JOURNAL OF AIRCRAFT
Vol. 45, No. 3, May–June 2008

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Mesh Generation and Deformation Algorithm for Aeroelasticity Simulations

Paul G. A. Cizmas* and Joaquin I. Gargoloff†
Texas A&M University, College Station, Texas 77843-3141

DOI: 10.2514/1.30896

Introduction

HIGH-FIDELITY flow solvers for aeroelastic applications require the use of computational meshes that deform as the structure is being displaced. High-aspect-ratio wings increase the demands on the robustness of the mesh-deforming algorithm, because these wings are extremely flexible and attain deformations that are a significant fraction of the span of the wing. The mesh deformation algorithm must be not only robust but also computationally inexpensive to avoid penalizing the turnaround time of the aeroelastic computations.

Different approaches have been developed to solve the moving-mesh problem. For meshes generated by using overlapping grids, a natural way to allow for grid motion is to slide the overlapping region of the grids [1,2]. The advantage of this method is that the body-fitted meshes do not deform during the body motion. A disadvantage of this approach is that the interpolation algorithm that communicates the solution between grids has to be updated for each overlapping position of the grids.

The tension spring analogy [3] is one of the most widely used mesh deformation strategies. In this approach, each edge of the mesh is represented by a spring for which the stiffness is proportional to the reciprocal of the length of the edge. By replacing the edges with springs, a deformation of the boundary translates into a deformation of the spring network, which adjusts its shape to the equilibrium position of the network. The displacements in each direction are decoupled and the equation that updates the position of the nodes is relatively easy to solve. A disadvantage of this method is that for highly distorted meshes, collapsed or negative volume cells may appear, especially on high-aspect-ratio cells such as those used for viscous flows.

An improvement over the tension spring analogy is the torsion spring analogy [4,5]. The torsion spring analogy consists of adding a torsional spring to the tension-spring-analogy technique. The stiffness of the torsional spring is related to the angle between the edges. As the angle tends to zero, the stiffness tends to infinity, thus preventing vertices from crossing over edges and avoiding negative-volume cells. The disadvantage of this method is the higher complexity and computational cost than with the tension spring analogy.

The transfinite-interpolation mesh deformation technique is based on the linear interpolation of the boundary motion [6]. The motion of a node located between a moving and a fixed boundary is equal to the motion of the moving boundary times a scale factor. This scale factor, assigned to each node of the mesh, depends on the distances from the node to the moving and the fixed surfaces. The scale factor is 1.0 for nodes on the moving boundary and 0.0 for nodes on the fixed boundary. The method guarantees a smooth transition between the moving boundaries and the fixed boundaries. One disadvantage of this method is that it cannot guarantee the mesh orthogonality at deforming surfaces, a condition that is important for viscous flows.

Another approach to simulate mesh deformation is to use the linear elasticity equations [7]. The deformed grid is obtained by solving the equilibrium equations for the stress field. The modulus of elasticity is chosen to be inversely proportional to the cell volume or to the distance from the deforming boundaries. Therefore, the cells close to the moving boundaries have small deformations, and the majority of the mesh deformation is relegated to the regions farther away from the moving boundary.

This Note presents a grid generation and deformation algorithm for wings with large deformations. The computational domain was discretized using a hybrid grid that consisted of structured hexahedra around the wing and unstructured triangular prisms elsewhere. The mesh was divided in layers that were topologically identical in the spanwise direction. The mesh deformation algorithm was applied in two steps. First, the spring analogy technique was applied to deform the nodes within a mesh layer. Second, the layers were deformed to be perpendicular to the boundaries of the domain and to the surface of the wing. The Note describes the mesh generation algorithm and the mesh deformation algorithm and shows results for a wing with large tip deformation.

Mesh Generation Algorithm

The mesh generation algorithm presented herein was developed to satisfy the following requirements:

1) Allow large wing deformations without the need to remesh the domain.

2) Allow a good control of the grid size in the boundary layer.

3) Facilitate parallel computation for the flow solver.

