**Fig. 8.** Surface geometry for weeble.

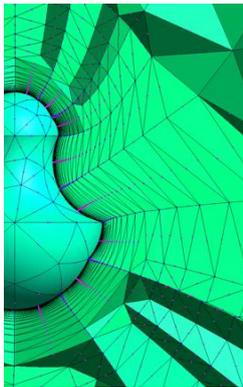


**Fig. 9.** Outer boundary linear surface mesh for weeble.

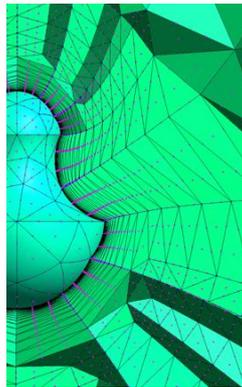


**Fig. 10.** Weeble surface linear mesh.

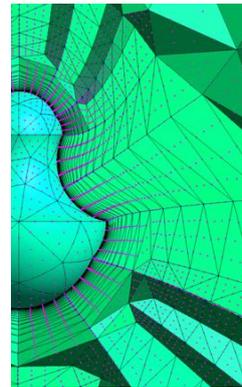
The P2, P3 and P4 meshes, are shown in Fig. 11, Fig. 12 and Fig. 13, respectively. The P2 contains 38,404 points. The P3 mesh has 129,073 points. And the P4 mesh has 305,370 points. The surfaces are smooth, and the sharp edges are retained and curved.



**Fig. 11.** Weeble P2 mesh with crinkle cut through the centerline.

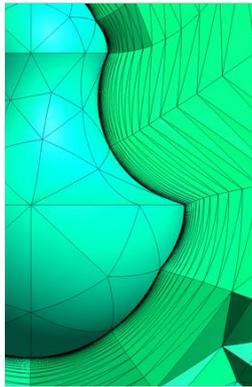


**Fig. 12.** Weeble P3 mesh with crinkle cut through the centerline.

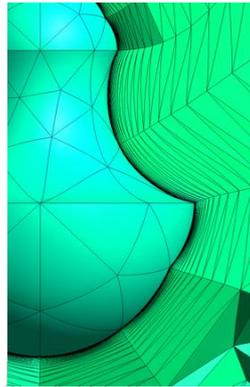


**Fig. 13.** Weeble P4 mesh with crinkle cut through the centerline.

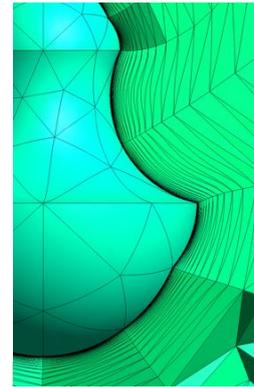
Magnified views of the cut through the volume meshes without the nodes displayed are shown in Fig. 14 through Fig. 16. The interior edges show a high degree of curving off the lower sharp edge. The P4 mesh shows an inflection in the edges just above the lower sharp edge on the surface.



**Fig. 14.** P2 volume edges near surface.

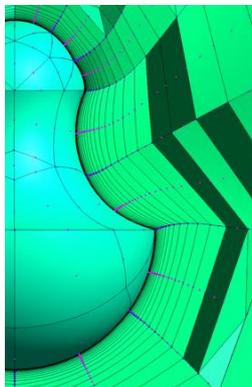


**Fig. 15.** P3 volume edges near surface.

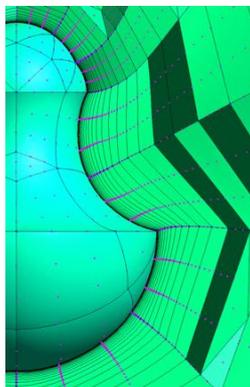


**Fig. 16.** P4 volume edges near surface.

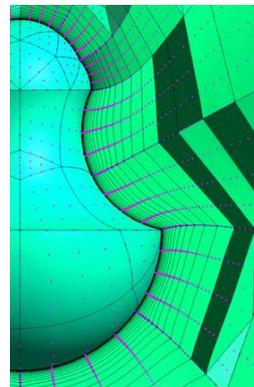
Hybrid versions of the meshes containing tetrahedra, pyramid, prisms and hexes were also curved, Fig. 17 through Fig. 19. The quadrilateral surface elements are coarser than the triangular elements from the all-tetrahedral mesh.



