A HIGH-ORDER METHOD USING UNSTRUCTURED GRIDS FOR THE AEROACOUSTIC ANALYSIS OF REALISTIC AIRCRAFT CONFIGURATIONS

Harold L. Atkins[*]and David P. Lockard[†]

NASA Langley Research Center

Hampton, VA 23681

Abstract

A method for the prediction of acoustic scatter from complex geometries is presented. The discontinuous Galerkin method provides a framework for the development of a high-order method using unstructured grids. The method's compact form contributes to its accuracy and efficiency, and makes the method well suited for distributed memory parallel computing platforms. Mesh refinement studies are presented to validate the expected convergence properties of the method, and to establish the absolute levels of a error one can expect at a given level of resolution. For a two-dimensional shear layer instability wave and for three-dimensional wave propagation, the method is demonstrated to be insensitive to mesh smoothness. Simulations of scatter from a two-dimensional slat configuration and a three-dimensional blended-wing-body demonstrate the capability of the method to efficiently treat realistic geometries.

Introduction

Despite recent advances in computational aeroacoustics (CAA), the simulation of acoustic scatter off realistic complex geometries remains problematic. The high-accuracy schemes developed for CAA generally have spatial operators with large stencils that require smooth meshes and are poorly suited to grids around realistic aircraft configurations. Even the process of generating a block-structured mesh without the smoothness required for a high-accuracy method is a time-consuming process often measured in weeks. Unstructured grids about complex geometries are more easily generated, and for this reason, methods using unstructured grids have gained favor for aerodynamic analyses. However, they have not been utilized for acoustics problems because the methods are generally low-order and incapable of propagating waves without unacceptable levels of dissipation and dispersion.

*Senior Member, AIAA. Senior Research Scientist NASA LaRC

†Senior Member, AIAA. Research Scientist NASA LaRC

Attempts to extend finite-difference and finite-volume methods for unstructured grids to high-order by increasing the stencil size have introduced storage and robustness problems.

The discontinuous Galerkin method[1,2] is a compact finite-element projection method that provides a practical framework for the development of a high-order method using unstructured grids. Higher-order accuracy is obtained by representing the solution as a high-degree polynomial whose time evolution is governed by a local Galerkin projection. This approach results in a compact and robust method whose accuracy is insensitive to mesh smoothness. The traditional implementation of the discontinuous Galerkin method employs numerical quadrature for the evaluation of the integral projections and is prohibitively expensive. Atkins and Shu[3] introduced the quadrature-free formulation in which the integrals are evaluated a-priori and exactly for a small collection of similarity elements. The approach has been demonstrated to possess the accuracy required for acoustics even in cases where the grid is not smooth. Other issues such as non-reflecting boundary conditions and the treatment of non-linear fluxes have also been studied in earlier work.[4,5]

A major advantage of the discontinuous Galerkin method is that its compact form readily permits a heterogeneous treatment of a problem. That is, the element topology, the degree of approximation, even the choice of governing equations, can be allowed to vary from element to element with no loss of rigor in the method. To take advantage of this flexibility, an object-oriented C++ computer program that implements the discontinuous Galerkin method has been development and validated. To date, however, most of the applications have involved benchmark problems for aeroacoustics[6] with relatively simple two-dimensional geometries. Recent work has been aimed at adding and validating additionally capability that is essential to the aeroacoustic analysis of large complex configurations.

This paper describes the extension of the method to three-dimensions, the treatment of nonuniform mean flows, and the efficient use of parallel computing platforms. With these new capabilities, this tool will enable rapid aeroacoustic analyses of realistic aircraft configurations. When coupled with currently available

grid generators and large parallel computers, the entire process of mesh generation, problem setup, and calculation can be performed in the time span of a few hours or days instead of weeks or months.

Governing Equations

Numerical Method

The discontinuous Galerkin method is readily applied to any equation of the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \vec{\mathbf{F}} = \mathbf{S} \tag{1}$$

on a domain that has been partitioned into non-overlapping elements that cover the domain. The method is defined by approximating the solution in each element Ω in terms of an appropriate local set of basis functions $\{ b_k \mid 1 \leq k \leq N \}$

$$\mathbf{Q} \approx \mathbf{V} = \sum_{k=1}^{N} \mathbf{v}_k b_k$$

and by performing a local integral projection of the governing equations onto the basis set in each element.

$$\int_{\Omega} b_k \left(\frac{\partial \mathbf{V}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{V}) \right) d\Omega = \int_{\Omega} b_k \mathbf{S}(\mathbf{V}) \, d\Omega \tag{2}$$
$$\forall \; 1 \leq k \leq N$$