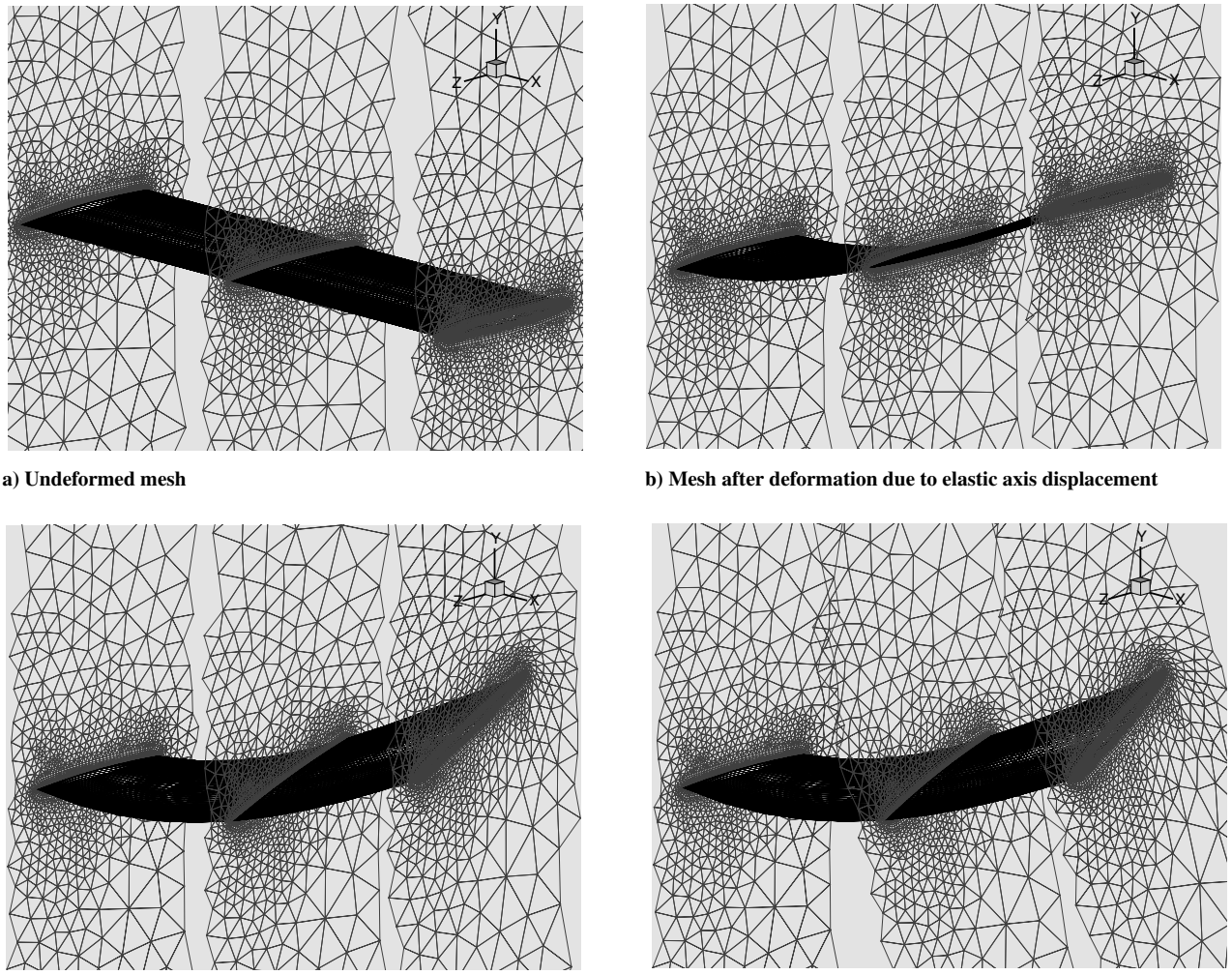
To satisfy these requirements, the grid generation and deformation algorithm used 1) layers of topologically identical elements in the spanwise direction, 2) a structured O-grid around the wing surface, and 3) an unstructured grid outside of the O-grid mesh that was deformed using the spring analogy technique.