**Fig. 17.** P2 hybrid volume mesh.



**Fig. 18.** P3 hybrid volume mesh.



**Fig. 19.** P4 hybrid volume mesh.

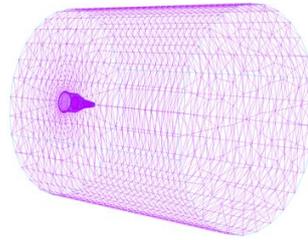
### 3.2 Generic Nozzle

The next example is a slightly bigger mesh with a significant amount of curvature on the surface. The generic nozzle geometry is shown in Fig. 20. The outer nozzle surface

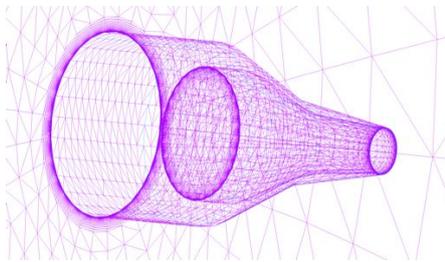
contains convex curvature and a sharp break in slope about  $\frac{3}{4}$  of the length of the surface. The interior nozzle surface contains smooth convex and concave curvature. The linear mesh, shown in Fig. 21, has 86,817 points and 506,623 tetrahedra. The mesh is clustered to the nozzle surface with a wall spacing of  $1e-03$ . The maximum aspect ratio is 385.6 located on the inner wall near the convex curvature region. A magnified view of the surface meshes is shown in Fig. 22. The challenging aspect of this case is the nozzle base region, magnified in Fig. 23. It has convex curvature in the plane on the interior edge and concave curvature on the outer edge. This requires more than just projecting the high-order nodes to the surface. It requires smoothing on the surface to create valid curved triangular elements. The smoothing process works on all nodes in the mesh, boundary nodes and interior volume nodes. Any tangling on the surface will be corrected as the volume mesh is smoothed. The boundary nodes are not processed separately from the interior nodes.



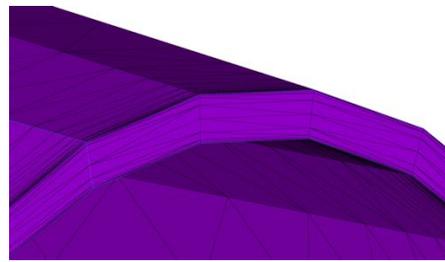
**Fig. 20.** Generic nozzle surface geometry.



