AIAA JOURNAL
Vol. 45, No. 12, December 2007

Mesh Deformation Strategy Optimized by the Adjoint Method on Unstructured Meshes

Zhi Yang* and Dimitri J. Mavriplis†
University of Wyoming, Laramie, Wyoming 82071

DOI: 10.2514/1.30592

An adjoint-based optimization procedure is proposed for improving the robustness and extending the range of linear-elasticity-based mesh deformation techniques. Using the values of the modulus of elasticity *E* defined in each mesh cell as the design variables, the procedure seeks to determine an optimum distribution of *E* throughout the mesh, to minimize a global objective function that reflects the skewness or lack of quality of the deformed mesh. The technique is applied to highly stretched mixed element meshes in two and three dimensions on complex geometries and is shown to be capable of recovering a valid mesh in a small number of optimization cycles for cases in which the nonoptimized linear-elasticity approach fails. However, the solution of the optimization problem remains relatively costly in terms of CPU time, compared with a nonoptimized mesh deformation calculation, making this technique best suited for precomputing improved *E* distributions before the simulation or for use as a plug-in module to be invoked in cases in which the nonoptimized procedure fails.

I. Introduction

UNSTRUCTURED mesh approaches have become well established for numerical flow simulations, due to the flexibility they afford for dealing with complex geometries. For problems involving moving boundaries, such as unsteady aeroelastics, control surface deflection problems, and shape optimization problems, robust mesh deformation techniques are necessary for maintaining a suitable discretization of the evolving computational domain. Several mesh deformation strategies, such as the tension spring analogy [1,2], the torsion spring analogy [3,4] and the linear-elasticity analogy [5,6], have been successfully demonstrated in the literature. These methods are designed to produce a valid mesh given a set of prescribed boundary displacements by recomputing new mesh point coordinates without the addition/deletion of new mesh points and without altering the connectivity or topology of the mesh. However, all these methods will eventually fail given large enough boundary displacements, because the original mesh topology cannot be retained indefinitely. On the other hand, resorting to mesh reconnection and refinement/derefinement schemes adds considerable complexity to any numerical simulation procedure, because this generally entails load-balancing operations, changing discretization stencils, and introduces nondifferentiable procedures into the simulation process, which presents difficulties when linearizations are required, as may be the case for sensitivity analysis within the context of error analysis or shape optimization problems. Thus, it is important to develop robust mesh deformation strategies that enable the retention of the mesh connectivity for a wide range of problems.

The various mesh deformation methods mentioned earlier contain a series of model parameters (i.e., spring stiffness and modulus of elasticity) that can be modified to increase the robustness of these schemes. For example, in the context of the linear spring analogy approach, the spring stiffnesses are often taken as inversely proportional to the length of the mesh edge raised to some power *p* [2]. For the linear-elasticity approach, the modulus of elasticity is often taken as inversely proportional to the cell volume [7–9]. This formulation of the linear-elasticity approach was found to be particularly robust, because by design, the linear-elasticity approach is capable of reproducing solid-body motion (translation and rotation) for stiff regions (high modulus of elasticity *E*). Thus, critical small-cell regions are displaced with little or no relative deformation or strain, and the most severe deformations are relegated to less critical regions of the mesh, which can sustain more strain. Alternate prescriptions such as taking the modulus of elasticity *E* inversely proportional to the distance from the moving wall have also been found to work well in various cases [9].

For cases that fail, further robustness improvement can be obtained by modifying the *E* distribution in regions in which poor mesh quality is detected. For example, one approach may consist of prescribing very large values of *E* at cells that become negative through the deformation process or, alternatively, updating the *E* values as inversely proportional to the deformed cell volumes while progressively deforming the mesh in a sequence of small steps. One of the drawbacks of this approach is that the mesh deformation equations become nonlinear, whereas the prescription of *E* as inversely proportional to cell volume remains somewhat arbitrary.

Alternatively, one may wonder about the best distribution of *E* that will result in the highest-quality deformed mesh or that will enable the retention of a valid mesh (positive volume cells) under the largest surface displacements. Once this optimal distribution is determined, the linear mesh deformation problem may be solved to obtain the new deformed mesh. The determination of the best *E* distribution corresponds to an optimization problem in which the values of *E* in each mesh cell constitute the design variables and the objective consists of some description of global mesh quality. Adjoint methods are well suited for these types of optimization problems, which involve a single objective function and a large number of design variables [6,10–12]. In this work, we propose an adjoint-based optimization procedure for determining an optimum or improved *E* distribution for the linear-elasticity mesh deformation approach, which results in a more robust mesh deformation strategy. One advantage of this approach is that it no longer relies on the arbitrary prescription of *E* as inversely proportional to the cell volume or the distance from the wall. Furthermore, the mesh deformation problem remains linear, and once the optimum *E* distribution has been determined, the efficiency of the previously developed linear mesh deformation solver is retained. On the other hand, the optimum *E* distribution is found to depend on the particular prescribed boundary displacement, and as will be shown, the solution of the optimization problem is generally nontrivial, thus reducing some of the perceived

Presented as Paper 557 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 8–11 January 2007; received 27 February 2007; revision received 22 July 2007; accepted for publication 12 August 2007.

*Postdoctoral Research Associate, Department of Mechanical Engineering; zyang@uwyo.edu. Member AIAA.

†Professor; mavriplis@uwyo.edu. Associate Fellow AIAA.

advantages of this approach over the previously mentioned nonlinear formulation. However, the resulting E distributions can be expected to result in a more robust mesh deformation process. Furthermore, these optimal E distributions may be precomputed for the most severe mesh deformation cases and then used subsequently for intermediate steps in the linear mesh motion-solution process, thus reducing the overall cost.

In this paper, we formulate an adjoint-based optimization problem for improving the robustness of linear-elasticity mesh deformation techniques and demonstrate this approach on various complex configurations. In the following section, we first outline the linear-elasticity mesh deformation strategy. We then formulate the optimization problem for improving the robustness of the mesh deformation strategy, including the formulation of the adjoint problem, a suitable objective, and techniques for carrying out the optimization. Solution of the optimization problem can be particularly challenging, given the large number of design variables (three-dimensional set corresponding to millions of mesh cells). Thus, considerable effort must be expended on devising a suitable convex objective function and using efficient optimization procedures. Finally, two two-dimensional and two three-dimensional test cases are presented to demonstrate the robustness and efficiency of the adjoint optimized mesh deformation method.

II. Mesh Deformation Strategy

The linear-elasticity approach was demonstrated by various authors [5–7] as a robust technique for computing dynamically deforming meshes. The motion of the computational mesh is assumed to obey the linear-elasticity equations, which can be written as

$$\nabla \cdot \sigma = -\mathbf{f} \tag{1}$$

where $\sigma = \mathbf{D}\epsilon$ ($\epsilon = \mathbf{A}\delta\mathbf{x}$ and $\mathbf{D}$ is the constitutive matrix, which is a function of modulus of elasticity E and the Poisson ratio ν), and $\mathbf{f}$ is the applied force. In three dimensions, the stresses σ, strains ϵ, displacements $\delta\mathbf{x}$, and the matrix $\mathbf{A}$ are given as

$$\sigma = \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{31}\}^T$$

$$\epsilon = \{\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12}, \epsilon_{23}, \epsilon_{31}\}^T, \qquad \delta\mathbf{x} = \{\delta\mathbf{x}_1, \delta\mathbf{x}_2, \delta\mathbf{x}_3\}^T$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^T$$

By introducing the shape functions $\mathbf{N}$, taken as linear functions over the mesh elements in our case, and applying a standard Galerkin method, we obtain

$$\mathbf{K}\,\delta\mathbf{x} = \mathbf{F} \tag{2}$$

where

$$\mathbf{K} = \int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\,\mathrm{d}S, \qquad \mathbf{F} = -\int_\Omega \mathbf{N}^T\mathbf{f}\,\mathrm{d}S, \qquad \mathbf{B} = \mathbf{A}\mathbf{N} \tag{3}$$

In the mesh deformation case, the boundary displacements are given so that the external force vector $\mathbf{F}$ is not required. Rather, the homogeneous problem $\mathbf{K}\delta\mathbf{x} = 0$ is solved, subject to Dirichlet conditions on the $\delta\mathbf{x}$ displacement vector. Hence, Eq. (2) can be rewritten as

$$\mathbf{K}\,\delta\mathbf{x} = \mathbf{F}(\mathbf{x}_b) \tag{4}$$

where $\mathbf{x}_b$ is the boundary displacement vector and $\mathbf{F}(\mathbf{x}_b)$ is a function of $\mathbf{x}_b$. Note that the stiffness matrix $\mathbf{K} = \mathbf{K}(\mathbf{E})$ is a function of the E values in each cell, as seen from Eq. (3). One advantage of the linear-elasticity approach is that regions of large E (modulus of elasticity) will be displaced as a solid body. Thus, an appropriate prescription of the distribution of E can be used to avoid severe mesh deformation in critical regions of the mesh. In previous work, we employed a distribution of E that is inversely proportional to the cell volume and/or to the distance from the deforming boundaries [7,8]. This turns out to be critical for avoiding invalid mesh cells in small-cell regions and in regions near the boundaries.

Although a distribution of the modulus E that is inversely proportional to the cell volume or to the distance from the deforming boundaries can minimize distortion for cells in critical near-body regions, negative volume cells may appear in unexpected regions of the domain for particular prescribed boundary displacements. An example is given in Sec. V for the spoiler deflection problem discussed subsequently. Thus, a modulus distribution that not only prevents the cells near the body from becoming negative but also keeps other cells valid as the mesh deforms is crucial to take full advantage of the linear-elasticity analogy. In the next section, we discuss the implementation of the adjoint approach for producing a more optimal distribution of E, resulting in a more robust mesh deformation strategy.

III. Adjoint-Based Optimization Procedure

To formulate the optimization problem for improving the robustness of linear-elasticity-based mesh deformation methods, we must construct a suitable objective to be minimized, determine an efficient approach for computing the sensitivities of this gradient with respect to the design variables or model parameters, and make use of an efficient and practical optimization technique for obtaining an improved set of model parameters.