The topologically identical layers spanned from the root of the wing past the tip of the wing to the lateral boundary of the computational domain. The use of these layers simplified the mesh deformation algorithm and the parallel communication in the flow solver. Each layer of the mesh included both structured and unstructured grids, as shown in Fig. 1. A structured O-grid was constructed around the wing to better control the mesh size in the boundary layer. An unstructured grid was generated outside the O-grid to fill the rest of the domain [8]. The cells in the domain were generated by extruding the two-dimensional rectangles and triangles

Presented as Paper 556 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 8–11 January 2007; received 8 March 2007; revision received 23 December 2007; accepted for publication 27 December 2007.

*Associate Professor, Department of Aerospace Engineering. Associate Fellow AIAA.

†Graduate Research Assistant, Department of Aerospace Engineering. Member AIAA.



a) Undeformed mesh

b) Mesh after deformation due to elastic axis displacement

c) Mesh after deformations due to elastic axis displacement and wing rotation about *x*-axis

d) Mesh after deformations due to elastic axis displacement and wing rotations about *x*-, *y*-, and *z*-axes

Fig. 1 Mesh deformations due to elastic-axis displacement and wing rotations; root, midspan, and tip layers are shown.

along the spanwise direction. Hexahedral cells were generated from the rectangular faces, and triangular prisms were generated from the triangular faces.

The unstructured mesh covered the domain with relatively few elements that grew in size as the distance to the wing increased. Figure 1a shows the undeformed mesh. Note the identical topology of the layers of the mesh. This mesh had 16 layers that spanned from the root of the wing past the tip of the wing. For clarity, Fig. 1 shows only three layers corresponding to wing root, midspan, and wing tip.

Mesh Deformation Algorithm

The computational mesh was deformed as a result of the wing displacement. The mesh connectivity was not modified in this process. The mesh deformation algorithm was split in two steps: first, a translation along the y–z plane and then rotations about the x, y, and z axes. The spring analogy technique was used to deform the unstructured mesh. This section presents the algorithm developed to deform the mesh as the wing changed position and shape. This section describes 1) the spring analogy technique, 2) the details of the deformations due to the wing translation and rotation, 3) the computation of the rotation angles, and 4) the cubic-mapping function used to deform the mesh.

Spring Analogy Technique

The spring analogy technique [3] was used to deform the unstructured mesh. This technique models each edge of the unstructured mesh as a spring. As the boundaries of the unstructured mesh move, the whole network of springs adjusts to the instantaneous position of the moving boundary. The spring stiffness for a given edge connecting nodes i and j was inversely proportional to the length of the edge:

$$k_{ij} = \frac{1}{\sqrt{(y_j - y_i)^2 + (z_j - z_i)^2}}$$

The exterior boundary of the unstructured domain was held fixed because it did not deform. The interior boundary of the unstructured domain followed the motion of the wing. The updated coordinates of the nodes of the unstructured mesh were found by solving the static equilibrium equations of the spring network:

$$F_{y_i} = \sum_{j=1}^{N} k_{ij} \cdot (\Delta y_i - \Delta y_j) \qquad \Delta y_i = y_i^{(n)} - y_i^{(n-1)} \tag{1}$$

$$F_{z_i} = \sum_{j=1}^{N} k_{ij} \cdot (\Delta z_i - \Delta z_j) \qquad \Delta z_i = z_i^{(n)} - z_i^{(n-1)} \tag{2}$$

where F_{y_i} is the force exerted along the y axis, F_{z_i} is the force exerted along the z axis, N is the number of nodes surrounding node i, Δy_i and Δz_i are the node displacements, n is the current mesh deformation step, and $n-1$ is the previous step.

The springs were in tension if the edge was stretched and were in compression if the edge was shrunk. The spring network was in equilibrium if $F_{y_i} = 0$ and $F_{z_i} = 0$ for each node of the mesh. Therefore, Eqs. (1) and (2) yield the node displacements Δy_i and Δz_i:

$$\Delta y_i = \left(\sum_{j=1}^{N} k_{ij} \cdot \Delta y_j \right) / \sum_{j=1}^{N} k_{ij} \tag{3}$$

$$\Delta z_i = \left(\sum_{j=1}^{N} k_{ij} \cdot \Delta z_j \right) / \sum_{j=1}^{N} k_{ij} \tag{4}$$