**Fig. 21.** Generic nozzle linear mesh.

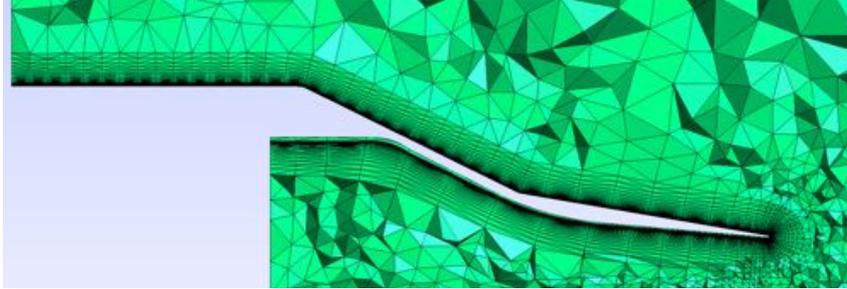


**Fig. 22.** Magnified view of surface mesh inside and outside of nozzle.

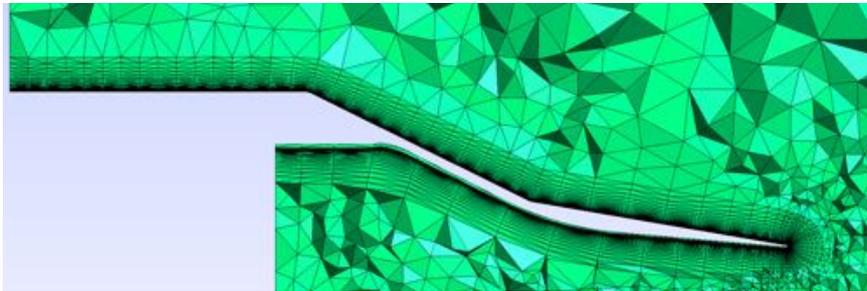


**Fig. 23.** Zoomed in view of nozzle base region.

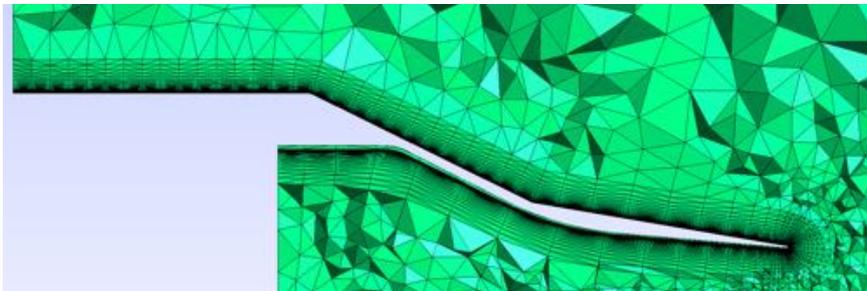
Crinkle cuts through the centerline of the P2, P3 and P4 meshes are shown in Fig. 24 through Fig. 26. The only noticeable difference is at the concave shoulder on the interior side of the nozzle. The P2 surface has a slight bulge that is not seen in the P3 and P4 meshes. This occurred because the length of the elements in the region spanned the transition from the horizontal flat piece to the converging section. The P2 mesh positioned the interior edge node mid-way between the corner nodes while the upstream node was on the flat section. The quadratic surface (P2) creates the bulging section. The P3 and P4 meshes show no bulge.



**Fig. 24.** Crinkle cut through P2 mesh.

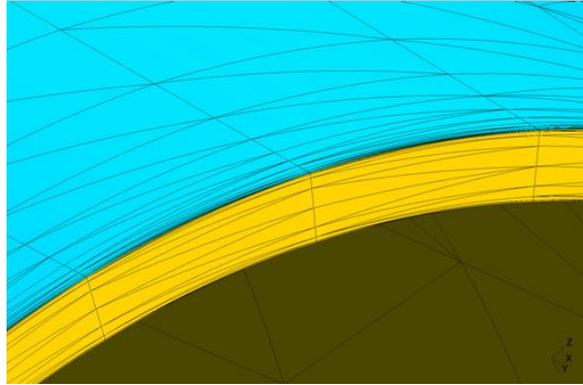


**Fig. 25.** Crinkle cut through P3 mesh.

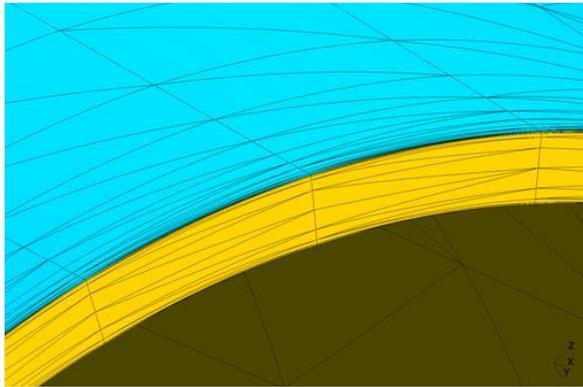


**Fig. 26.** Crinkle cut through P4 mesh.

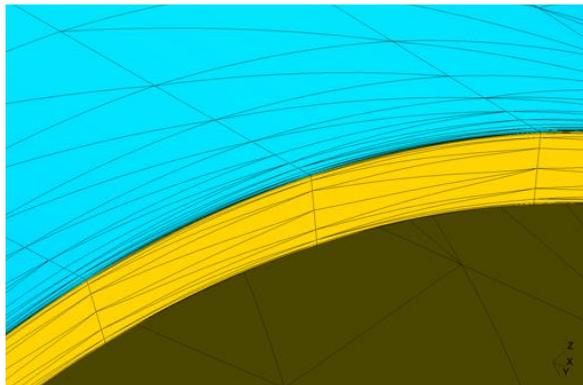
The nozzle base region for each mesh is shown in Fig. 27 through Fig. 29. All three meshes show the same character. The clustering in the plane to the inner and outer curves is maintained. The triangular elements in the middle of the base tend to maintain the straight-edged shapes.



