COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng 2008; **24**:435–448
Published online 20 June 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm.1017

On the use of general elements in fluid dynamics simulations

S. Chalasani[‡], V. Senguttuvan[§], D. Thompson[*, †, ¶] and E. Luke[‖]

Computational Simulation and Design Center, Mississippi State University, Box 9627, Mississippi State, MS 39762, U.S.A.

SUMMARY

As flow solvers have become more sophisticated, many now have the capability to handle meshes with general polyhedral elements, i.e. general elements. In response, researchers have begun to develop mesh generation strategies that exploit nonstandard element types. General elements have been employed locally in near-body extruded meshes to improve mesh quality and have also been used in solution adaptive mesh refinement algorithms. In this paper, we present two mesh generation strategies that employ general elements. We also discuss how a specific flow solver is formulated to work with general elements. We demonstrate our approach by presenting simulation results for two realistic configurations. We close with a discussion that highlights future research directions. Copyright © 2007 John Wiley & Sons, Ltd.

Received 31 March 2006; Revised 1 February 2007; Accepted 17 April 2007

KEY WORDS: hybrid mesh; solution adaptive mesh; general elements

1. INTRODUCTION

Recent advances in flow solver technology include the development of discretization algorithms that are valid for meshes with nonstandard elements, i.e. elements other than tetrahedrons, hexahedrons, prisms, and pyramids [1–3]. In these implementations, there are no restrictions on the number of edges that define a face or the number of faces that define an element. From the dual standpoints of code development and maintenance, the primary advantage of this approach is that a single flow solver may be employed for both structured grids and unstructured meshes. In response, researchers

*Correspondence to: D. Thompson, Computational Simulation and Design Center, Mississippi State University, Box 9627, Mississippi State, MS 39762, U.S.A.
†E-mail: dst@simcenter.msstate.edu
‡Graduate Research Assistant, currently at Synopsys, Mountain View, CA, U.S.A.
§Graduate Research Assistant, currently at Blue Sky Animation, White Plains, NY, U.S.A.
¶Associate Professor.
‖Assistant Professor.

Contract/grant sponsor: National Science Foundation; contract/grant number: ITR/ACS-00859669
Contract/grant sponsor: NASA Glenn Research Center; contract/grant number: NCC3-994

have begun investigating meshing algorithms that exploit general elements. The resulting meshes, which may contain arbitrary polyhedra, have been dubbed generalized meshes [2].

One application of general elements that has received interest in the literature is their use locally in extruded meshes to improve quality [4–7]. Such an extruded mesh is typically employed as the near-body component of a hybrid mesh [8, 9] near viscous boundaries. These highly anisotropic extruded meshes provide an efficient mechanism to resolve the gradients that are primarily in the direction normal to the viscous boundary. The remainder of the domain is filled with an isotropic tetrahedral mesh. Unfortunately, many hybrid mesh generation methods produce near-body meshes that may contain poor quality elements near convex and concave regions of the geometry. These poor quality elements may lead to a correspondingly poor quality transition between the extruded mesh and the void-filling tetrahedral mesh. General elements have been employed to alleviate this problem by forcing the extruded mesh near the transitional surface to be as isotropic as possible.

General elements have also been employed for solution adaptive mesh refinement [10–13]. The primary advantage of using general elements for mesh refinement is the elimination of problems associated with terminating the refinement. Depending on the specific strategy, refinement in one element may produce a hanging node or a face mismatch on an adjacent element that is not refined. In some sense, all four approaches referenced here attempt to mimic Cartesian mesh refinement [14] because they allow hanging nodes. If hanging nodes are not allowed, neighbour elements must be refined according to some strategy until all hanging nodes are eliminated. One solution to this problem is to define templates that specify allowable refined configurations [15, 16]. By limiting the number of templates, the programming effort and costs can be reduced. As an alternative, Rivara [17] developed a successive bisection method for triangles which minimizes propagation without limiting the number of patterns instead of the usual self-similar refinement. Liu and Joe [18] then extended Rivara's method to three-dimensional tetrahedral meshes.

In this paper, we describe two mesh generation strategies that utilize general elements [7, 13] as well as how a specific flow solver [3] is structured to handle general elements. We present results for two configurations to demonstrate the effectiveness of our approach and make recommendations for future work.

2. MESH QUALITY

In any discussion of mesh quality, the first point that must be addressed is to specify what characteristics define a high-quality mesh. It is well known that solution quality depends on many factors—most importantly, the flow solver itself. Additionally, a 'quality mesh' may not produce a 'quality solution' due to locally insufficient local resolution, etc. Further, a given test case may or may not be discriminating for a specific mesh. In our opinion, the mesh quality issue can only be answered in the context of a specific flow solver.

Mesh quality has been studied extensively and is well understood for simplex elements such as triangles and tetrahedra [19]. Owen *et al.* [20] examined the quality of a pyramidal element by decomposing it into tetrahedral elements and then applying a tetrahedral element quality metric to the constituent elements. Wa and Chen [21] attempted to quantify quality for hexahedral and prismatic elements using a combination of face properties to define a quality metric. Inherent to their approach is the assumption that a volume element is of good quality if its faces are of good quality. This is the philosophy that we have adopted. However, Knupp [22] points out that face-based element quality estimates may not be adequate in some cases. We have made a preliminary

DOI: 10.1002/cnm

attempt to address this issue for meshes with general elements in Chalasani *et al.* [23] for the CHEM flow solver. Additional mesh quality/solver validation efforts are ongoing.

3. QUALITY IMPROVEMENTS IN EXTRUDED NEAR-BODY MESHES

The near-body mesh generation algorithm of Chalasani and Thompson [7] is an advancing layer-type approach that generates a semistructured mesh by extruding from a surface mesh of arbitrary topology. Since generating surface meshes is labour intensive, the objective was to develop a mesh extrusion algorithm that would work with any 'reasonable' surface mesh. Additionally, it was desired to provide an isotropic outer surface upon which any tetrahedral mesher would be able to build a void-filling volume mesh.

The extrusion is a three-step process, shown for clarity in two dimensions in Figure 1. For each layer of the mesh, two layers of an orthogonal reference mesh are algebraically constructed. Nodes 1–3 represent the initial surface mesh for layer k. Nodes 4–6 of the reference mesh are then smoothed using a Poisson equation [24, 25]. The positions of nodes 7–9 are adjusted after each smoothing iteration to enhance orthogonality. Once the smoothing process is complete, local techniques such as edge collapse and face refinement can be employed on the interface between layers k and $k + 1$, i.e. nodes 4–6, to improve the mesh quality near concave and convex regions, respectively. It is this step that produces the general elements in layer k. These operations do not change the connectivity of the initial surface mesh. Nodes 7–9 are discarded and nodes 4–6 are used as the initial surface mesh for the next layer.