As opposed to shape optimization problems, in which the design variables are used to define the shape of the geometry, the set of design variables for the mesh optimization problem constitutes a two- or three-dimensional set that involves the value of the modulus of elasticity in each mesh cell. Thus, the number of design variables will typically exceed several million, making the use of an adjoint-based approach imperative for gradient-based optimization methods. Additionally, a single objective function is desirable for the adjoint method, which accurately reflects the overall quality of the mesh and increases rapidly when negative volume cells are created. Although the formulation of the adjoint problem is relatively straightforward, the success of the mesh optimization problem in terms of robustness and efficiency depends critically on the choice of a suitable objective function, a good initial distribution of model parameters, and a robust optimization strategy.

A. Objective Function

The mesh deformation optimization problem is defined as minimizing an objective function $\mathbf{L}$ by varying the modulus of elasticity E_i in each cell i, thus treating the E_i as the design variables in an optimization framework. The objective function is designed in such a way that minimizing the objective function $\mathbf{L}$ will produce valid cells with minimal distortion in the deforming mesh. The objective function $\mathbf{L}$ is defined as a function of the cell volumes $\mathbf{V}$:

$$\mathbf{L} = \mathbf{L}(\mathbf{V}) \tag{5}$$

The construction of a suitable objective function for mesh deformation optimization problems represents a nontrivial task. The objective should be representative of global mesh quality and should increase rapidly for vanishing cell volumes. Additionally, the objective should be a convex function of the design variables to facilitate the optimization procedure. For two-dimensional problems, the objective function given by

$$\mathbf{L} = \sum_{i=1}^{N} L_i \quad \text{and} \quad L_i = \begin{cases} \frac{a_-}{b_-^n}(\xi_i - 1)^n & \text{if } \xi_i \le 0 \\ \frac{a_+}{b_+^n}(\xi_i - 1)^n & \text{if } \xi_i \ge 0 \end{cases} \tag{6}$$

was found to be suitable for a wide range of problems, and the form given by

$$\mathbf{L} = \sum_{i=1}^{N} L_i \quad \text{and} \quad L_i = e^{a(\xi_i - 1)^n} - 1 \tag{7}$$

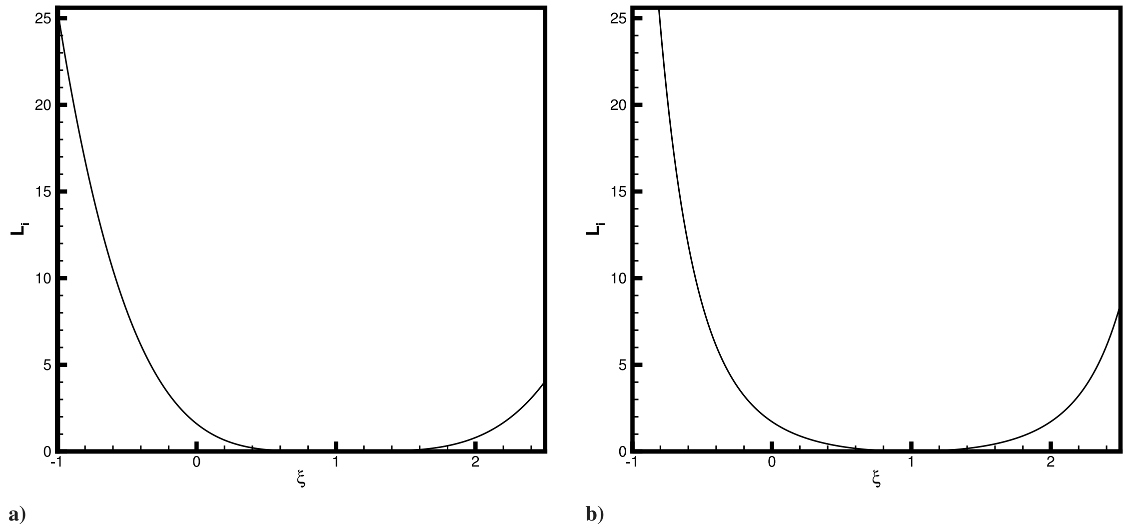


Fig. 1 Illustration of the variation of the global objective function a) defined in Eq. (6) with respect to the volume ratio ξ at a single cell using the parameter values $a_- = a_+ = 0.1$, $b_- = b_+ = 0.5$, $n = 4$ and b) defined in Eq. (7) with respect to the volume ratio ξ at a single cell using the parameter values $a = 5$ and $n = 2$.

was used for three-dimensional problems. In the preceding equations, N is the total number of the design variables; a_-, b_-, a_+, b_+, a and n are the control coefficients and n is set to an even positive number; and the parameter ξ_i in each cell is defined as $\xi_i = V_i/V_{0i}$, where V_{0i} and V_i are the volume of the ith cell before and after the deformation, respectively. Figure 1 provides an illustration of the objective functions described in Eqs. (6) and (7). When ξ_i equals unity, which corresponds to $V_i = V_{0i}$, L_i vanishes. When ξ_i approaches zero, which means the volume V_i approaches zero, L_i become very large. The objective function is designed in such a way that the cell volume V_i is pushed toward the original cell volume V_{0i} when the objective function is minimized by varying the design variables, thus avoiding collapsed cells as the mesh deforms. It should be noted that this optimization problem need not be completely convergent (i.e., the objective will not be reduced to zero) because cells usually cannot retain the same volume as the mesh is deformed.

B. Adjoint Method

For a gradient-based optimization strategy, we need to compute the sensitivity of the objective function with respect to all the design variables or model parameters, which in this case are the moduli of elasticity for each mesh cell. Because we will generally have millions of design variables, the adjoint method is used to compute these sensitivities. We consider the volume V of a deformed-mesh cell as a function of mesh point displacements $\delta\mathbf{x}$ and the initial coordinates of the mesh points $\mathbf{x}_0$:

$$\mathbf{V} = \mathbf{V}(\mathbf{x}_0 + \delta\mathbf{x}) \tag{8}$$

According to Eq. (4), the mesh deformations $\delta\mathbf{x}$ are a function of the prescribed boundary deformations $\mathbf{x}_b$, the coordinates of the initial mesh configuration $\mathbf{x}_0$ (which appears through the shape functions M), and the moduli of elasticity E_i, because different distributions of $\mathbf{E}$ will lead to a different stiffness matrix $\mathbf{K}(\mathbf{E})$ and thus different mesh displacements. Therefore, the mesh displacements can be written in the following functional form:

$$\delta\mathbf{x} = \delta\mathbf{x}(\mathbf{E}, \delta\mathbf{x}_b, \mathbf{x}_0) \tag{9}$$

which can be computed by solving the mesh motion equation:

$$\mathbf{K}\,(\mathbf{E})\delta\mathbf{x} = \mathbf{F}(\delta\mathbf{x}_b) \tag{10}$$

The modulus coefficients $\mathbf{E}$ are the design variables and Eq. (5) can therefore be rewritten as

$$\mathbf{L} = \mathbf{L}(\mathbf{E}) \tag{11}$$

A variation in the design variables $\delta\mathbf{E}$ produces a corresponding variation of the objective $\delta\mathbf{L}$ as

$$\delta\mathbf{L} = \frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{E}}\delta\mathbf{E} \tag{12}$$

where the sensitivity derivative may be calculated using Eqs. (5) and (8) as

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}E_i} = \frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{V}}\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\delta\mathbf{x})}\frac{\partial(\delta\mathbf{x})}{\partial E_i} \tag{13}$$

By differentiating Eq. (10) with respect to the ith design variable, we obtain

$$\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}(\delta\mathbf{x}) + \mathbf{K}\frac{\partial(\delta\mathbf{x})}{\partial E_i} = 0 \quad \text{or} \quad \frac{\partial(\delta\mathbf{x})}{\partial E_i} = -\mathbf{K}^{-1}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}(\delta\mathbf{x}) \tag{14}$$

where E_i is ith component of the vector $\mathbf{E}$ associated with cell i. Substituting Eq. (14) into Eq. (13), we get

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}E_i} = \underbrace{-\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{V}}\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\delta\mathbf{x})}\mathbf{K}^{-1}}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}(\delta\mathbf{x}) \tag{15}$$

Defining

$$\Lambda^T = -\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{V}}\frac{\mathrm{d}\mathbf{V}}{d(\delta\mathbf{x})}\mathbf{K}^{-1}$$

we obtain the adjoint equation, which must be solved as

$$\mathbf{K}^T\Lambda = -\left(\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{V}}\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\delta\mathbf{x})}\right)^T \tag{16}$$

In Eq. (16), $\mathrm{d}\mathbf{L}/\mathrm{d}\mathbf{V}$ can be easily obtained by differentiating Eq. (6) or Eq. (7). Additionally, using the equality

$$\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\delta\mathbf{x})} = \frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\mathbf{x}_0 + \delta\mathbf{x})} = \frac{\mathrm{d}\mathbf{V}}{\mathrm{d}(\mathbf{x})} \tag{17}$$

the last term in Eq. (16) corresponds to the derivative of the deformed-mesh cell volume with respect to the mesh coordinates, using the deformed-mesh coordinates. This can easily be evaluated from the formulas used to compute the cell volumes. For example, the area of a two-dimensional triangle can be computed as

$$V = \frac{1}{2}(c_1 b_3 - c_3 b_1) \tag{18}$$

where $b_1 = y_2 - y_3$, $b_2 = y_3 - y_1$, $b_3 = y_1 - y_2$, $c_1 = x_3 - x_2$, $c_2 = x_1 - x_3$, and $c_3 = x_2 - x_1$ using a cyclical indexing system $i = 1, 2, 3$ for the three vertices of the triangle. The required derivatives are then given as

$$\frac{\mathrm{d}V}{\mathrm{d}(\delta \mathbf{x})} = \frac{1}{2}\{b_1, c_1, b_2, c_2, b_3, c_3\}^T \tag{19}$$