**Fig. 27.** Base region of P2 mesh.



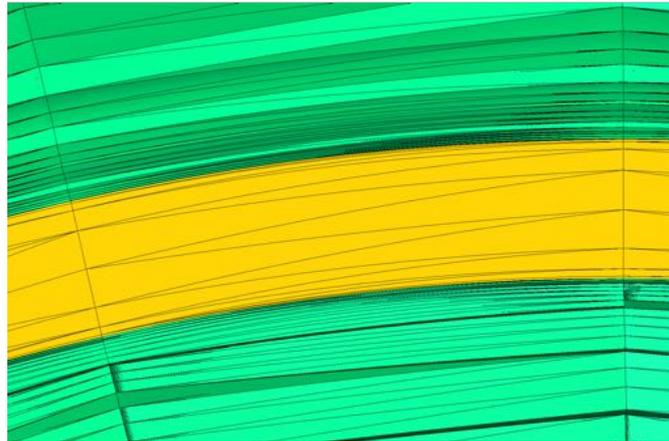
**Fig. 28.** Base region of P3 mesh.



**Fig. 29.** Base region of P4 mesh.

The volume elements near the base deform to match the surface shape, shown in Fig. 30. The crinkle cut through an X coordinate is just ahead of the base and reveals convex

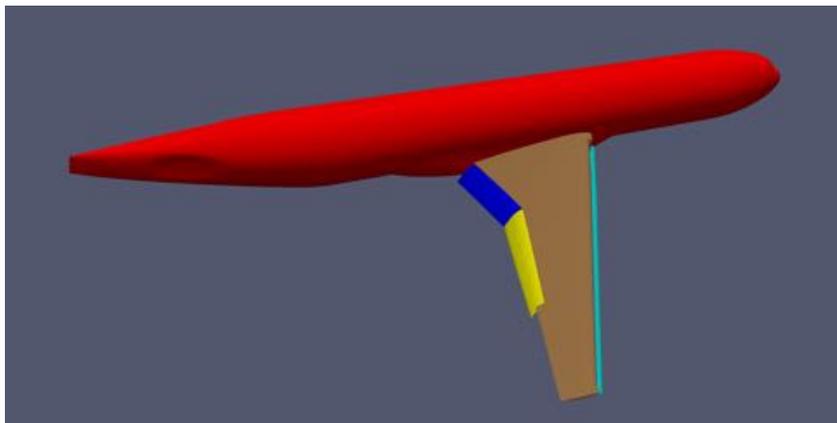
curvature on the outside and concave curvature on the inside. The P2 mesh transitions from curved edges at the boundary to straight edges in a smooth manner.



**Fig. 30.** Crinkle cut through P2 mesh at nozzle base.

### 3.3 High Lift Common Research Model

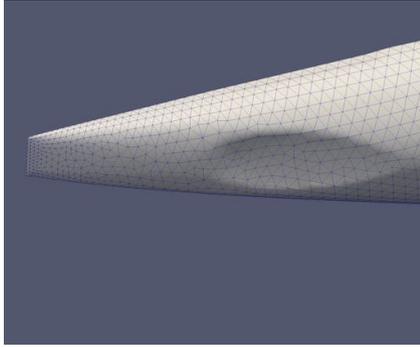
The last case is a complex air vehicle, the NASA/Boeing High-Lift Common Research Model (HL-CRM) [18]. The geometry is shown in Fig. 31, colored by the different components. It contains a fuselage, wing, leading edge slat and two trailing edge flaps. The linear mesh contains 5,918,418 nodes and 35,031,781 tetrahedra. This was the “coarse” mesh in a sequence. The elevated P2 mesh contains 47,062,366 nodes.