3.1. Edge collapse

Figure 2 shows how the edge collapse operation can improve the transition from the near-body mesh to the isotropic tetrahedral mesh that occurs at the outer surface of the extruded mesh. In Figure 2(a), faces are being compressed as the mesh is extruded from the concave corner, which produces a highly nonuniform transitional surface mesh. In Figure 2(b), collapsing selected edges

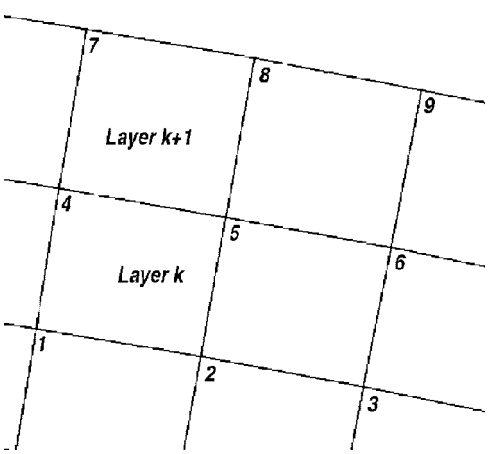


Figure 1. Two layers of a reference mesh are generated algebraically from the initial mesh for the current layer.

DOI: 10.1002/cnm

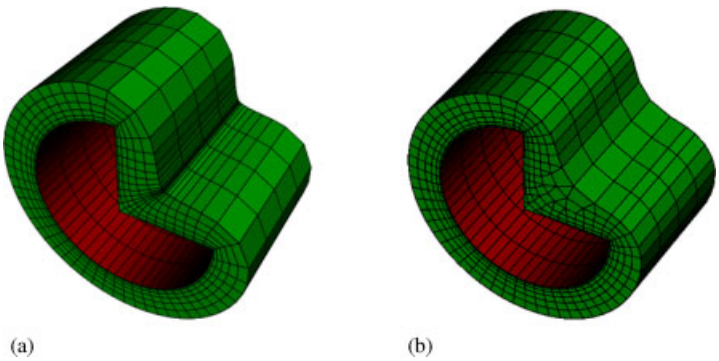


Figure 2. Edge collapse and face refinement can provide more uniform surface for generation of tetrahedral volume mesh: (a) no topology modification and (b) with topology modification.

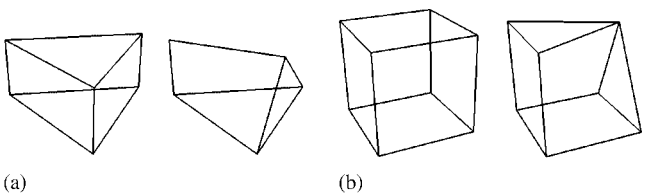


Figure 3. Edge collapse may produce general elements: (a) prism and (b) hexahedron.

produces an outer surface mesh that is more isotropic. Note that the initial surface mesh is the same for both cases. The edges are selected for collapse according to a face-based geometrical sensor, which we call the face aspect ratio. Referring to Figure 1, we can loosely define the face aspect ratio as the height to width ratio of face 1–2–4–5. If this ratio is large, as it occurs near the concave corner in Figure 2, then edge 4–5 is collapsed to a point, either to its midpoint or one of its endpoints. After edge collapse, the face and element topologies are updated. This operation may result in general elements depending on the number and relative positions of edges selected for collapse. Figure 3 shows the general elements that are obtained if a single edge is collapsed for a prism and for a hexahedron.

3.2. Face refinement

A face refinement operation can provide better resolution near convex regions of the geometry and improve the transition from the near-body extruded mesh to the isotropic tetrahedral mesh. In Figure 2(a), the faces near the two convex corners are stretched as the mesh is extruded which, again, produces nonuniformity in the outer surface mesh. Figure 2(b) shows that face refinement can improve the quality of the transitional surface mesh. Edges are selected for refinement based on a face-based geometrical property we denote as the divergence angle. Again referring to Figure 1, the divergence angles for a face are the angles between edges 1–2 and 1–4 and between edges 2–1 and 2–5. If the divergence of a face exceeds a prescribed threshold, as it does near the convex corners in Figure 2, the opposing edge, in this case edge 4–5, is marked for refinement and bisected. A face is refined according to the number of its edges that are selected for refinement.

DOI: 10.1002/cnm

After refinement, the face and element topologies are updated. Edge-based templates are employed to determine how a face should be refined [7]. This operation produces 'hanging nodes' and results in general elements.

4. ISOTROPIC SOLUTION ADAPTIVE MESH REFINEMENT

An isotropic, face-based refinement strategy that exploits general elements to localize the effects of refinement is described by Senguttuvan *et al.* [13]. The isotropic refinement strategy is denoted as face based because it is the refinement of the faces that determines how the elements are refined. An n-sided face is subdivided into n faces by inserting a new node at the centroid of the face and connecting it to the midpoint of each edge. These newly created edges form faces that divide the interior of the element. This approach can be thought of as being analogous to Cartesian mesh refinement. A step-by-step procedure is described below for refinement of any type of element:

1. Some criteria are applied to determine if an element should be subdivided. An element is also marked for subdivision if a neighbouring element has two or more additional levels of subdivision. The following steps are performed for all the elements marked for subdivision.
2. Each edge in the element is subdivided by inserting a node at its midpoint.
3. For each n-sided face in the element:
   - A new node is inserted at the face centroid.
   - The node at the midpoint of each edge is connected to the node at the face centroid to form n new edges.
   - Each node on the face lies exactly on two edges and it forms a four-sided face with the midpoints of those two edges and the centroid of the face so that n four-sided faces are formed.
4. For the n-faced element with e edges and v nodes:
   - A new node is inserted at the element centroid.
   - The node at the centroid of each face is connected to the new node to form n new edges.
   - Each edge lies exactly on two faces and the node at the midpoint of the edge forms a four-sided face with the centroid of the two faces and the centroid of the element. e four-sided faces are formed.
   - For each node, an element is formed. New faces on the surface containing this node are added to the element. If three of the four nodes of a face (the fourth is the centroid of the element) are part of the formed element, the face is added to the element. This creates v elements.
5. After the first four steps are completed, the connectivity of the mesh is updated. If any face contains a refined edge, that edge is removed and is replaced by the subdivided and newly formed edges. If any element contains a refined face, that face is removed and is replaced by the subdivided and newly formed faces.