Equations (3) and (4) were solved iteratively until they reached their final value, corresponding to the equilibrium state of the spring network. The node positions were updated using

$$y_i^{(n)} = y_i^{(n-1)} + \Delta y_i \qquad z_i^{(n)} = z_i^{(n-1)} + \Delta z_i$$

Deformations Because of Elastic-Axis Displacement

The first step of the mesh deformation algorithm was a translation of the nodes along the y–z plane, following the elastic-axis displacement. The O-grid layers were translated and possibly deformed, depending on the structural model used for the wing. The unstructured grid was deformed in the y and z directions using the spring analogy technique. The nodes of each layer were also translated in the x direction according to the spanwise wing deformation. The magnitude of the translational deformations were set by the new coordinates of the deformed elastic axis (X_{EA}, Y_{EA}, and Z_{EA}), which were calculated by the structural solver.

Figure 1b shows the mesh after the deformations due to the wing translations. The three layers correspond to the wing root, midspan, and wing tip. The wing-tip deformation along the y axis was 20% of the wing semispan.

Deformations Because of Wing Rotations

The second step of the mesh deformation algorithm was the rotation of the layers of nodes about the x, y, and z axes. The spring analogy technique was used to deform the unstructured mesh, and a cubic-mapping function was used to deform the layer in the x and z directions. The rotations were done in the following order: 1) a rotation about the x axis, 2) a rotation about the y axis, and 3) a rotation about the z axis.

Rotation About the x Axis

The first rotation applied to the mesh layer was a rotation about the x axis, which is analogous to a torsional deformation. This rotation about the x axis took place about the elastic axis. Figure 1c shows the mesh after the deformations due to wing x-axis rotations. The torsional x-axis rotation along the elastic axis was 28.6 deg at the wing tip. In this case, the O-grid rectangular cells were rotated without being deformed because the structural solver assumed zero chordwise deformation [9]. The unstructured triangular cells were deformed in the y and z directions using the spring analogy technique.

The mesh layer was initially in the y–z plane. Consequently, the vector normal to the mesh layer was oriented along the x axis. The mesh layer was assumed to have an initial rotation θ_0 about the x axis. The initial leading-edge vector and normal vector before the x-axis rotation had the following components:

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}^{(0)}_{\mathrm{LE}} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{\mathrm{EA}} + \begin{Bmatrix} 0 \\ d \cdot \sin\theta_0 \\ -d \cdot \cos\theta_0 \end{Bmatrix} \qquad \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix}^{(0)} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

where the superscript 0 indicates that the vector refers to the position before the x-axis rotational deformation, LE denotes the leading edge, EA denotes the elastic axis, and d is the distance from the leading edge to the elastic axis.

The final position of the leading edge after a rotation along the x axis was obtained by rotating the mesh layer by the amount $\theta_x - \theta_0$ about the x axis. The coordinates of the leading edge were

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}^{(1)}_{\mathrm{LE}} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{\mathrm{EA}} + \begin{Bmatrix} 0 \\ d \cdot \sin\theta_x \\ -d \cdot \cos\theta_x \end{Bmatrix}$$

where the superscript 1 indicates that the vector refers to the position after the x-axis rotational deformation. The rotation with respect to the x axis did not change the normal vector.

Rotation About the y Axis

The second rotation applied to the mesh layer was a rotation about the y axis, which is analogous to an in-plane bending deformation. The amplitude of this rotation is usually small because the wing has a high in-plane stiffness.

The final position of the leading edge after a rotation along the y axis was obtained by rotating the mesh layer by the amount θ_y about the y axis. The coordinates of the leading edge were

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}^{(2)}_{\mathrm{LE}} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{\mathrm{EA}} + \begin{Bmatrix} -d \cdot \cos\theta_x \cdot \sin\theta_y \\ d \cdot \sin\theta_x \\ -d \cdot \cos\theta_x \cdot \cos\theta_y \end{Bmatrix}$$

where the superscript 2 indicates that the vector refers to the position after the y-axis rotational deformation. The normal vector after a rotation along the y axis was

$$\begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix}^{(2)} = \begin{Bmatrix} \cos\theta_y \\ 0 \\ -\sin\theta_y \end{Bmatrix}$$

Thus the normal vector was in the x–z plane.