For a three-dimensional tetrahedron, pyramid, prism, or hexahedron, the volume may be computed as

$$V = \sum_{i=1}^{q} w_i \det \mathbf{J_i} \tag{20}$$

where w_i and $\mathbf{J_i}$ are the ith weight function and Jacobian matrix, respectively, and q is the number of quadrature points. Then the required derivatives are given as

$$\frac{\mathrm{d}V}{\mathrm{d}(\delta \mathbf{x})} = \sum_{i=1}^{q} w_i \frac{\mathrm{d}(\det \mathbf{J_i})}{\mathrm{d}\mathbf{x}} \tag{21}$$

Note that the $\mathrm{d}V/\mathrm{d}(\delta \mathbf{x})$ term is nominally a sparse matrix of size $n\,\mathrm{cells} \times 3 \cdot n\,\mathrm{points}$, where n cells represents the number of cells in the mesh, and n points denotes the number of mesh vertices, although only the locations corresponding to the vertices that constitute the corner points of a given element are nonzero. These terms can be assembled on the fly, and the resulting product $(\mathrm{d}\mathbf{L}/\mathrm{d}V)[\mathrm{d}V/\mathrm{d}(\delta \mathbf{x})]$ consists of a vector of length $3 \cdot n$ points, as required to make Eq. (16) dimensionally correct. Once the adjoint Eq. (16) has been solved, the final sensitivity derivatives for all design variables can be obtained by performing the matrix-vector multiplication:

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}E_i} = \Lambda^T \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}(\delta \mathbf{x}) \tag{22}$$

In practice, the calculation of the sensitivities, which must be performed at each design iteration, requires the solution of one mesh deformation problem (10) and one mesh adjoint problem (16). The solution of the mesh motion problem yields the values of the mesh point displacements $\delta \mathbf{x}$, which are required both in Eq. (22) and for determining the right-hand side of the adjoint problem (16). Next, the adjoint problem is solved, after which all terms are available for evaluating the right-hand-side of Eq. (22). The $\mathrm{d}\mathbf{K}/\mathrm{d}E_i$ term in this equation is easily obtained by differentiating the functional form of the stiffness matrix. Although this term represents a sparse matrix of dimension $3 \cdot n\,\mathrm{points} \times 3 \cdot n\,\mathrm{points} \times n\,\mathrm{cells}$, only the terms corresponding to the corner points of a given cell are nonzero, and the matrix need not be evaluated explicitly. Rather, the complete sequence of matrix-vector products is assembled on an element basis, gathering the values from the corner points of each element, and scattering/accumulating the results back to these points. The final result consists of a vector of length n cells, as required to make the equation dimensionally consistent.

Rewriting Eq. (22) as

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}E_i} = \mathbf{\Lambda}^T \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}(\delta \mathbf{x}) = \mathbf{\Lambda}^T \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}E_i}\mathbf{K}^{-1}\mathbf{F}(\delta \mathbf{x}_b) \tag{23}$$

clearly shows that the sensitivities required for the optimization problem depend on the particular prescribed boundary displacement $\delta \mathbf{x}_b$. Thus, different boundary displacements can be expected to result in different optimum E distributions. Furthermore, although we retain the linear form of the mesh motion equations, and the mesh adjoint equation is by definition a linear problem, the optimization problem itself [i.e., $\mathbf{L} = \mathbf{L}(\mathbf{E})$] is highly nonlinear.

C. Optimization Methods

The optimization procedure consists of finding the distribution of $\mathbf{E}$ that minimizes the objective function $\mathbf{L}$. At the kth iteration, the new distribution $\mathbf{E}_{k+1}$ is computed as

$$\mathbf{E}_{k+1} = \mathbf{E}_k + \lambda_k \mathbf{p}_k \quad \text{or} \quad \Delta \mathbf{E}_k = \lambda_k \mathbf{p}_k \tag{24}$$

where λ_k and $\mathbf{p}_k$ are the step length and the search direction, respectively. The steepest-descent method represents a simple optimization algorithm [13] that has been used successfully for aerodynamic shape optimization [10]. In this approach, the search directions $\mathbf{p}_k$ are taken as $-(\mathrm{d}\mathbf{L}/\mathrm{d}\mathbf{E})_k$. Once the objective function sensitivities $(\mathrm{d}\mathbf{L}/\mathrm{d}\mathbf{E})_k$ have been computed using the method described earlier, an increment in the design variables is prescribed as

$$\Delta \mathbf{E}_k = -\lambda \left(\frac{\mathrm{d}\tilde{\mathbf{L}}}{\mathrm{d}\mathbf{E}}\right)_k \tag{25}$$

where λ is a small time step determined empirically, and $(\mathrm{d}\tilde{\mathbf{L}}/\mathrm{d}\mathbf{E})_k$ is the smoothed gradient $(\mathrm{d}\mathbf{L}/\mathrm{d}\mathbf{E})_k$. The advantage of the steepest-descent method is its simplicity and suitability for optimization problems with a large number of design variables. Howeverb the convergence speed of the steepest-descent method was found to be very slow for the mesh optimization problems discussed in this paper. A better choice of the step length λ or the search directions $\mathbf{p}_k$ can be used to help accelerate the convergence of the optimization problem. To this end, a line-search method [13,14] was implemented to compute an appropriate step size λ within the steepest-descent approach. Defining a function

$$\phi(\lambda) = \mathbf{L}(\mathbf{E}_k + \lambda \mathbf{p}_k) \tag{26}$$

we attempt to find a step length λ that minimizes the function ϕ or satisfies the Wolfe condition (sufficient decrease condition) [13]:

$$\mathbf{L}\,(\mathbf{E}_k + \lambda_k \mathbf{p}_k) \leq \mathbf{L}(\mathbf{E}_k) + c_1 \lambda_k \left(\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{E}}\right)_k^T \mathbf{p}_k \tag{27}$$

where c_1 is a constant and $0 < c_1 < 1$. Considering Eq. (26), the sufficient decrease condition (27) can be rewritten as

$$\phi(\lambda_k) \leq \phi(0) + c_1 \lambda_k \phi'(0) \tag{28}$$

where

$$\phi'(0) = \left(\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}\mathbf{E}}\right)_k^T \mathbf{p}_k$$

If an initial guess λ_0 satisfies Eq. (28), then we set $\lambda_k = \lambda_0$. Otherwise, a quadratic function is constructed using the values of $\phi(0)$, $\phi'(0)$, and $\phi(\lambda_0)$:

$$\phi_q(\lambda) = a_q \lambda^2 + b_q \lambda + c_q \tag{29}$$

where the coefficients a_q, b_q, and c_q are determined by $\phi_q(0) = \phi(0)$, $\phi'_q(0) = \phi'(0)$, and $\phi_q(\lambda_0) = \phi(\lambda_0)$. A new step length λ_q can be found in $[0, \lambda_0]$ by minimizing the quadratic function (29):

$$\lambda_q = -\frac{\phi'(0)\lambda_0^2}{2[\phi(\lambda_0) - \phi(0) - \phi'(0)\lambda_0]} \tag{30}$$

If λ_q satisfies Eq. (28), then let $\lambda_k = \lambda_q$. Otherwise, a cubic function is constructed using the values of $\phi(0)$, $\phi'(0)$, $\phi(\lambda_0)$, and $\phi(\lambda_q)$:

$$\phi_c(\lambda) = a_c \lambda^3 + b_c \lambda^2 + c_c \lambda + d_c \tag{31}$$

where the coefficients a_c, b_c, c_c, and d_c are determined by $\phi_c(0) = \phi(0)$, $\phi'_c(0) = \phi'(0)$, $\phi_c(\lambda_0) = \phi(\lambda_0)$, and $\phi_c(\lambda_q) = \phi(\lambda_q)$. By minimizing Eq. (31), a new step length λ_c in $[0, \lambda_q]$ is given by

$$\lambda_c = \frac{-b_c + \sqrt{b_c^2 - 3a_c \phi'(0)}}{3a_c} \tag{32}$$

The algorithm described earlier is the so-called cubic interpolation algorithm and is used in this paper to compute the step size λ.

Additional details for the cubic interpolation algorithm can be found in .

Convergence of the optimization problem can also be improved using more sophisticated algorithms for computing improved search directions $\mathbf{p}_k$. In the BFGS algorithm [13,14], the search directions $\mathbf{p}_k$ in Eq. (24) are computed as

$$\mathbf{p}_k = -\mathbf{B}_k\left(\frac{d\mathbf{L}}{d\mathbf{E}}\right)_k \tag{33}$$

where $\mathbf{B}_k$ is the inverse Hessian matrix and the size of $\mathbf{B}_k$ is $N \times N$, where N is the number of design variables, which is typically on the order of several million. Storage of the $N \times N$ inverse Hessian matrix $\mathbf{B}_k$ is thus not practical for such large optimization problems. Therefore, we have also investigated the use of the limited-memory BFGS method (LBFGS) presented by Liu and Nocedal [15], because this approach does not require the storage of the Hessian matrix. Instead, an approximate inverse Hessian matrix $\mathbf{B}$ is computed at every iteration using m vector pairs $\{s_i, y_i\}$ [15]:

$$\begin{aligned}\mathbf{B}_k &= \left(\mathbf{Y}_{k-1}^T \cdots \mathbf{Y}_{k-m}^T\right)\mathbf{B}_k^0(\mathbf{Y}_{k-m} \cdots \mathbf{Y}_{k-1}) \\ &+\rho_{k-m}\left(\mathbf{Y}_{k-1}^T \cdots \mathbf{Y}_{k-m+1}^T\right)\mathbf{s}_{k-m}\mathbf{s}_{k-m}^T(\mathbf{Y}_{k-m+1} \cdots \mathbf{Y}_{k-1}) \\ &\vdots \\ &+\rho_{k-1}\mathbf{s}_{k-1}\mathbf{s}_{k-1}^T\end{aligned} \tag{34}$$

where $\mathbf{s}_k = \mathbf{E}_k - \mathbf{E}_{k-1}$, $\mathbf{y}_k = (d\mathbf{L}/d\mathbf{E})_k - (d\mathbf{L}/d\mathbf{E})_{k-1}$, $\rho_k = 1/\mathbf{y}_k^T\mathbf{s}_k$, and $\mathbf{Y}_k = \mathbf{I} - \rho_k\mathbf{y}_k\mathbf{s}_k^T$. The initial matrix $\mathbf{B}_k^0$ is a diagonal matrix and

$$\mathbf{B}_k^0 = \frac{\mathbf{y}_k^T\mathbf{s}_k}{\|\mathbf{y}_k\|^2}\mathbf{I} \tag{35}$$

By storing the m previous values of the vectors $\{s_i, y_i\}$, the matrix $\mathbf{B}_k$ can be computed at each iteration implicitly. According to Liu and Nocedal [15], m is taken in the range of $3 \sim 25$. In our computations, the value $m = 5$ is used. The search directions $\mathbf{p}_k = -\mathbf{B}_k(d\mathbf{L}/d\mathbf{E})_k$ are then computed by a sequence of inner products s_i, y_i, and $(d\mathbf{L}/d\mathbf{E})_k$. Further details on the LBGFS algorithm and the computation of $\mathbf{B}_k(d\mathbf{L}/d\mathbf{E})_k$ are given by Nocedal [16].