A mesh state file that contains information such as element refinement level and parent details is outputted to ensure that subsequent refinement cycles are performed in a consistent manner [13].

This approach has many of the advantages and disadvantages of Cartesian mesh refinement. While the isotropic refinement maintains element quality, the newly created nodes and edges

DOI: 10.1002/cnm

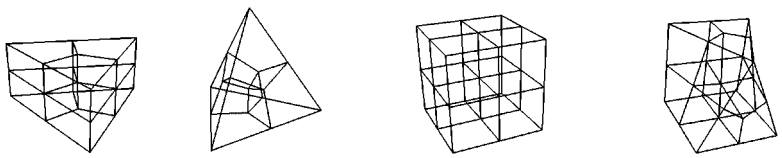


Figure 4. Face-based isotropic refinement produces quadrilateral faces and hexahedrons unless more than three edges are incident on a node: (a) prism; (b) tetrahedron; (c) hexahedron; and (d) general element.

can increase the size of the mesh quite remarkably. Refinement of several element types using the face-based strategy is shown in Figure 4. As shown in the figure, faces formed through refinement of an element are always quadrilaterals. Of course, hanging nodes on adjacent elements will produce other face types.

Since no smoothing or reconnection is employed, the isotropic refinement strategy described here tends to maintain element quality, either good or bad, in the sense that existing face angles are not subdivided. Introduced faces, which are not triangular, may have large (approximately 180°) face angles, however, and solution quality depends on the flow solver.

5. FLOW SOLVER

In this section, details of the formulation of a robust Navier–Stokes solver [3] for meshes with general elements are described briefly. A temporally and spatially second-order accurate method is developed using a finite-volume approach formulated for general elements.

The Navier–Stokes equations integrated over an arbitrary control volume Ω_c (closed by a boundary $\partial\Omega_c$) are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_c(t)} Q\,\mathrm{d}V + \int_{\partial\Omega_c(t)} (F_\mathrm{i} - F_\mathrm{v})\cdot\tilde{n}\,\mathrm{d}S = 0 \tag{1}$$

where the vectors of conservative state variables, Q, inviscid flux, F_i, and viscous flux, F_v, are given by

$$Q = \begin{bmatrix} \rho \\ \rho\tilde{u} \\ \rho e_0 \end{bmatrix}, \quad F_\mathrm{i} = \begin{bmatrix} \rho\tilde{u} \\ (\rho\tilde{u}\tilde{u} + p\tilde{\tilde{I}}) \\ (\rho e_0 + p)\,\tilde{u} \end{bmatrix}, \quad F_\mathrm{v} = \begin{bmatrix} 0 \\ \tilde{\tilde{\tau}} \\ (\tilde{u}\cdot\tilde{\tilde{\tau}} + \tilde{q}) \end{bmatrix} \tag{2}$$

where $\tilde{u}$ is the fluid velocity vector, p is the fluid pressure, e_0 is the total internal energy of the fluid, $\tilde{\tilde{\tau}}$ is the deviatoric stress tensor, and $\tilde{q}$ is the heat flux vector due to conduction.

5.1. Numerical approximations to spatial integrals

The numerical solution of the governing equations, Equation (1), is obtained by applying the finite-volume method. This approach is frequently used because it guarantees that numerical truncation errors do not violate conservation laws providing for more accurate simulations of fluids, in particular, problems involving moving shocks. The numerical integration of Equation (1) begins

DOI: 10.1002/cnm

with approximations to volume and surface integrals. For the volume integrals, a second-order midpoint rule is used. For example, the numerical integration of Q results in

$$\int_{\Omega_c(t)} Q(\tilde{x}, t)\, \mathrm{d}V = Q_c(t)\mathscr{V}_c(t) \tag{3}$$

where $Q_c(t)$ is the value of Q at the centroid of element c, and $\mathscr{V}_c(t)$ is defined by

$$\mathscr{V}_c(t) = \int_{\Omega_c(t)} \mathrm{d}V \tag{4}$$

The numerical integration of the surface integral in Equation (1) is accomplished by summing the contributions of each of the NF faces of element c. Each individual contribution is again approximated using the midpoint rule. The flux function itself requires additional numerical treatment, such as an approximate Riemann solver. For now, assume that the flux can be approximated by a function, $\hat{F}$, of conservative values to the left and right of the face. The numerical integration of $F = F_\mathrm{i} - F_\mathrm{v}$ results in the following:

$$\int_{\partial\Omega_c(t)} F\, \mathrm{d}S = \sum_{f=1}^{\mathrm{NF}_c} \int_{\partial\Omega_{c,f}(t)} F\, \mathrm{d}S \approx \sum_{f=1}^{\mathrm{NF}_c} \mathscr{A}_{c,f}(t)\hat{F}_f \tag{5}$$

where the area of the face, $\mathscr{A}_{c,f}(t)$, is defined as

$$\mathscr{A}_{c,f}(t) = \int_{\partial\Omega_{c,f}(t)} \mathrm{d}S \tag{6}$$

At this point, Equation (1) is numerically approximated by

$$\frac{\mathrm{d}}{\mathrm{d}t}[\mathscr{V}_c(t)Q_c(t)] + \sum_{f=1}^{\mathrm{NF}_c} \mathscr{A}_{c,f}(t)\hat{F}_f = \mathscr{V}_c(t)\dot{W}_c(t) \tag{7}$$

Note that the differential term that remains in this equation applies to the product of volume and conservative state vector for the element. For deforming volumes, careful treatment of this term is required, in particular, to ensure satisfaction of the geometric conservation law [26].

For a second-order treatment of the above integrations, a piecewise linear reconstruction is used to approximate the solution variables within elements. This technique can be used to provide the left and right states to the inviscid flux (with appropriate limiting) and the gradients required for the viscous fluxes (with appropriate face averaging). The linear reconstruction is derived from element values and gradients by applying a second-order Taylor-series expansion of given function Y:

$$Y(\tilde{x}) = Y(\tilde{x}_c) + \nabla Y(\tilde{x}_c) \cdot \delta\tilde{r} + O((\delta\tilde{r})^2) \tag{8}$$

where ∇Y is the gradient of a selected reconstruction variable Y and $\delta\tilde{r}$ is the vector from the centre of the element $\tilde{x}_c$ to the desired point $\tilde{x}$.