Rotation About the z Axis

The third rotation applied to the mesh layer was a rotation about the z axis, which is analogous to an out-of-plane bending deformation. This rotation took place about the elastic axis. Figure 1d shows the mesh after the deformation due to wing rotation along the z axis.

The final position of the leading edge after a rotation along the z axis was obtained by rotating the mesh layer by θ_z about the z axis. The final coordinates of the leading edge were

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}^{(3)}_{\mathrm{LE}} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}_{\mathrm{EA}} + \begin{Bmatrix} -d \cdot \cos\theta_x \cdot \sin\theta_y \cdot \cos\theta_z - d \cdot \sin\theta_x \cdot \sin\theta_z \\ -d \cdot \cos\theta_x \cdot \sin\theta_y \cdot \sin\theta_z + d \cdot \sin\theta_x \cdot \cos\theta_z \\ -d \cdot \cos\theta_x \cdot \cos\theta_y \end{Bmatrix} \tag{5}$$

where the superscript 3 indicates that the vector refers to the position after the z-axis rotational deformation. The normal vector after a rotation along the z axis was

$$\begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix}^{(3)} = \begin{Bmatrix} \cos\theta_y \cdot \cos\theta_z \\ \cos\theta_y \cdot \sin\theta_z \\ -\sin\theta_y \end{Bmatrix} \tag{6}$$

Computation of the Rotation Angles

This section describes how to calculate the rotation angles θ_x, θ_y, and θ_z using Eqs. (5) and (6). The structural solver must provide the coordinates of the elastic axis (X_{EA}, Y_{EA}, and Z_{EA}) and the coordinates of the leading edge (X_{LE}, Y_{LE}, and Z_{LE}) along the wing

span. The components (n_x, n_y, and n_z) of the vector tangent to the elastic axis and normal to the wing layer are then calculated using the coordinates of the elastic axis.

The last row of Eq. (6) yields $\theta_y = \arcsin(-n_z)$, where it was assumed that $-90\,\text{deg} \le \theta_y \le 90\,\text{deg}$. The second row of Eq. (6) yields $\theta_z = \arcsin(n_y/\cos\theta_y)$, where it was assumed that $-90\,\text{deg} \le \theta_z \le 90\,\text{deg}$. The last row of Eq. (5) yields

$$-d \cdot \cos\theta_x = (Z_{\text{LE}} - Z_{\text{EA}})/\cos\theta_y \tag{7}$$

Equation (7) provides the absolute value of θ_x but not the sign of θ_x. To find the absolute value of θ_x and its sign, Eq. (7) was combined with the second row of Eq. (5):

$$Y_{\text{LE}} - Y_{\text{EA}} = \frac{Z_{\text{LE}} - Z_{\text{EA}}}{\cos\theta_y} \cdot \sin\theta_y \cdot \sin\theta_z + d \cdot \sin\theta_x \cdot \cos\theta_z \tag{8}$$

The rotation angle θ_x was calculated from Eq. (8):

$$\theta_x = \arcsin\left(\frac{Y_{\text{LE}} - Y_{\text{EA}}}{d \cdot \cos\theta_z} - \frac{Z_{\text{LE}} - Z_{\text{EA}}}{d} \cdot \tan\theta_y \cdot \tan\theta_z\right) \tag{9}$$

where it was assumed that $-90\,\text{deg} \le \theta_x \le 90\,\text{deg}$.

Cubic-Mapping Function

All the nodes of the initial undeformed layers were coplanar. The layer nodes remained coplanar during the deformations due to elastic-axis displacement and rotation about the x axis. The nodes situated on the wing surface remained coplanar during the rotations about the y and z axes. The rest of the nodes ceased to be coplanar after the rotations about the y and z axes. The position of these nodes was modified in the x direction so that the layers remain normal to the wing and the exterior boundary surfaces. This grid deformation in the x direction as a function of the y-independent variable was specified using the cubic function

$$X(y) = A + By + Cy^2 + Dy^3 \tag{10}$$

A similar function was used for the deformation in the x direction as a function of the z-independent variable.