IV. Mesh Motion and Adjoint Solution Strategies

As mentioned previously, each design iteration involves the solution of one mesh motion problem (10) and one mesh adjoint problem (22). As with any optimization problem, two techniques are required to minimize the overall cost of the optimization. On the one hand, the use of sophisticated optimization algorithms such as the LBFGS method described earlier [13] are used to reduce the overall number of design cycles. On the other hand, the cost of each design cycle can be reduced by employing efficient solvers for the mesh motion and adjoint problems.

In previous work, we demonstrated the use of agglomeration multigrid methods and line-implicit preconditioning, originally developed for accelerating the solution of the discretized flow equations [17,18], for accelerating the solution of the mesh motion equations using either the spring analogy or the linear-elasticity analogy [8,9]. In the current work, we use these techniques for efficiently solving the mesh motion equations, as well as the mesh adjoint equations.

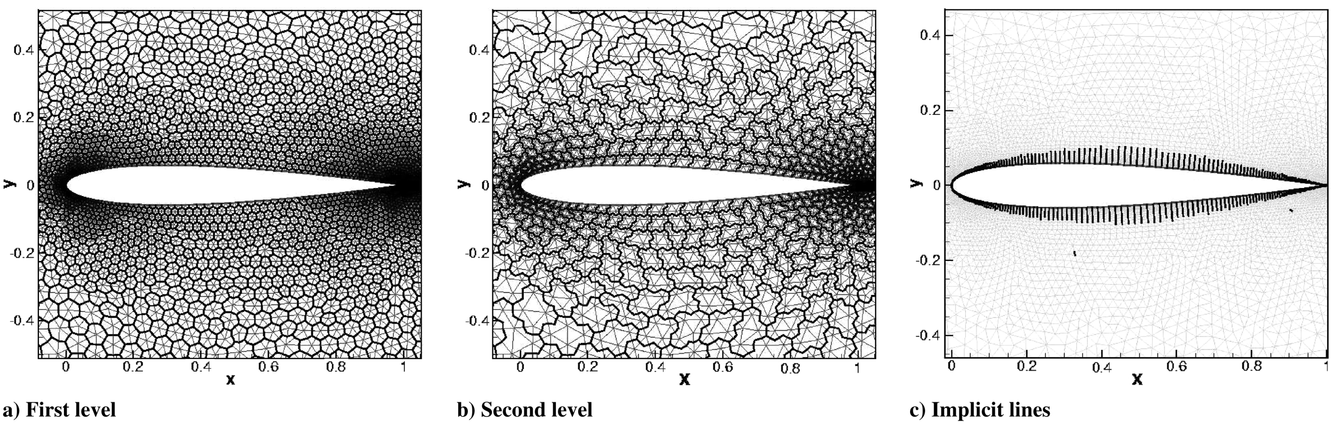
The idea of a multigrid algorithm is to accelerate the solution on a fine grid by iteratively computing corrections to the fine-grid problem on coarser-grid levels, for which the cost of the iterations are lower and the global error components are more easily reduced. For unstructured meshes, coarse-multigrid levels can be constructed by grouping together or agglomerating fine-level control volumes into a smaller set of coarser-level control volumes, as shown in Figs. 2a and 2b. Figure 2a shows the initial fine-grid and the dual control volumes that are constructed around each mesh point, and Fig. 2b shows the second-level coarse grid, in which each coarse-level control volume is a combination of several fine-level control volumes.

For high-Reynolds-number viscous flows, when highly stretched meshes are required to capture the thin boundary-layer regions near the wall, the effectiveness of the multigrid approach degrades due to the anisotropic stiffness induced by the grid stretching. To relax this stiffness, an implicit line-solution technique was introduced in [18]. The lines are constructed along the strong-coupling direction in the mesh, as shown by the example depicted in Fig. 2c for an unstructured mesh about a NACA0012 airfoil. In these regions, a block tridiagonal algorithm is used to solve all quantities along each line implicitly, thus replacing the simple explicit approach on all grid levels of the multigrid scheme.

A linear variant of the line-implicit agglomeration multigrid scheme is used to solve the mesh motion and adjoint problems, both of which are linear problems. Although in theory, a duality-preserving implementation of the multigrid algorithm can be used to guarantee similar convergence rates between the mesh motion and adjoint problems [11,12,19], in practice, the same multigrid solver is used on both problems, and similar convergence rates are generally observed for both problems, achieving suitable convergence in 50 to 100 multigrid cycles.

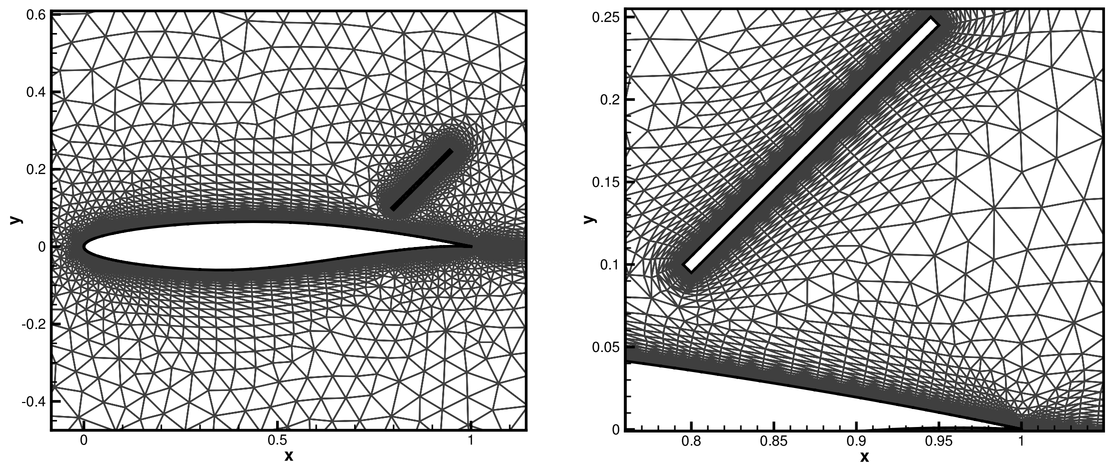
V. Results and Discussion

Our first test case consists of a two-dimensional airfoil-spoiler configuration, which is a simplified model of a three-dimensional wing-body-spoiler configuration. The unstructured mesh is a viscous mesh with high stretching near the body, which includes approximately 8700 vertices and 17,000 triangles, as shown in Fig. 3. Figure 4a shows the deformed mesh generated by the linear-elasticity analogy, with a modulus of elasticity prescribed as inversely proportional to the cell volume, as the spoiler rotates



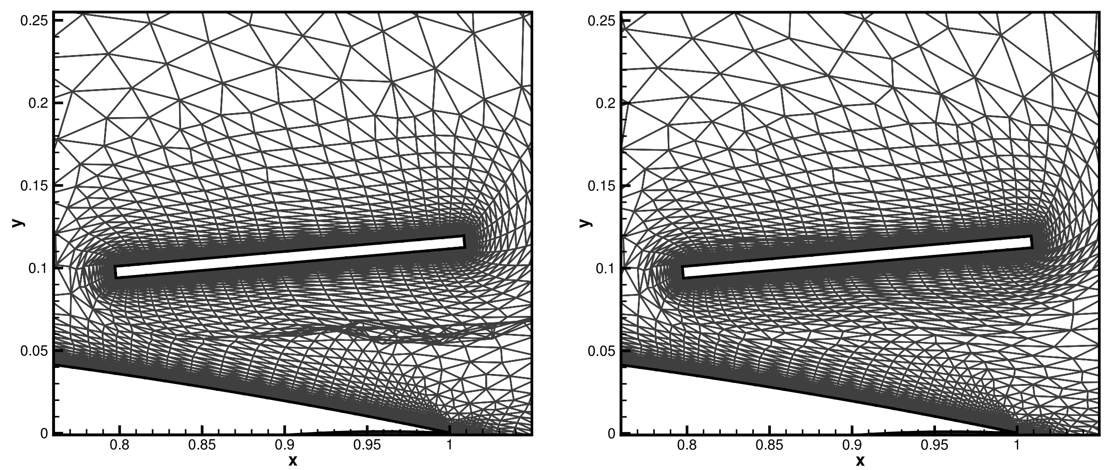
a) First level   b) Second level   c) Implicit lines

Fig. 2 Illustration of a) fine mesh and dual control volume structure used for coarse-level agglomeration multigrid construction, b) first coarse-level agglomerated grid, and c) line structures used in boundary-layer regions for the implicit line solver.



a) b)