Thus, a piecewise linear reconstruction of the primitive variables in element c with centroid $\tilde{x}_{c,c}$ is given by

$$Y_c^\mathrm{R}(\tilde{x}) = Y_c + \nabla Y_c \cdot \delta\tilde{r} \tag{9}$$

where $Y_c^\mathrm{R}(\tilde{x})$ is the reconstructed function, Y_c is the value of function Y at the element centroid, and ∇Y_c is the computed gradient within element c.

DOI: 10.1002/cnm

The gradient at the element centroid, ∇Y_c, is evaluated by minimizing the weighted error between the reconstructed function and neighbouring element values. Thus, the error of the reconstructed function minimizes the weighted L_2-error in the neighbourhood of element c

$$\text{error} = \sqrt{\sum_{f=1}^{\text{NF}_c} (\mathscr{A}_{c,f}[Y_c^{\text{R}}(\tilde{x}_{c,d}) - Y_d])^2} \tag{10}$$

where $\mathscr{A}_{c,f}$ is the area of the common face shared by the adjoining elements c and d, and $\tilde{x}_{c,d}$ is the centroid of element d. Area weighting of the error, inspired by a Green's theorem perspective of gradient computations, is used to recover correct thin-layer behaviour for viscous meshes near curved surfaces, where highly anisotropic prismatic elements are usually found. This weighting also more appropriately weights errors for general elements containing hanging nodes than the more commonly used reciprocal distance weighting approach. The gradient that minimizes Equation (10) is obtained by using a standard linear least-squares approach. A QR factorization method is used to solve the resulting (overdetermined) linear system for enhanced numerical stability.

5.2. *Geometric integrations*

In the previous section, numerical integrations over surfaces and volumes were described without explicitly defining the formulas used to evaluate Equations (4) and (6). Here, the numerical integrations that approximate the volumes and areas of general elements and faces are discussed in more detail. Several considerations have to be made regarding these computations. First, when the nodes of a general face are not co-planar, the geometry of the face is not uniquely specified. A unique specification of the geometry of such faces can be obtained by subdividing the face into triangles: using a symmetric decomposition, each edge of the face, combined with the face centroid, creates a triangle. This strategy allows the face geometry to be independent of data structures used to describe generalized meshes (the geometry is the same regardless of the ordering of the face edges). Using this technique, the area for the face is determined numerically using the following summation:

$$\tilde{\mathscr{A}}_{c,f} = \frac{1}{2} \sum_{e=1}^{\mathbf{NE}_f} (\tilde{x}_{1,e} - \tilde{x}_{c,f}) \times (\tilde{x}_{2,e} - \tilde{x}_{c,f}) \tag{11}$$

where $\tilde{x}_{1,e}$ and $\tilde{x}_{2,e}$ are the positions of the two nodes of edge e in counter-clockwise order, and $\tilde{x}_{c,f}$ is the face centroid. Here, the computed area is a vector, which is represented as the product of the face normal vector and the face area, $\tilde{\mathscr{A}} = \mathscr{A}\tilde{n}$.

For numerical approximations to the volume integrals, it is essential to achieve a geometric conservation of volume, avoiding truncation errors in integrating volumes that would yield inaccurate total volume computations. To accomplish this, the volume of an element is determined by decomposing it into nonoverlapping tetrahedra and then summing the individual tetrahedral volumes. The volume of a tetrahedron is given by

$$\mathscr{V}_{\text{tet}} = \frac{1}{3!}(\tilde{a} \cdot (\tilde{b} \times \tilde{c})) \tag{12}$$

where $\tilde{a}$, $\tilde{b}$, and $\tilde{c}$ are the three edge vectors from a given node of the tetrahedron. The volume of a general element is then computed by using the triangulation of each face and the element

DOI: 10.1002/cnm

centroid: each surface triangle and the element centroid define a tetrahedron. The element volume is the sum of the tetrahedral volumes. Thus, the volume of an element is computed by combining Equations (11) and (12) and factoring to obtain

$$\mathscr{V}_c = -\frac{1}{3}\sum_{f=1}^{\mathrm{NF}_c} (\tilde{x}_{c,c} - \tilde{x}_{c,f}) \cdot \tilde{\mathscr{A}}_{c,f} \tag{13}$$

where $\tilde{x}_{c,c}$ is the element centroid. Note that this computation assumes that the face normals point outward from the element, which explains the negative sign. This approach can be used to compute the correct volume of any element, including highly nonconvex elements.

One comment should be made regarding the centroids used in these computations. We have experimented with using two different definitions of centroid: one where the centroid is computed assuming that mass is distributed evenly through the element volume, and another where the mass is assumed to be distributed evenly over the element surface. Through experimentation we have determined that while both centroids recover second-order accuracy in the reconstruction step, the surface mass centroid appears to produce qualitatively better solutions (e.g. better contour shapes). Thus, by default we used a surface mass centroid in the above formulas.

5.3. Error estimation

One method that can be employed to determine where to apply mesh refinement is in error estimation methods. For the work presented here we use a simple error estimation scheme that is applicable to schemes that employ discontinuous reconstructions. In this error estimation scheme, we associate the jump in the discontinuities of reconstructed variables at element faces as a measure of the solution error. For smooth functions, these interface jumps will drop in proportion to the solver truncation error of the inviscid operator, and thus provide a reasonable error estimate. While the approach overestimates error when the solution is discontinuous, it is often warranted to refine discontinuous regions as well, at least in the early refinement steps. Once the error estimate is computed, each node that has an error more than one standard deviation above the mean error is marked for refinement.

6. RESULTS

In this section, we present meshes and flow simulations for the X38 crew return vehicle (CRV) and a hydrogen/oxygen single element injector (SEI). In both cases, a baseline hybrid mesh is generated, a preliminary solution is obtained on this mesh, a solution adaptive mesh refinement is performed, and a solution is obtained on the refined mesh. Multiple adaptation cycles are performed for the SEI. It is not our intent to report an in-depth validation effort in this paper. This section was included to demonstrate that these technologies can be employed to simulate flow fields with complex physics for realistic configurations.