Figure 2 shows the mesh after applying the cubic-mapping function (10). Note that all the mesh layers are perpendicular to the wing and the exterior boundary of the domain. Herein, x_w and y_w denote the coordinates of a node on the wing surface, x_b and y_b denote the coordinates of a node on the exterior boundary, α_w denotes the angles between the y axis and the normal to the wing surface, and α_b denotes the angles between the y axis and the normal to the exterior boundary.

The coefficients of the mapping function were calculated by imposing the following boundary conditions:

1) On the wing, the value of the cubic-mapping function was set equal to the wing coordinate x_w, and the slope of the mapping function was set to the wing slope α_w.

2) On the exterior boundary, the value of the cubic-mapping function was set equal to the boundary coordinate x_b, and the slope of the mapping function was set to α_b.

The expressions for these coefficients were [10]

$$A = x_b - By_b - Cy_b^2 - Dy_b^3 \qquad B = \tan\alpha_b - R_{\alpha y}y_b + 3Dy_w y_b$$

$$C = \frac{R_{\alpha y}}{2} - \frac{3}{2}D(y_b + y_w)$$

$$D = \frac{2}{(y_b - y_w)^2}\left(\frac{\tan\alpha_b + \tan\alpha_w}{2} - \frac{x_b - x_w}{y_b - y_w}\right)$$

where

$$R_{\alpha y} = \frac{\tan\alpha_b - \tan\alpha_w}{y_b - y_w}$$

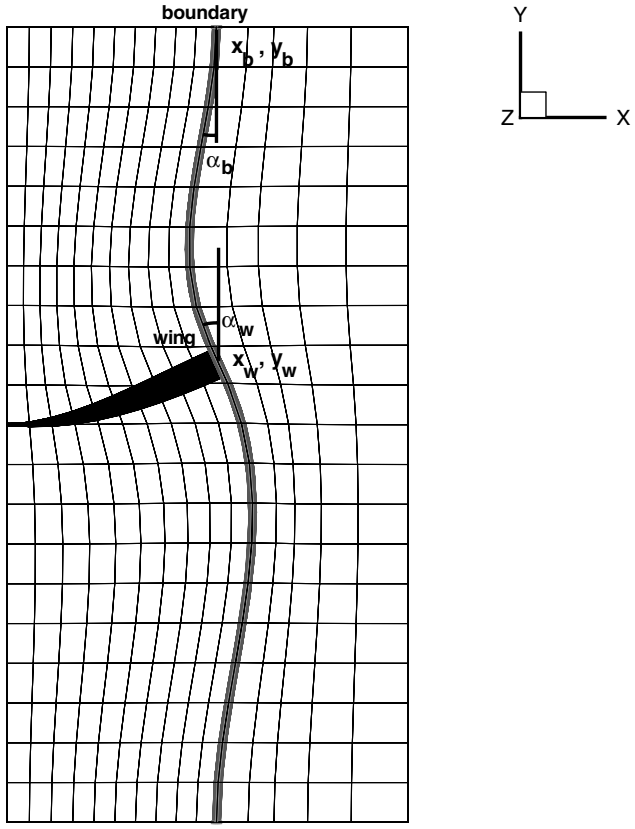


Fig. 2 Cubic-mapping function and front view of rotational deformations.

Results

This section presents the results of the mesh deformation algorithm applied to the Goland [11] wing. This algorithm was also applied to nonsymmetric tapered wings, but those results were not included here due to space limitations. For the Goland wing, the quality of the mesh was checked as the wing deformed. The computational times for the grid generation and grid deformation algorithms were also reported.

Mesh Quality Computation

As the wing deforms, the mesh cells are twisted, stretched, and/or compressed. The quality of the flow solution depends on the quality of the mesh. High-quality meshes produce more accurate flow results. This section defines the mesh-quality measures used herein to assess the merit of the grid generation and deformation algorithm.