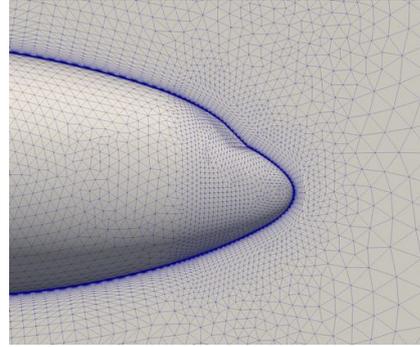
**Fig. 31.** NASA/Boeing High Lift Common Research Model geometry.

Surface meshes are shown in Fig. 32 through Fig. 35. The viscous clustering, evident in the symmetry plane of Fig. 33, had an initial cell height of 0.00175. The total fuse-

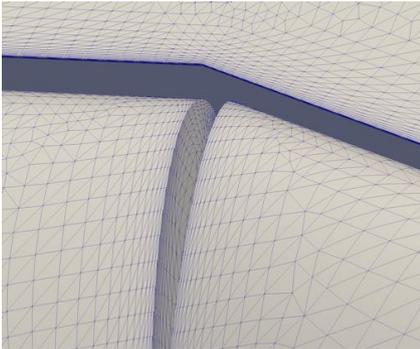
lage length was approximately 2500 units long. The maximum aspect ratio in the symmetry plane mesh was 8,941, which is indicative of the viscous clustering over the entire aircraft surface.



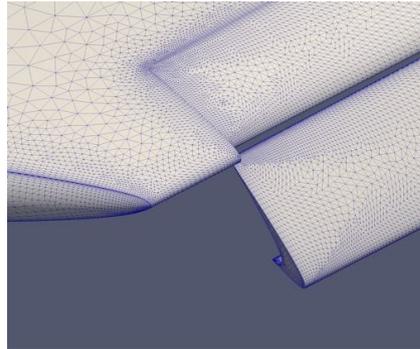
**Fig. 32.** P2 mesh on HL-CRM aft fuselage.



**Fig. 33.** P2 mesh on HL-CRM forward fuselage and symmetry plane.



**Fig. 34.** P2 mesh on HL-CRM flap gap.



**Fig. 35.** P2 mesh on HL-CRM wing tip leading edge.

The challenging aspect of this case was the viscous clustering and overall size of the mesh. The curving computer code is serial and required approximately 13 GBytes. The surface resolution was comparable to the resolution typically utilized for linear meshes. This results in good resolution of surface curvature which is easily handled by the curving method.

## 4 Conclusions

A high-order mesh elevation and curving method has been described that is applicable to high Reynold's number viscous cases for finite element CFD solvers. Viscous clustering introduces significant challenges to mesh curving, especially in regions of high

surface curvature. The elevation process in convex regions of viscous meshes can invalidate the high order mesh by inverting volume elements near the curved surface. The curving process uses an optimization-based node perturbation scheme to reposition nodes. The cost function contains two components. The first component uses a Weighted Condition Number function that imposes a desired shape, defined by the elevated but unperturbed mesh. The second cost component uses a normalized Jacobian that imposes sizing restrictions where the high order mesh has been squeezed but the surface curvature. It also guarantees the resulting mesh contains all positive Jacobians; a requirement for the finite element solvers.

Geometry for the curving program is defined by CAD groups (surfaces and edges). Surface node projections are performed using a lightweight geometry kernel, Project Geode. The CAD groups are stored in separate BSP trees for efficient searches. A mesh linkage defines the specific CAD group for each surface mesh node.

Three example cases were curved; all containing viscous clustering to geometries containing high concave and convex surface curvature. High order meshes of polynomial degrees 2, 3 and 4 were generated for the first two cases. A P2 mesh was created for the larger aircraft case. Most of the curved meshes were all-tetrahedral meshes. A hybrid mesh containing tetrahedra, pyramids, prisms and hexes was generated for the first case. All meshes generated were valid with positive Jacobians.