Evaluating the computational cost of this refinement strategy is not straightforward. For the present solver, the computational cost, in residual, matrix assembly, and sparse matrix solution, is dominated by the number of faces in the mesh. Thus, the number of faces in the mesh provides a rough approximation to the computational cost. The number of elements is directly proportional to the degrees of freedom in the problem, and provided they are useful degrees of freedom, can be used to estimate the effectiveness of the adaptation. Thus, one measure of the computational

DOI: 10.1002/cnm

efficiency is the number of faces per element, with smaller values indicating a higher efficiency. A refinement strategy that introduces hanging nodes will introduce additional faces, and thus reduce computational efficiency of the approach, with the advantage of placing degrees of freedom exactly where they are required. A better measure, but more difficult to estimate, would be the ratio of the number of faces to the productivity of the introduced degrees of freedom. For example, if we add elements to the mesh to remove hanging nodes, it may appear to improve the computational efficiency by the straightforward ratio of faces to elements. However, these elements were added to satisfy a topological constraint instead of satisfying error estimator requirements. Thus, the resulting degrees of freedom will presumably be less productive in reducing solution error. In the following examples, we provide metrics of the mesh elements, faces, and their ratio in order to provide a rough estimate of the computational efficiency of the approach.

6.1. X38 CRV

The first example we consider is the simulation of the X38 CRV under the flight profile of Mach 10 flow at an angle of attack of 60° at high altitude conditions: air composed of 78% nitrogen by mass at a pressure of 29 800 Pa and a temperature of 223 K. To model the effects of the chemical reactions occurring in the flow, we employ a five-species, 17-reaction finite-rate chemistry model for air originally developed by Kang and Dunn [27]. For this simulation, the vibrational contributions to internal energy are assumed to be in thermodynamic equilibrium and we neglect effects of ionization that would be more prevalent at higher Mach numbers.

The baseline mesh employed for this simulation was generated using the extrusion algorithm described in Section 3 for the near-body mesh. The remainder of the domain was filled with a tetrahedral mesh. In this simulation, a strong bow shock forms at some distance from the body. Since the flow physics can be highly sensitive to accurately capturing this shock structure and properly predicting its standoff distance, we employ the solution adaptive mesh refinement procedure described in Section 4. Two levels of refinement are performed based on error estimates in the temperature and pressure field reconstructions. The solution and refined mesh are shown in Plate 1. The original mesh is composed of approximately 8.46×10^6 faces and 3.71×10^6 elements, while solution adaptive refinement increases the number of faces by 76.7% and the number of elements by 53.4%. The ratio of the number of faces to the number of elements increases from 2.278 to 2.624 with refinement. General elements represent 1.9 and 7.2% of the total number of elements in the baseline and refined meshes, respectively. The bow shock region is resolved much more sharply on the refined mesh as can be seen in Plate 2.

Figure 5 shows a comparison of metrics computed in the final layer of the extruded near-body mesh for the baseline mesh and a mesh extruded from the same surface mesh without edge collapse and face refinement. These metrics are designed to illustrate the quality of the transition from the extruded mesh to the isotropic tetrahedral mesh. As can be seen from the figure, the use of general elements forces the mesh toward isotropy so that the last layer transitions well to the isotropic tetrahedral mesh. In particular, note the reduction in the number of high aspect ratio faces (left and centre images) and the number of faces with small included angles (right images).

Figure 6 shows plots comparing the distribution of included face angles for the baseline mesh and the refined mesh. We have chosen to show the actual number of occurrences of a given range of included angles instead of using a normalized value because the number of elements increases significantly with refinement. This may give the impression that the number of occurrences of a certain angle range is decreasing, when in fact the opposite may be true. We note that in the

DOI: 10.1002/cnm

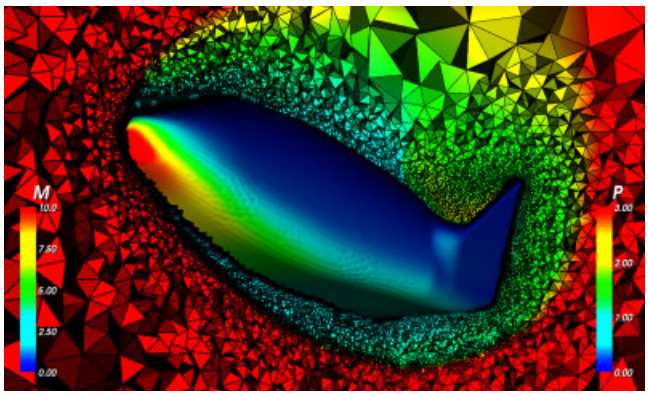
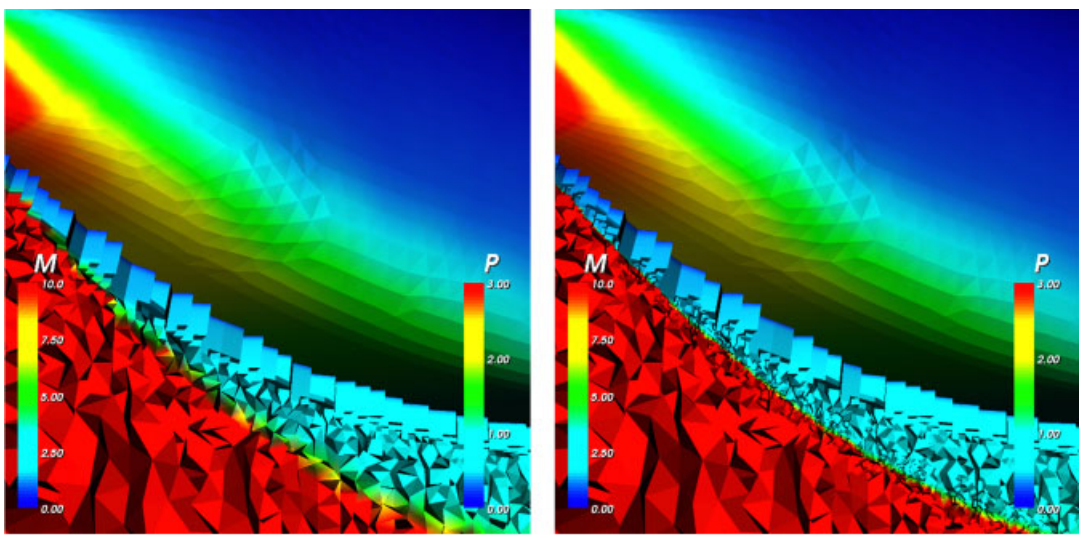


Plate 1. Solution generated for X38 CRV on solution adaptive mesh. Two cutting planes were employed: the body symmetry plane and a plane normal to the longitudinal axis of the body. Body surface is coloured by pressure and field surfaces are coloured by Mach number.