Two measures were used to evaluate the quality of the triangular prisms: one measure was based on areas and the other measure was based on angles. The quality measure based on areas was defined as the ratio of the area of the triangular face to the sum of the squares of the length of each edge [8]:

$$Q_A = \frac{4\sqrt{3}A}{L_{12}^2 + L_{23}^2 + L_{31}^2}$$

where A is the area of the triangle face, and L_{12}, L_{23}, and L_{31} are the lengths of the edges of the triangle. This measure has a maximum value of 1.0 for equilateral triangles.

The quality measure based on angles was defined as [10]

$$Q_{\alpha i} = 1 - \frac{|\alpha_i - 60\,\text{deg}|}{120\,\text{deg}}$$

where α_i is one of the internal angles of the triangular face, and $Q_{\alpha i}$ has a maximum value of 1.0 for equilateral triangles. For each triangle, two quality measures based on angle values were calculated: an averaged value,

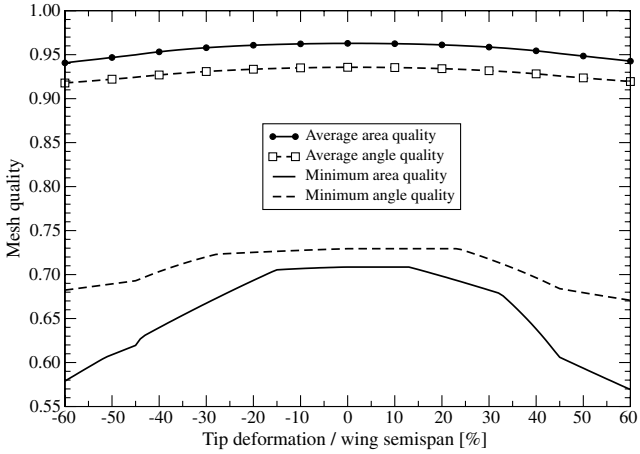


Fig. 3 Variation of mesh quality as the wing deformed.

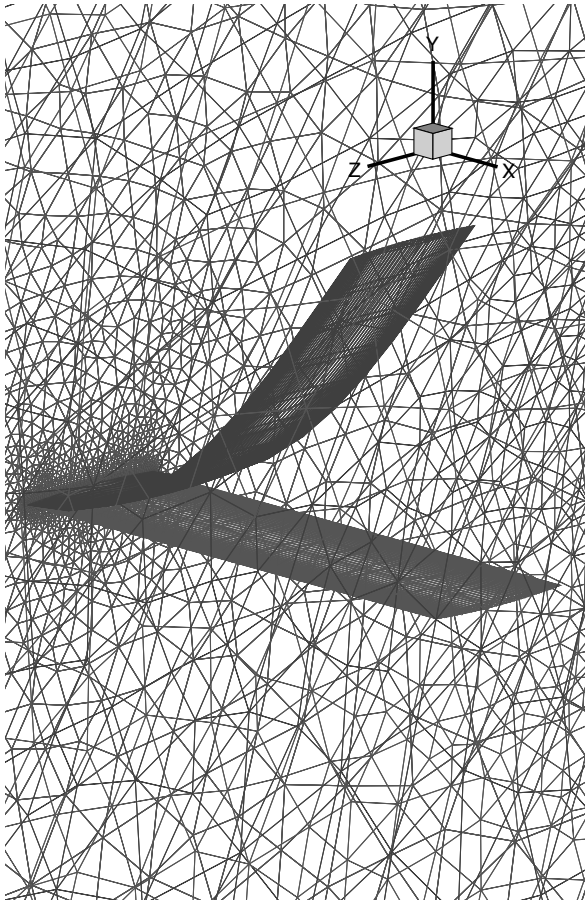


Fig. 4 Goland [11] wing with a tip deformation equal to 60% of the wing semispan.

$$Q_{\alpha\text{avg}} = \frac{Q_{\alpha 1} + Q_{\alpha 2} + Q_{\alpha 3}}{3}$$

and a minimum value, $Q_{\alpha\,\min} = \min(Q_{\alpha 1}, Q_{\alpha 2}, Q_{\alpha 3})$.

Goland Wing

The Goland [11] wing is a widely used test case for aeroelastic applications. This wing has a rectangular planform and an elliptic airfoil section. The wing had a semispan of 20 ft and a chord of 6 ft.