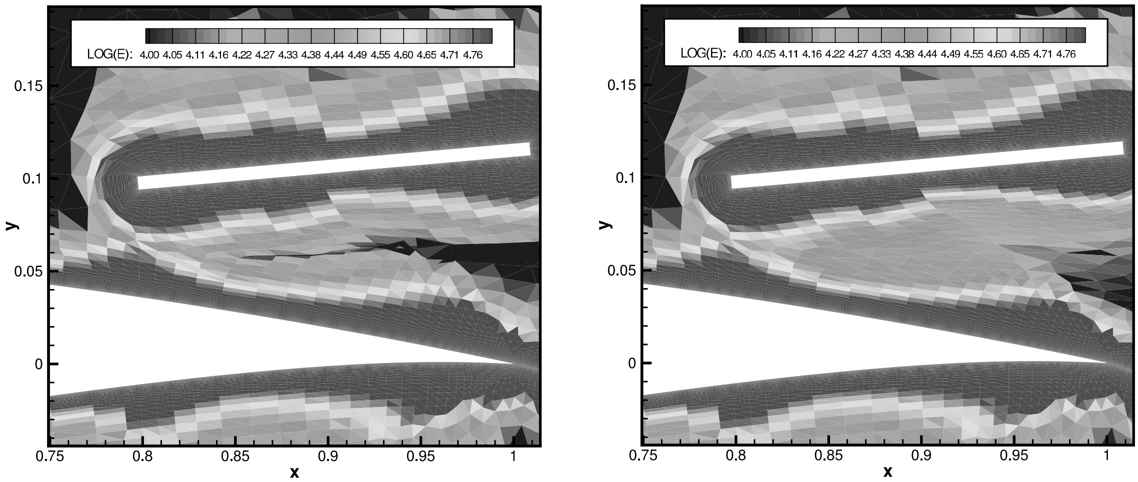
Fig. 3 Illustration of the initial two-dimensional unstructured mesh for the spoiler deflection problem with high stretching in the boundary-layer regions.



a) b)

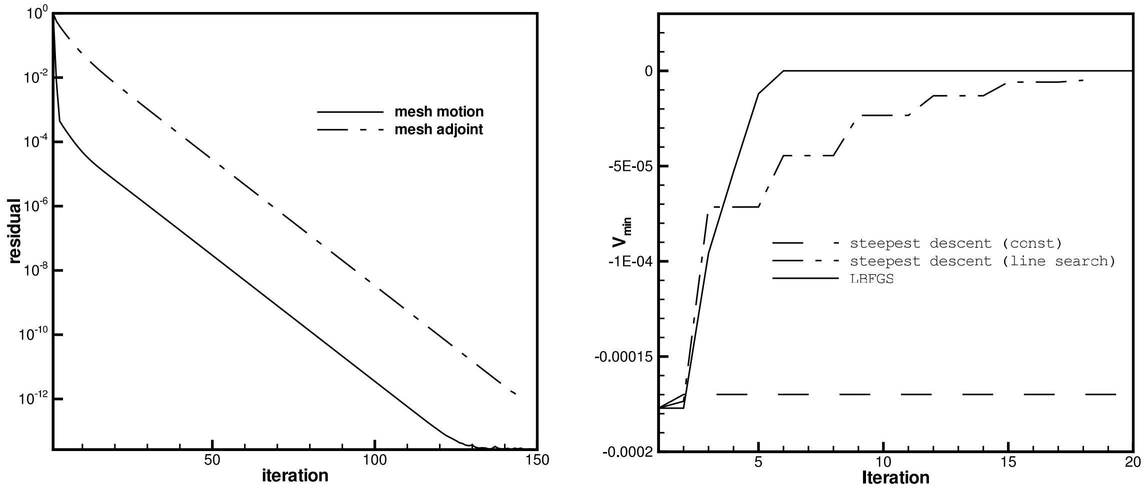
Fig. 4 Deformed mesh for spoiler deflection problem using a) prescribed *E* distribution illustrating the region of invalid mesh cells and b) optimized *E* distribution illustrating the recovery of valid mesh cells throughout the domain.

downward 40 deg. A region of invalid cells is clearly visible in the region between the spoiler and the airfoil, in which the original cell sizes are not particularly small. The original modulus distribution (inversely proportional to the cell volume or the distance from the wall) is inadequate for preventing the volume of these cells from becoming negative. The objective function used in this case is that given by Eq. (6). The coefficients are set as $a_- = a_+ = 0.1$, $b_- = b_+ = 0.5$, and $n = 4$. The optimization procedure acts by



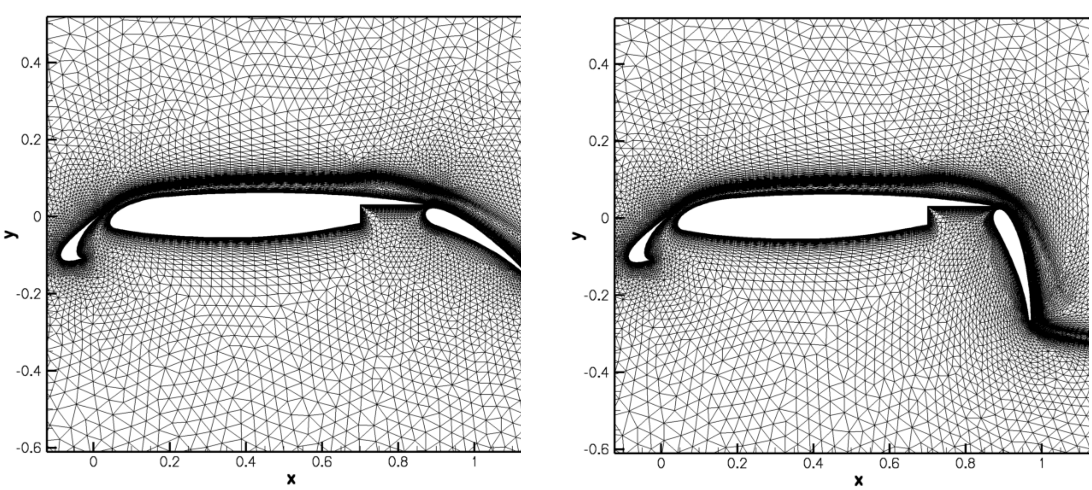
a) b)

Fig. 5 Modulus *E* distribution: a) prescribed in the deformed spoiler mesh and b) optimized in the deformed spoiler mesh.



a) b)

Fig. 6 Convergence history of the a) line-implicit multigrid solver for mesh deformation and mesh adjoint equations for the two-dimensional spoiler deflection case and b) various optimization strategies for two-dimensional spoiler deflection case as measured by the number of optimization cycles required to recover mesh with fully positive volume cells.



a) b)

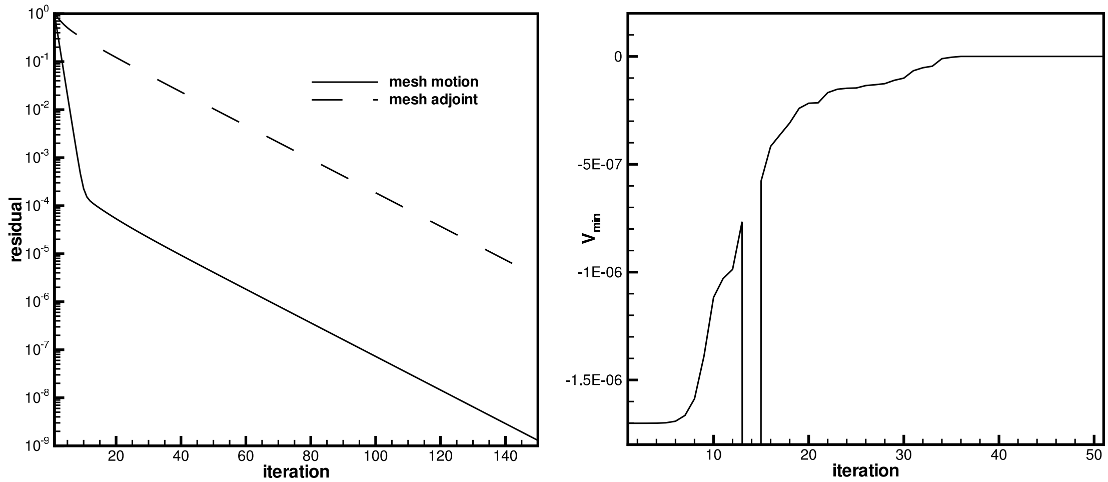
Fig. 7 Illustration of a two-dimensional unstructured mesh for a three-element airfoil with high stretching in boundary-layer regions for a) initial and b) deformed cases.

identifying the influence of these areas on the global objective and increasing the critical local modulus E variables accordingly. Figure 4b shows the mesh generated using the optimal distribution of the modulus E, as determined by the optimization procedure, illustrating the elimination of all negative volume cells. Figures 5a and 5b depict the initial and final modulus E distributions, demonstrating how the optimization process results in increased E values in the region of collapsing mesh cells between the spoiler and the airfoil, thus eliminating the invalid cells produced with the original prescribed E distribution.

Figure 6a shows the convergence of the line-implicit multigrid algorithm for the mesh motion equations and the mesh adjoint equations, respectively. Both problems converge at similar rates, and the residuals are reduced by approximately 8 to 10 orders of magnitude in 100 multigrid cycles for both problems in this case. Figure 6b compares the performance of the various optimization algorithms in recovering a valid grid, by plotting the minimum cell volume versus the number of design cycles. The LBFGS method produces a valid grid (recovering all negative cells) in six design iterations, whereas the steepest-descent method with the line-search algorithm has not completely eliminated all negative cells after 18 iterations. The steepest-descent method with constant λ produces an almost flat curve, indicating that this approach is impractical, because a very large number of design iterations are needed to eliminate all invalid cells. In these plots, a design iteration is defined as a mesh motion and adjoint solution, because the time spent in the optimization algorithm is trivial compared with the time required for the mesh motion and adjoint solutions.