(a) (b)

Plate 2. Detail of X38 solution—baseline mesh and solution adaptive mesh. Body surface coloured by pressure and field surfaces coloured by Mach number: (a) baseline mesh and (b) refined mesh.

DOI: 10.1002/cnm

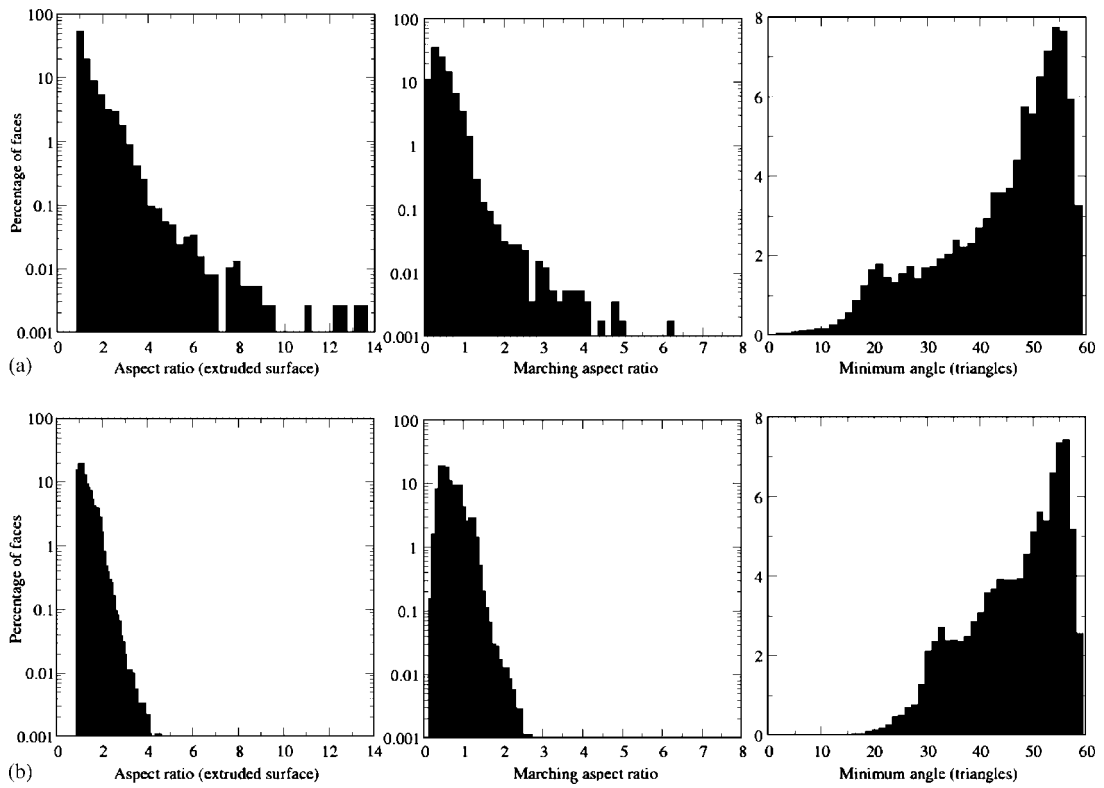


Figure 5. Quality metrics for last layer of extruded baseline X38 mesh—with/without edge collapse and face refinement. The use of general elements improves the transition to the isotropic tetrahedral mesh: (a) no quality improvements and (b) with quality improvements.

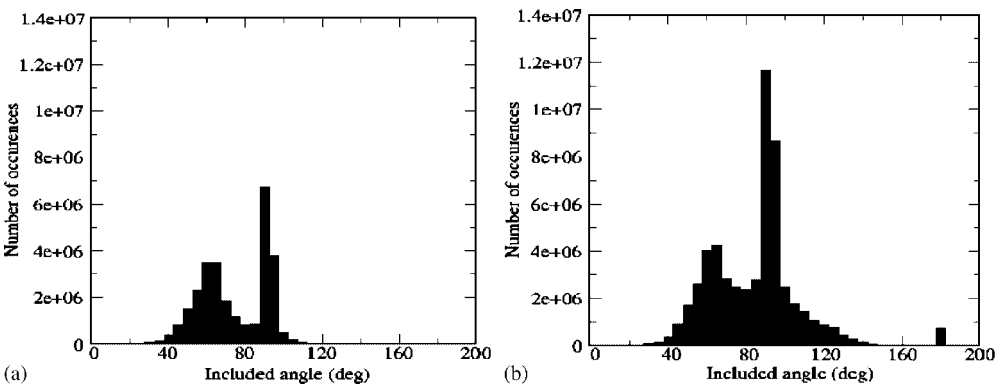


Figure 6. Quality plots for X38 meshes measured using included face angle show effects of isotropic face-based refinement: (a) baseline mesh and (b) mesh after one refinement cycle (two levels of refinement).

DOI: 10.1002/cnm

baseline mesh, there are peaks at 60° and 90° that are associated with the triangular faces in the tetrahedrons and prisms and quadrilateral faces in the prisms, respectively. In the refined mesh, there is a significant increase in the occurrences of included angles bracketing 90°, which occurs due to the subdivision of triangular and quadrilateral faces, and at 180°, which occurs because of the introduction of hanging nodes.

6.2. Single-element injector (SEI)

The second example we consider is simulation of internal flow where close proximity of surfaces poses additional challenges for mesh generation. We simulate the flow of the fuel and oxidizer through the coaxial injector element into the combustion chamber of the SEI rocket motor. The reactants (gaseous preheated hydrogen) and oxidizer (gaseous preheated oxygen) mix and combust. The hot combustion products exit the chamber through a choked nozzle. This simulation represents a single case from a family of experiments performed at the Propulsion Engineering Research Center at Pennsylvania State University [28] for the purpose of validating rocket combustion simulation codes.

For the simulation of the rocket motor we employ the six species 28 reaction hydrogen oxygen combustion model of Evans and Schexnayder [29]. Gaseous oxygen flows through the centre of the injector element with a mass flow of 0.0455 kg/s from the pre-burner at 644 K with a water vapour content of 5.4% by mass. Gaseous hydrogen flows through the annulus region of the injector with a mass flow of 0.0165 kg/s from the pre-burner at a temperature of 877 K with a water vapour content of 58.8% by mass. The experimental chamber pressure was measured at 750 psi for this case. The geometry of the simulation includes the 6.6 in. long injector post, the 11 in. long combustion chamber, and the 2 in. long nozzle. Comparisons to experimental heat fluxes for this and other cases are presented elsewhere [23].