## 5 Acknowledgement

This work was partially supported by a NASA SBIR contract “High Order Mesh Curving and Geometry Access”, Phase I contract NNX17CL83P and Phase II contract 80NSSC18C0109. NASA’s support is greatly appreciated.

## References

1. Ryan Glasby, Nicholas Burgess, Kyle Anderson, Li Wang, Steven Allmaras, and Dimitri Mavriplis. "Comparison of SU/PG and DG Finite-Element Techniques for the Compressible Navier-Stokes Equations on Anisotropic Unstructured Meshes", 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings, (AIAA 2013-0691) <https://doi.org/10.2514/6.2013-691>.
2. Jon T. Erwin, Li Wang, W Kyle Anderson, and Sagar Kapadia. "High-Order Finite-Element Method for Three-Dimensional Turbulent Navier-Stokes", 21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, (AIAA 2013-2571) <https://doi.org/10.2514/6.2013-2571>.
3. Glasby, Ryan S., Erwin, J. Taylor, Stefanski, Douglass L., Allmaras, Steven R., Galbraith, Marshall C., Anderson, W. Kyle, and Nichols, Robert H., “Introduction to COFFE: The Next-Generation HPCMP CREATETM-AV CFD Solver”, AIAA-2016-0567, AIAA SciTech 2016.
4. Hormann, Kai, and Floater, Michael S., “Mean Value Coordinates for Arbitrary Polygons”, <http://www.cs.jhu.edu/misha/ReadingSeminar/Papers/Hormann06.pdf>, Submitted to ACM Transactions on Graphics.

5. Chen, Chih-Hao, "A Radial Basis Functions Approach to Generating High-Order Curved Element Meshes for Computational Fluid Dynamics", Master's Thesis, Imperial College, London, 2013.
6. Per-Olof Persson and Jaime Peraire. "Curved Mesh Generation and Mesh Refinement using Lagrangian Solid Mechanics", 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings, (AIAA 2009-0949) <https://doi.org/10.2514/6.2009-949>.
7. Moxey, D., Ekelschot, D., Keskin, U., Sherwin, S. J., and Peiro, J. "High-order curvilinear meshing using a thermo-elastic analogy," *Computer-Aided Design*, 2015. doi: <http://dx.doi.org/10.1016/j.cad.2015.09.007>.
8. Fortunato, M. and Persson, P.-O., "High-order Unstructured Curved Mesh Generation Using the Winslow Equations", *Journal of Computational Physics*, Vol. 307, pp 1-14, 2016.
9. Eloi Ruiz-Girones, Joseph Sarrate and Xevi Roca, "Generation of curved high-order meshes with optimal quality and geometric accuracy," 25<sup>th</sup> International Meshing Roundtable (IMR25), 2016.
10. Steve L. Karman, J T. Erwin, Ryan S. Glasby, and Douglas Stefanski. "High-Order Mesh Curving Using WCN Mesh Optimization", 46th AIAA Fluid Dynamics Conference, AIAA AVIATION Forum, (AIAA 2016-3178) <https://doi.org/10.2514/6.2016-3178>.
11. Freitag, L. A., and Knupp, P. M. "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number," *8th International Meshing Roundtable*. South Lake Tahoe, CA, 1999.
12. Steve L. Karman. "Adaptive Optimization-Based Improvement of Tetrahedral Meshes", *AIAA Journal*, Vol. 54, No. 5 (2016), pp. 1578-1590. <https://doi.org/10.2514/1.J054439>.
13. Steve L. Karman and Michael G. Remotigue. "Optimization-Based Smoothing for Extruded Meshes", 54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2016-1671) <https://doi.org/10.2514/6.2016-1671>.
14. Pointwise. <http://www.pointwise.com>.
15. "CFD General Notation System Standard Interface Data Structures.", <http://cgns.github.io/>.
16. Gmsh. <http://www.gmsh.info>.
17. ParaView. <http://www.paraview.org>.
18. Doug S. Lacy and Anthony J. Sclafani. "Development of the High Lift Common Research Model (HL-CRM): A Representative High Lift Configuration for Transonic Transports", 54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2016-0308) <https://doi.org/10.2514/6.2016-0308>