The second test case consists of a three-element airfoil configuration, which includes 34,943 grid points and 69,302 triangles. The unstructured mesh is a viscous mesh with high stretching near the body, as shown in Fig. 7a. The prescribed motion is the rotation of the rear flap and the other elements are held fixed. The linear-elasticity analogy, with a modulus of elasticity prescribed as inversely proportional to the cell volume, can generate a valid mesh up to a rotation angle of 28 deg. In [20], the use of a large Poisson ratio (tending toward infinity) is shown to result in a more robust approach for linear-elasticity mesh motion problems. Using this modification (a Poisson ratio of 0.3 was used for all other calculations), the maximum rotation angle achieved could be increased to 30 deg.‡ On the other hand, when optimization of the E distribution is invoked, a valid mesh can be maintained up to a 42-deg flap rotation, as shown in Fig. 7b. If the rotation angle is more than 42 deg, the object function cannot drive the optimization process to eliminate all the negative volume cells, even as the objective is

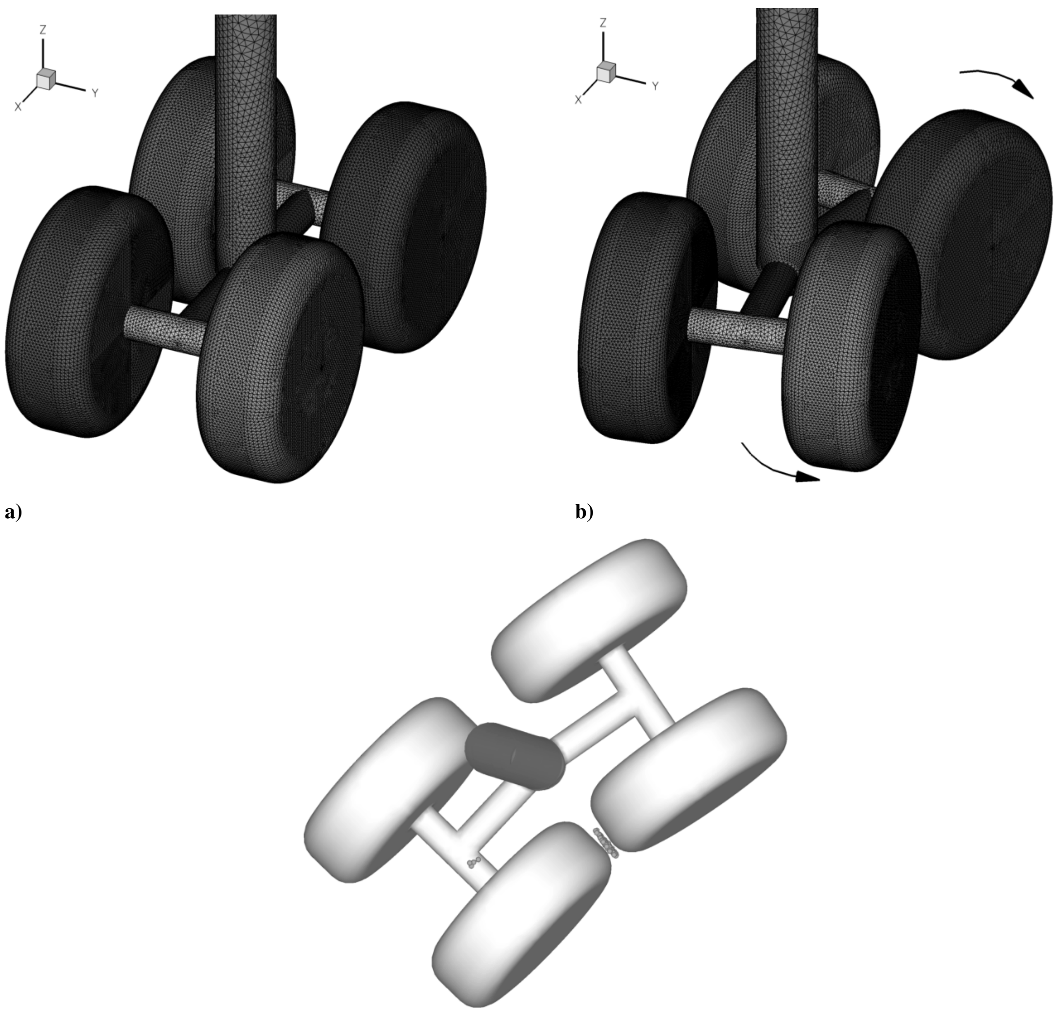
‡Private communication with R. P. Dwight, January 2007.



a) b)

Fig. 8 Convergence history of the a) line-implicit multigrid solver for mesh deformation and mesh adjoint equations for two-dimensional three-element airfoil case and b) LBFGS optimization algorithm for two-dimensional three-element airfoil case as measured by the number of optimization cycles required to recover mesh with fully positive volume cells.

reduced and the optimization problem converges. The effect of using a large Poisson ratio in conjunction with an optimized E distribution was not found to provide any additional robustness. Figure 8a shows the convergence of the multigrid algorithm for the mesh motion equations and the mesh adjoint equations, in which the residual for the mesh motion problem is reduced by nine orders of magnitude and the residual for the mesh adjoint problem is reduced by about six orders of magnitude in approximately 150 line-implicit multigrid



a) b) c)

Fig. 9 Illustration of a) initial and b) deformed mixed element mesh (containing tetrahedral, prismatic and pyramidal cells) on landing-gear configuration and c) location of negative cells for deformed mesh using original prescribed modulus E distribution; the number of points is 600,000 an the number of cells is 1.4 million.

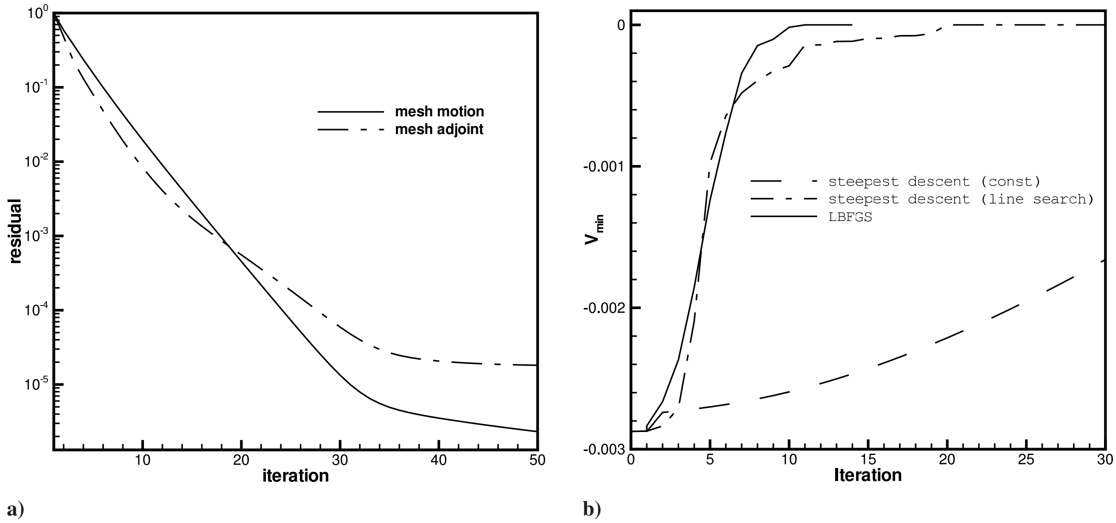


Fig. 10 Convergence history of the a) line-implicit multigrid solver for mesh deformation and mesh adjoint equations for three-dimensional landing-gear case and b) various optimization strategies as measured by the number of optimization cycles required to recover mesh with fully positive volume cells.

cycles. In Fig. 8b, the LBFGS optimization algorithm is seen to result in a valid mesh in 35 design iterations, and the other optimization methods are not used for this case. It should be noted in Fig. 8b that large negative volumes appear and disappear in the initial phases of the optimization, due to the fact that the variations in E driven by the minimization of the objective function do not necessarily guarantee monotonic increases in the small-cell volumes throughout the optimization process.