For this simulation, the Reynolds-averaged (RANS) two-equation shear stress transport (SST) [30] turbulence model is employed to obtain a steady-state solution of the combustion in the rocket motor. In the simulation of the combustion chamber, a thin flame forms that attaches to the injector post tip. For the initial baseline mesh, the resolution in the region near the tip is insufficient to model this thin flame region and so adaptation is used to properly resolve the flame structure. For this simulation, we perform three refinement cycles where error in temperature and hydroxyl mass fractions are used to mark elements for refinement.

An axisymmetric mesh with 428 131 faces and 114 119 elements was employed as the baseline mesh. Three refinement cycles introduce a modest increase in mesh size, a 37.2% increase in the number of faces and a 33.8% increase in the number of elements. The ratio of the number of faces to the number of elements increases from 3.752 to 3.847 with refinement. The flame in the injector post region is significantly more resolved in the refined mesh as can be seen in the solution comparisons shown in Plate 3. The baseline mesh produces an unrealistically thick and smeared flame region, while the refined mesh produces a crisp flame front as is particularly evident in the hydroxyl plot. We also note in plots of the estimated error in the hydroxyl radical, adaptation significantly reduced estimated error (by more than a factor of 10) as can be seen in Plate 4. This error reduction resulted in a significantly different flame length as is shown in Plate 5.

The upper part of the plot shows that the solution on the refined mesh has a crisp flame resolution near the injector post resulting in less flame spread and a longer overall flame structure. This result is not surprising as the effect of the coarse mesh is to introduce nonphysical numerically induced

DOI: 10.1002/cnm

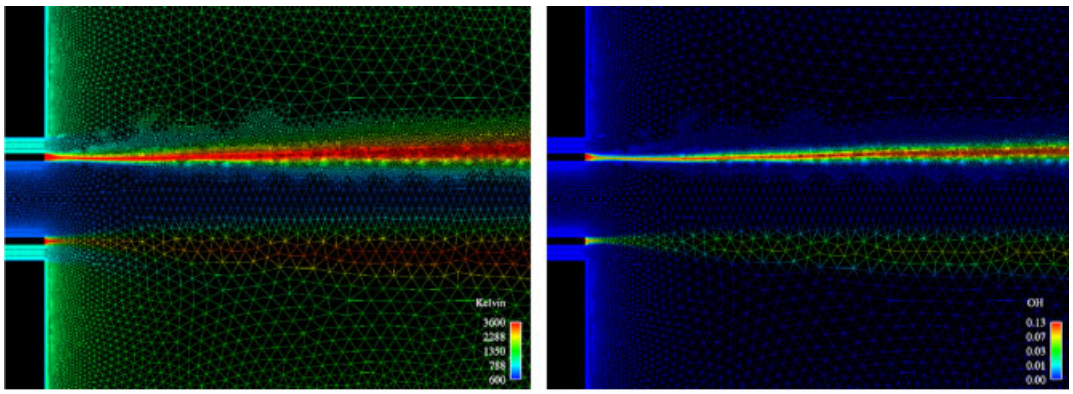


Plate 3. The flame is better resolved in the refined mesh (top) than in the baseline mesh (bottom): (a) temperature and (b) hydroxyl species (OH) mass fractions.

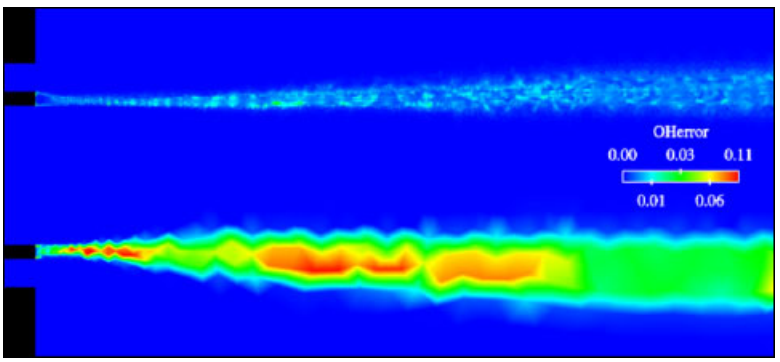


Plate 4. Comparison of estimated error on baseline (bottom) and refined (top) meshes shows that the error is reduced significantly in the refined mesh.

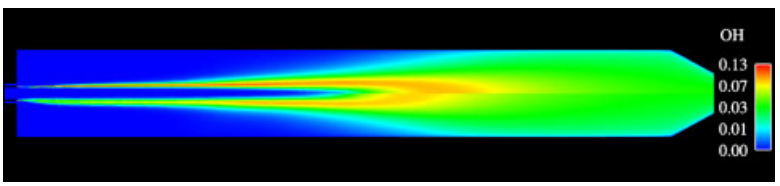


Plate 5. Comparison of hydroxyl species mass fraction shows that predicted flame length for the baseline mesh (bottom) is significantly shorter than in the solution on the refined mesh (top).

DOI: 10.1002/cnm

mixing. The refined mesh possessed much less of this error, which resulted in a longer flame length.

7. CONCLUSION

We consider the near-body mesh extrusion algorithm a maturing technology and do not foresee significant additional research in this area. In contrast, solution adaptive mesh refinement has many unanswered questions. Although our isotropic refinement strategy is well developed, we believe that anisotropic refinement may ultimately prove more effective because of its enhanced efficiency relative to isotropic refinement. We have developed an edge-based strategy [13] for anisotropic refinement, so-called because subdivision of the edges drives how an element is refined. Once the nodes are marked, it is necessary to determine how individual faces and elements are going to be subdivided in an intelligent fashion to maintain mesh quality. Of course, for anisotropic refinement to be effective, there must be a preferred direction in the physical phenomenon of interest and element faces should be aligned with these features for maximum effectiveness. In cases where there is no identifiable direction, isotropic refinement is preferable.