The Goland [11] wing was used to monitor the variation of the area and angle quality measures of the triangular prisms as the wing deformed. Figure 3 shows four quality measures: 1) the average of area quality Q_A, 2) the minimum of area quality Q_A, 3) the average of angle quality $Q_{\alpha\text{avg}}$, and 4) the minimum of angle quality $Q_{\alpha\,\min}$. These quality measures were calculated for wing-tip deformations of up to 60% of the wing semispan. The average quality measures varied between 92 and 96%. The average of the quality measure based on areas was higher than the average of the quality measure based on angles. The average quality measures were slightly affected by the wing deformation. The minimum quality measures varied between 58 and 73%. As the wing deformation increased, the minimum area quality deteriorated more than the minimum angle quality.

Figure 4 shows the deformed mesh for a tip deformation equal to 60% of the wing semispan. The darker mesh is the deformed mesh, and the lighter mesh is the undeformed mesh. This case proves that the mesh deformation algorithm is robust and can be applied to wings with large deformations.

The mesh of the Goland [11] wing shown in Fig. 4 had 60,376 nodes. The mesh generation took 20 s on a PowerMac workstation with a 2.0-GHz processor, whereas the mesh deformation took only 4 s. Consequently, a speedup factor of 5 was obtained by deforming the mesh, as opposed to remeshing it.

Conclusions

This Note presents a grid generation and deformation algorithm designed for aeroelastic applications that model highly deforming high-aspect-ratio wings. The mesh deformation algorithm was applied to wings with tip deformation of up to 60% of the semispan. The grid generation algorithm was designed to facilitate parallel processing for the flow solver by splitting the computational domain in slices that were topologically identical. A speedup of 5 was obtained for grid deformation as opposed to grid remeshing. In addition, the grid remeshing algorithm had the benefit of keeping the grid topology unchanged, which is advantageous for a parallel flow solver.

References

[1] Collard, J., and Cizmas, P. G. A., "The Effects of Vibrating Blades on Turbomachinery Rotor-Stator Interaction," International Journal of Turbo and Jet-Engines, Vol. 20, No. 1, Jan. 2003, pp. 17–39.

[2] Rodriguez, B., Benoit, C., and Gardareing, P., "Unsteady Computations of the Flowfield around a Helicopter Rotor with Model Support," 43rd AIAA Aerospace Sciences Meeting, Reno, NV, AIAA Paper 2005-0466, Jan. 2005.

[3] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, Aug. 1990, pp. 1381–1388.

[4] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "An Improved Method of Spring Analogy for Dynamic Unstructured Fluid Meshes," AIAA Paper 1998-2070, 1998.

[5] Duvigneau, R., and Visonneau, M., "Shape Optimization of Incompressible and Turbulent Flows Using the Simplex Method," AIAA Paper 2001-2533, June 2001.

[6] Davis, G. A., and Bendiksen, O. O., "Unsteady Transonic Two-Dimensional Euler Solutions Using Finite Elements," *AIAA Journal*, Vol. 31, No. 6, June 1993, pp. 1050–1059.

[7] Cavallo, P. A., Hosangadi, A., Lee, R. A., and Dash, S. M., "Dynamic Unstructured Grid Methodology with Application to Aero/Propulsive Flowfields," AIAA Paper 1997-2310, 2002.

[8] Kim, K. S., and Cizmas, P. G. A., "Three-Dimensional Hybrid Mesh Generation for Turbomachinery Airfoils," *Journal of Propulsion and Power*, Vol. 18, No. 3, May–June 2002, pp. 536–543.

[9] Gargoloff, J. I., Cizmas, P. G. A., Strganac, T. W., and Beran, P. S., "Parallel Algorithm for Fully Nonlinear Aeroelastic Analysis," 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Newport, RI, AIAA Paper 2006-2073, May 2006.

[10] Gargoloff, J. I., "A Numerical Method for Fully Nonlinear Aeroelastic Analysis," Ph.D. Thesis, Texas A&M Univ., College Station, TX, May 2007.

[11] Goland, M., "The Flutter of a Uniform Cantilever Wing," *Journal of Applied Mechanics*, Vol. 12, No. 4, Dec. 1945, pp. A197–A208.