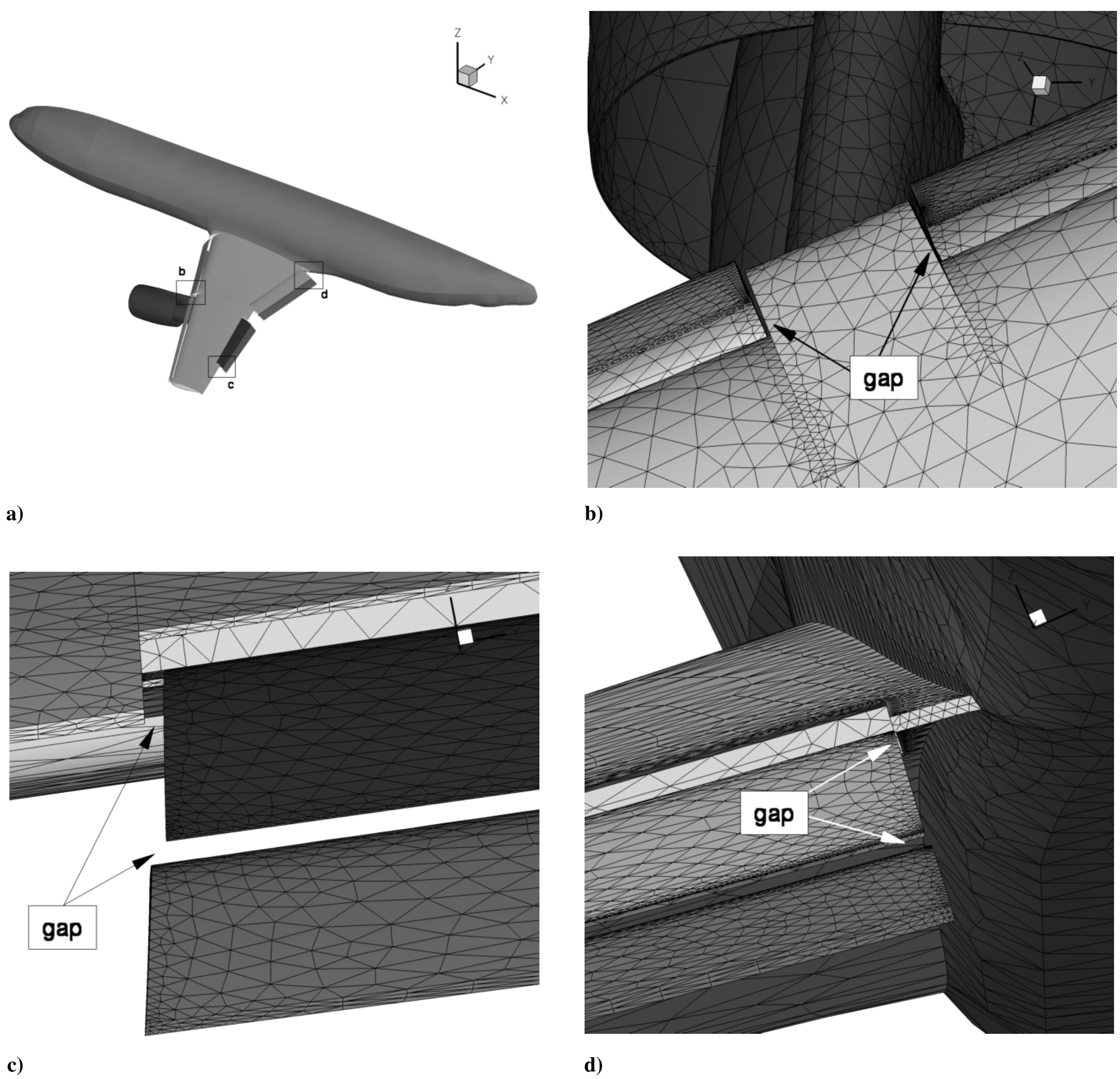


Fig. 11 Illustration of surface mesh on complex wing-body-slat-flap-nacelle-pylon geometry and location of critical gap regions. Hybrid mesh contains 3 million points and 11 million cells (tetrahedra, prisms, and pyramids).

The third test case consists of a three-dimensional landing-gear configuration, which includes approximately 600,000 grid points and 1.4 million cells. The unstructured grid consists of a viscous hybrid mesh, which contains mixtures of tetrahedra, pyramids, and prisms. Figure 9a shows the surface mesh of the landing-gear configuration. A twisting motion, as depicted in Fig. 9b, is prescribed on the geometry to test the mesh deformation procedure. For a twisting angle of 12 deg, the deformed mesh computed by the linear-elasticity approach, with a modulus of elasticity E prescribed as inversely proportional to the cell volume, results in 121 negative volume cells in the region between the two converging wheel geometry boundaries, as depicted in Fig. 9c.

For the optimization problem, the objective function used in this case is given by Eq. (7). The coefficients are set as $a = 5$ and $n = 2$. Figure 10a shows the convergence of the multigrid algorithm for the mesh motion equations and the mesh adjoint equations, in which the residuals for both problems are reduced by five orders of magnitude in approximately 50 line-implicit multigrid cycles. In Fig. 10b, the LBFGS optimization algorithm is seen to result in a valid mesh in 10 design iterations, whereas the steepest-descent method with the line-search algorithm eliminates all negative cells in 20 iterations. The steepest-descent method with constant step size λ fails to produce a valid mesh after 30 iterations.

The final test case is a more complicated three-dimensional wing-body-flap-engine configuration, which consists of a fuselage, main wing, leading-edge slat, double-slotted trailing-edge flaps, and a nacelle-pylon configuration, as shown in Fig. 11. The unstructured mesh is a highly stretched viscous hybrid mesh that consists of approximately 3 million grid points and a total of 11 million cells (tetrahedra, pyramids, and prisms). There are many small gaps between the leading/trailing-edge flaps and the main wing, as shown in Fig. 11. A twisting motion is prescribed on the main wing-flap assembly, and the trailing-edge flap is also prescribed a 10-deg pitching motion, as shown in Fig. 12. The deformed mesh generated by the linear-elasticity approach with the modulus E prescribed as inversely proportional to the cell volume results in a total of 46 negative volume cells in the flap gap region, as shown in Fig. 13.

The objective function used for the optimization problem in this case is given by Eq. (7). The coefficients are set as $a = 1$ and $n = 2$. Figure 14a depicts the convergence of the line-implicit multigrid algorithm for the mesh motion equations and the mesh adjoint equations, in which the residuals of the mesh motion equations are reduced by six orders of magnitude over 150 multigrid cycles, and the residual for the mesh adjoint equations are reduced by approximately 2.5 orders of magnitude over 150 multigrid cycles. The slower convergence of the adjoint problem for this case is unexpected, although the overall impact on the optimization problem was found to be minimal. In Fig. 14b, the LBFGS algorithm recovers a valid grid in six design iterations, whereas the steepest-descent approach with the line-search algorithm fails to recover a valid mesh after 15 design cycles.

In the preceding examples, one design cycle or optimization iteration represents a single mesh motion and mesh adjoint solution. For the line-search and LBFGS algorithms, these function calls may be used either to update the E distribution or to improve the step size and search direction estimates. For each design iteration, the time spent in the optimization algorithm is trivial compared with the time required for the mesh motion and adjoint solutions. The cost of a

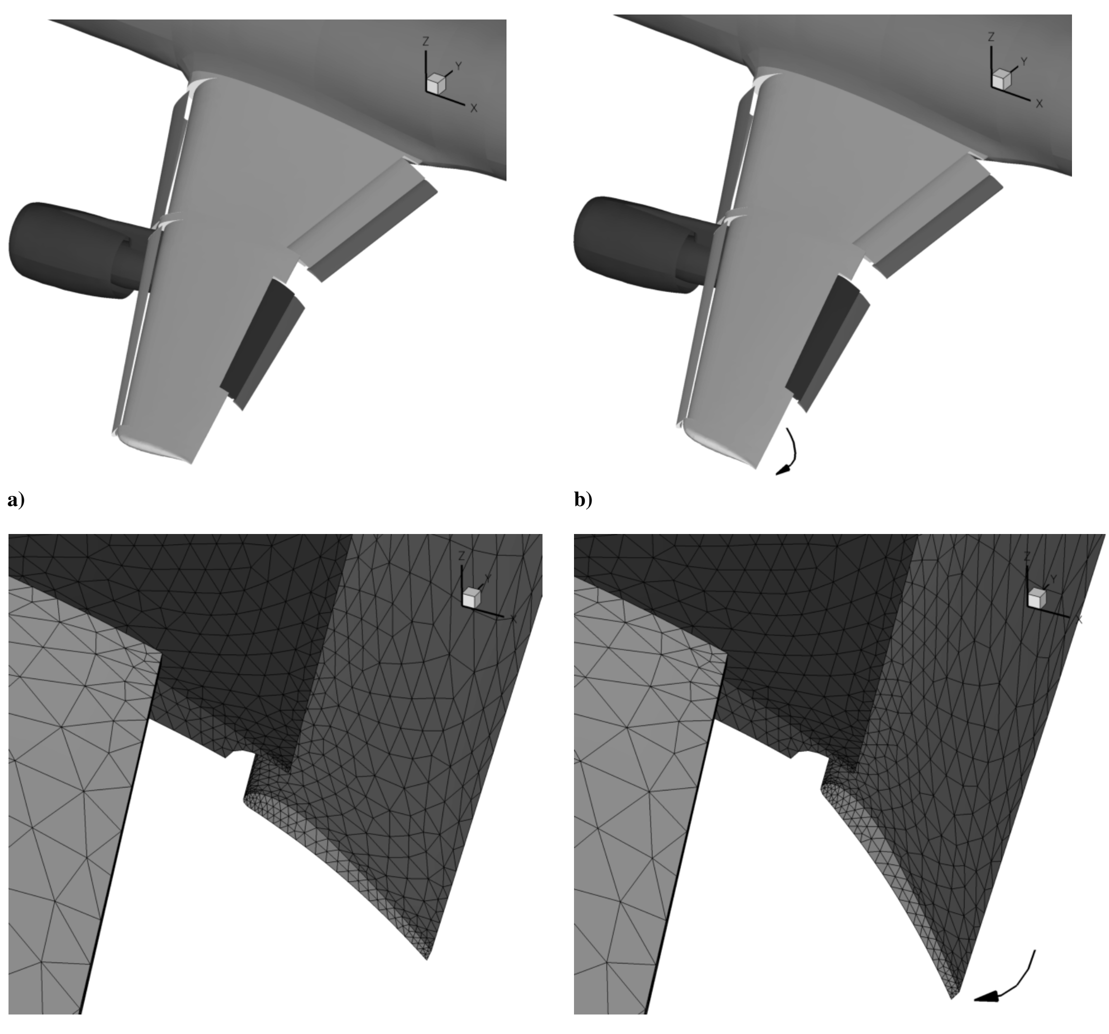


Fig. 12 Illustration of a–b) wing twisting and c–d) flap pitching motion imposed on geometry for mesh deformation test case.