Significant work is also needed to determine the limitations of the formulation discussed here. We have developed a new stochastic approach to perform code verification using the method of manufactured solutions (MMS) [31]. In this technique, a periodic function $f(x, y, z, t)$ is employed and the mesh is translated to random locations within a period of this function to determine the mean error generated for that mesh. To measure convergence, the mesh is scaled to smaller dimensions and a random sampling of error measurements is employed to determine the mean error on the mesh. Using this technique, the order of accuracy of the scheme can be developed using the same mesh throughout the process. This stochastic MMS technique will be employed to evaluate the effects of different mesh topologies on both the absolute error as well as the solver accuracy.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their helpful comments. This work was partially funded by the National Science Foundation under the Information Technology Research Program (ITR/ACS-00859669) and by NASA Glenn Research Center as part of the Constellation University Institutes Program (NCC3-994).

REFERENCES

1. Strang W, Tomaro R, Grismer M. The defining methods of Cobalt60: a parallel, implicit, unstructured Euler/Navier–Stokes flow solver. *AIAA Paper 99-0786*, January 1999.
2. Koomullil RP, Soni BK. Generalized grid techniques in computational field simulation. In *Numerical Grid Generation in Computational Field Simulations*, Cross M, Eiseman PR, Hauser J, Soni BK, Thompson JF (eds). International Society of Grid Generation (ISGG): Mississippi State, MS, 1998; 521–531.
3. Wu J, Tang L, Luke EA, Tong X-L, Cinnella P. Comprehensive numerical study of jet-flow impingement over flat plates. *Journal of Spacecraft and Rockets* 2002; **39**(3):337–366.
4. Thompson DS, Soni BK. Semistructured grid generation in three dimensions using a parabolic marching scheme. *AIAA Journal* 2002; **40**(2):391–393.
5. Leatham M, Stokes S, Shaw JA, Cooper J, Appa J, Blaylock TA. Automatic mesh generation for rapid-response Navier–Stokes calculations. *AIAA Paper 2000-2247*, June 2000.
6. Cary A, Michal T. Generalized prisms for improved grid quality. *AIAA Paper 2001-2552*, June 2001.

DOI: 10.1002/cnm

7. Chalasani S, Thompson DS. Quality improvements in extruded meshes using topologically adaptive generalized elements. *International Journal for Numerical Methods in Engineering* 2004; **60**(6):1139–1159.
8. Shaw JA. Hybrid grids. In *Handbook of Grid Generation*, Thompson JF, Soni BK, Weatherill NP (eds). CRC Press: Boca Raton, FL, 1998; 23-1–23-18.
9. Kallinderis Y. Hybrid grids and their applications. In *Handbook of Grid Generation*, Thompson JF, Soni BK, Weatherill NP (eds). CRC Press: Boca Raton, FL, 1998; 25-11–25-18.
10. Spragle GS, Smith WA, Weiss JM. Hanging node solution adaptation on unstructured meshes. *AIAA Paper 95-0216*, January 1995.
11. Mueller J, Schonfeld T, Rudgyar M. A comparison of the treatment of hanging nodes for hybrid grid refinement. *AIAA Paper 97-1859*, June 1997.
12. Zhang SJ, Liu J, Chen Y-S, Wang T-S. Adaptation for hybrid unstructured grid with hanging node method. *AIAA Paper 2001-2657*, June 2001.
13. Senguttuvan V, Chalasani S, Luke EA, Thompson DS. Adaptive mesh refinement using general elements. *AIAA Paper 2005-0927*, January 2005.
14. Aftosmis M, Berger M, Melton J. Adaptation and surface modeling for Cartesian mesh methods. *AIAA Paper 95-1725*, June 1995.
15. Mavriplis DJ. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *International Journal for Numerical Methods in Fluids* 2003; **34**(2):93–111.
16. Biswas R, Strawn RC. Tetrahedral and hexahedral mesh adaptation for CFD problems. *Applied Numerical Mathematics* 1998; **26**:135–151.
17. Rivara M-C. A grid generator based on 4-triangles conforming mesh refinement algorithms. *International Journal for Numerical Methods in Engineering* 1987; **24**(7):1343–1354.
18. Liu A, Joe B. Quality local refinement of tetrahedral meshes based on bisection. *SIAM Journal on Scientific Computing* 1995; **16**(6):1269–1291.
19. Weatherill NP. Unstructured grids: procedures and applications. In *Handbook of Grid Generation*, Thompson JF, Soni BK, Weatherill NP (eds). CRC Press: Boca Raton, FL, 1998; 26-1–26-37.
20. Owen S, Canann S, Saigal S. Pyramid elements for maintaining tetrahedra to hexahedra conformability. *Proceedings on Trends in Unstructured Mesh Generation* 1997; **AMD 220**:123–129.
21. Wa K, Chen Z. A simple and effective mesh quality metric for hexahedral and wedge elements. *Proceedings of the 9th International Meshing Roundtable*. Sandia National Laboratories: Albuquerque, NM, 2000; 325–333.
22. Knupp PM. On the invertibility of the isoparametric map. *Computer Methods in Applied Mechanics and Engineering* 1990; **78**(3):313–329.
23. Chalasani S, Luke E, Senguttuvan V, Thompson DS. Assessing generalized mesh quality via CFD solution validation. *AIAA Paper 2005-0627*, January 2005.
24. Knupp PM. Winslow smoothing on two-dimensional unstructured meshes. *Proceedings of the 7th International Meshing Roundtable*. Sandia National Laboratories: Albuquerque, NM, 1998; 449–457.
25. Thompson D, Chalasani C, Soni BK. Generation of volume meshes by extrusion from surface meshes of arbitrary topology. *Proceedings of the 9th International Meshing Roundtable*. Sandia National Laboratories: Albuquerque, NM, 2000; 385–393.
26. Thomas PD, Lombard CK. The geometric conservation law—a link between finite-difference and finite-volume methods of flow computation on moving grids. *AIAA Paper 78-1208*, September 1978.
27. Kang SW, Dunn MG. Theoretical, measured electron-density distributions for the RAM vehicle at high altitude. *AIAA Paper 72-689*, January 1972.
28. Marshall W, Pal S, Woodward R, Santoro R. Benchmark wall heat flux data for a GO2/GH2 single element combustor. *AIAA Paper 2005-3572*, July 2005.
29. Evans JS, Schexnayder CJ. Influence of chemical kinetics and unmixedness on burning in supersonic hydrogen flames. *AIAA Journal* 1980; **18**(2):188–193.
30. Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal* 1994; **32**(8):1598–1605.
31. Hebert S, Luke EA. Honey, I shrunk the grids! A new approach to CFD verification. *AIAA Paper 2005-0685*, January 2005.

DOI: 10.1002/cnm