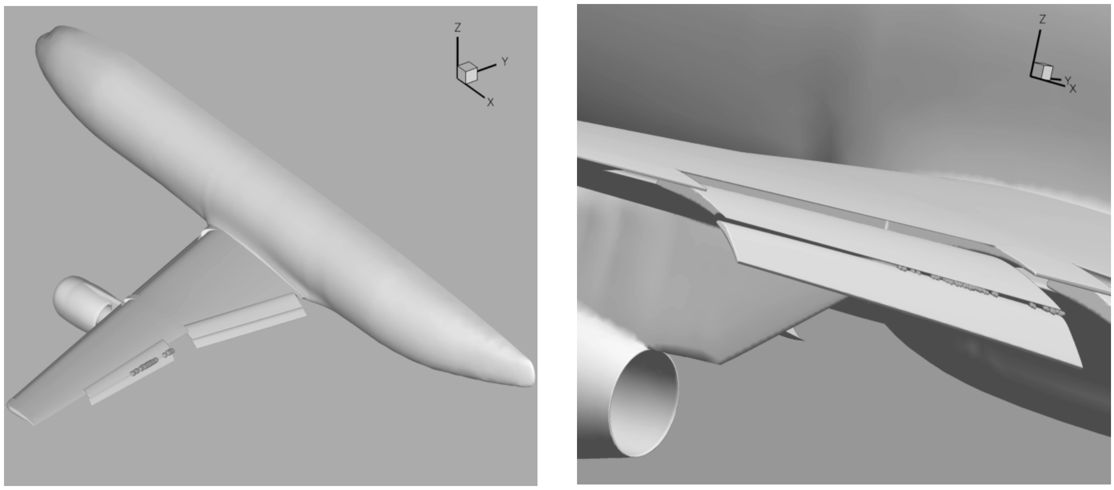


Fig. 13 Location of negative cells produced by mesh deformation solution using prescribed modulus E distribution.

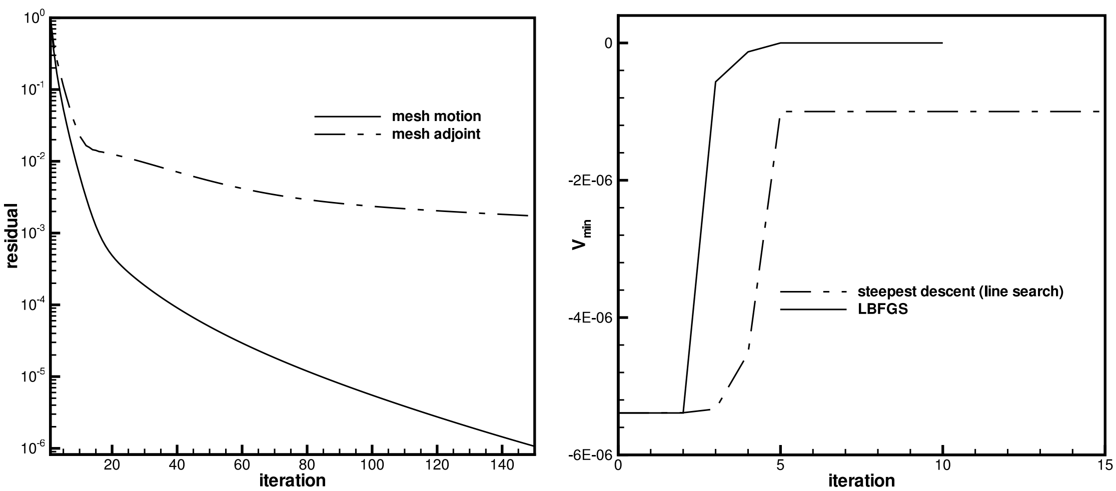


Fig. 14 Convergence history of the a) line-implicit multigrid solver for mesh deformation and mesh adjoint equations for three-dimensional complex wing-body-slat-flap-nacelle-pylon geometry and b) various optimization strategies as measured by the number of optimization cycles requires to recover mesh with fully positive volume cells.

mesh adjoint solution is equivalent to the cost of a mesh motion solution, because both problems contain the same number of unknowns and converge at similar rates. From the preceding test cases, the LBFGS algorithm is seen to require on the order of 5 to 10 design cycles to produce a valid grid. Thus, the mesh optimization problem requires roughly one order of magnitude additional CPU time over a single linear-elasticity mesh deformation calculation. In previous work [9], we compared the cost of solving the mesh deformation problem with the cost of solving an implicit time step for the flow equations, using the line-implicit agglomeration multigrid algorithm for both problems. The mesh motion problem was found to require between 20 and 35% of the CPU time required for the solution of the flow equations at an implicit time step. Therefore, invoking the mesh optimization procedure at each implicit time step within an unsteady flow calculation will not be practical. Rather, the mesh optimization procedure can be employed as a plug-in module that is called at critical time steps in the simulation to repair the mesh when negative volume or poor quality cells are detected. Alternatively, if the geometry motion is prescribed, an optimal E distribution can be precomputed for the most severe deformation case and employed throughout the simulation with the linear-elasticity mesh deformation equations. For aeroelasticity problems, in which the geometry motion is computed as part of the solution, optimal E distributions may be computed for each modal displacement, and linear combinations of these distributions may be formed during run time, based on the modal response of the structure, or a multipoint optimization problem may be formulated.

VI. Conclusions

An optimization procedure for linear-elasticity-based mesh deformation techniques was formulated and validated in this work. The technique seeks to compute an optimal distribution of the modulus of elasticity used in the linear-elasticity analogy, to enhance the robustness and extend the range of applicability of this mesh deformation technique for large displacement cases.

Note that an alternate optimization approach would be to simply use the grid point coordinates as the design variables to minimize the mesh deformation objective function, as proposed in [21]. However, our approach of using a linear-elasticity analogy and optimizing the distribution of E within this framework is designed, on the one hand, to capitalize on the solid-body displacement preservation property of the linear-elasticity method for regions with high E. On the other hand, we are also able to start with a good initial guess for the distribution of E (i.e., inversely proportional to the cell size or distance from the wall), which should make the optimization problem much more local and thus tractable, compared with a brute-force approach based on grid point coordinates.

Although our proposed optimization procedure is successful at removing distorted or negative volume cells for difficult mesh deformation problems, the optimal E distributions depend on the imposed boundary displacements, and the optimization procedure generally requires an order of magnitude more computational effort than a simple linear-elasticity mesh deformation calculation. The optimization procedure is thus best-suited for use as a plug-in module that is invoked when the nonoptimized mesh deformation approach fails or for precomputing optimal modulus of elasticity distributions that can be used in subsequent simulations.

Future work will concentrate on improving the efficiency of the solution of the optimization problem, including the use of more powerful optimizers, experimentation with different objective functions, and the effect of partially converging intermediate mesh motion and adjoint problems at intermediate design iterations. The formulation of multipoint optimization problems for precomputing optimal E distributions for a range of possible mesh motion problems will also be investigated.

Acknowledgments

This work was supported in part by grant NNL04AA63G from the NASA Langley Research Center. The first author would like to thank M. Nemec, NASA Ames Research Center, for help on the optimization methods.

References

[1] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.
[2] Venkatakrishnan, V., and Mavriplis, D. J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 127, 1996, pp. 380–397. doi:10.1006/jcph.1996.0182
[3] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "Torsional Springs for Two-Dimensional Dynamic Unstructured Fluid Meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 163, No. 1, 1998, pp. 231–245. doi:10.1016/S0045-7825(98)00016-4
[4] Murayama, M., Nakahashi, K., and Matsushima, K., "Unstructured Dynamic Mesh for Large Movement and Deformation," AIAA Paper 2002-0122, 2002.
[5] Baker, T., and Cavallo, P. A., "Dynamic Adaptation for Deforming Tetrahedral Meshes," AIAA Paper 1999-3253, 1999.
[6] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1155–1163.
[7] Johnson, A. A., and Tezduyar, T. E., "Simulation of Multiple Spheres Falling in a Liquid-Filled Tube," *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, 1996, pp. 351–373. doi:10.1016/0045-7825(95)00988-4
[8] Yang, Z., and Mavriplis, D. J., "Unstructured Dynamic Meshes with Higher-Order Time Integration Schemes for the Unsteady Navier-Stokes Equations," AIAA Paper 2005-1222, 2005.
[9] Yang, Z. and Mavriplis, D. J., "Higher-Order Time Integration Schemes for Aeroelastic Applications on Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 138–150. doi:10.2514/1.22847; also AIAA Paper 2006-441, 2006.
[10] Jameson, A., Alonso, J. J., Reuther, J. J., Martinelli, L., and Vassberg, J. C., "Aerodynamic Shape Optimization Techniques Based on Control Theory," AIAA Paper 1998-2538, 1998.
[11] Mavriplis, D. J., "A Discrete Adjoint-Based Approach for Optimization Problems on Three-Dimensional Unstructured Meshes," AIAA Paper 2006-0050, Jan. 2006.
[12] Mavriplis, D. J., "Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Grids," *AIAA Journal*, Vol. 44, No. 1, Jan. 2006, pp. 42–50.
[13] Nocedal, J., and Wright, S., *Numerical Optimization*, Springer, New York, 2000.
[14] Dennis, J. J. E., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Upper Saddle River, NJ, 1983.
[15] Liu, D. C., and Nocedal, J., "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, Vol. 45, 1989, pp. 503–528. doi:10.1007/BF01589116
[16] Nocedal, J., "Updating Quasi-Newton Matrices with Limited Storage," *Mathematics of Computation*, Vol. 35, No. 151, July 1980, pp. 773–782. doi:10.2307/2006193
[17] Mavriplis, D. J., "Multigrid Techniques for Unstructured Meshes," VKI Lecture Series, No. 1995-02, Von Karman Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, 1995.
[18] Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, No. 1, Sept. 1998, pp. 141–165. doi:10.1006/jcph.1998.6036
[19] Nielsen, E. J., Lu, J., Park, M. A., and Darmofal, D. L., "An Implicit Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, Nov. 2004, pp. 1131–1155. doi:10.1016/j.compfluid.2003.09.005
[20] Dwight, R. P., "Robust Mesh Deformation Using the Linear Elasticity Equations," *Fourth International Conference on Computational Fluid Dynamics*, Springer, New York (to be published); also available at http://www.maths.manchester.ac.uk/~rdwight/ [retrieved 27 Sept. 2007].
[21] Lopez, E. J., Nigro, N. M., Storti, M. A., and Toth, J. A., "A Minimal Element Distortion Strategy for Computational Mesh Dynamics," *International Journal for Numerical Methods in Engineering*, Vol. 69, No. 9, 2007, pp. 1898–1929. doi:10.1002/nme.1838

Z. Wang
Associate Editor