The coefficients of the approximate solution $\mathbf{v}_k$ are the new unknowns, and the local integral projection generates a set of equations governing these unknowns. Equation (2) is solved in the weak form and, for convenience, is rewritten in terms of local element coordinates (ξ, η, ζ)

$$\int_{\Omega} \left(b_k \frac{\partial \mathbf{V}}{\partial t} - \nabla b_k \cdot \mathbf{J}^{-1} \vec{\mathbf{F}}(\mathbf{V}) \right) J d\Omega$$
$$+ \sum_{e=1}^{E} \int_{\partial \Omega_e} \mathbf{J}^{-1} \mathbf{F}_e^R(\bar{\mathbf{V}}, \bar{\mathbf{W}}) J ds = \int_{\Omega} b_k \mathbf{S}(\mathbf{V}) J d\Omega \tag{3}$$

where E denotes the number of edges, $\bar{\mathbf{V}}$ denotes the trace of $\mathbf{V}$ on edge e, $\bar{\mathbf{W}}$ denotes the trace of the approximate solution in the neighboring element on edge e,

$$\mathbf{J} = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)}, \quad \text{and} \quad J = |\mathbf{J}|.$$

The term edge will be used to refer to any segment of an element boundary that is shared with a neighboring element or with the physical boundary of the domain. Edges will be referred to as interior edges or boundary edges when it is necessary to distinguish between the two.

Because each element has a distinct local approximate solution, the solution on each interior edge is double valued and discontinuous. An approximate

Riemann flux $\mathbf{F}_e^R(\bar{\mathbf{V}}, \bar{\mathbf{W}})$ resolves the discontinuity and provides the only mechanism by which adjacent elements communicate. The fact that this communication occurs in an edge integral means the solution in a given element $\mathbf{V}$ depends only on the edge trace of the neighboring solution $\bar{\mathbf{W}}$, not on the whole of the neighboring solution $\mathbf{W}$. Also, because the approximate solution within each element is stored as a function, the edge trace of the solution is obtained without additional approximations.

The compactness of the discontinuous Galerkin method has broad implications on the accuracy, robustness, and efficiency of the method. Boundary conditions are easily implemented[4] either by supplying the approximate Riemann flux with the exact external solution $\bar{\mathbf{W}} = \bar{\mathbf{W}}_{\text{exact}}$ (if known), or by reformulating the approximate Riemann flux in terms of the interior solution $\bar{\mathbf{V}}$ and the physical boundary conditions. In either case, the basic formulation of a boundary element is no different from that of an interior element. This is in stark contrast to high-order finite-difference and finite-volume methods in which the standard interior operator must be reformulated near the boundary, usually with a negative impact on the accuracy and robustness of the method.

When the basis functions are polynomials of degree p, the order of accuracy of the method has been proven[1] to be at least $p + \frac{1}{2}$. In practice, the order of accuracy of the method is observed in most cases to be $p + 1$. In the present work, the basis set will be the complete set of local polynomials of a specified degree p:

$$\{ b_k \} = \left\{ \xi^i \eta^j \zeta^k \; \text{for} \; 0 \leq i + j + k \leq p \right\}$$

The Riemann flux will be approximated by a simple Lax-Friedrechs flux of the form

$$\mathbf{F}_e^R(\bar{\mathbf{V}}, \bar{\mathbf{W}}) \equiv \left[\bar{\mathbf{F}}(\bar{\mathbf{V}}) + \bar{\mathbf{F}}(\bar{\mathbf{W}}) - \lambda \left(\bar{\mathbf{W}} - \bar{\mathbf{V}} \right) \right] / 2$$

where $\bar{\mathbf{F}}(\bar{\mathbf{V}}) = \mathbf{J}^{-1} J \vec{\mathbf{F}}(\bar{\mathbf{V}}) \cdot \vec{n}$, λ is greater in magnitude than the eigenvalues of the Jacobian of $\left[\bar{\mathbf{F}}(\bar{\mathbf{V}}) + \bar{\mathbf{F}}(\bar{\mathbf{W}}) \right] / 2$, and $\vec{n}$ is an outward pointing edge normal.

Discretization

The discrete form of equation (3) is usually obtained by evaluating the integrals using quadrature formulas of the required order[7] which is $2p$ for the volume integral and $2p + 1$ for the edge integral. Although this approach is straightforward, it limits the usefulness of the discontinuous Galerkin method. Tensor products of one-dimensional quadrature formulas can be used to integrate quadrilateral and hexahedral elements to any required degree; however, the number of terms in the quadrature summation exceeds the number of unknowns by a considerable margin (e.g. 3.5 times greater for $p = 4$ in three dimensions). Aside from Dudiner's[8] approach in which triangles and

2

tetrahedrons are mapped into quadrilaterals and hexahedrons, there are no general quadrature formulas for elements such as triangles or tetrahedrons. For these general elements, the only available quadratures are those that have been computed numerically and tabulated for a limited range of p. This has restricted most implementations of the discontinuous Galerkin method to quadrilateral, hexahedral, or relatively low order triangular elements. Furthermore, the implementations are computationally expensive. The authors are not aware of any discontinuous Galerkin implementations using tetrahedral elements.

The quadrature-free approach[3] was developed to circumvent this difficulty and to allow the discontinuous Galerkin method to be easily and efficiently implemented on general unstructured grids to any order of accuracy. To implement the quadrature-free approach, the fluxes and sources are also written as an expansion in terms of the basis functions:

$$\vec{\mathbf{F}}(\mathbf{Q}) \approx \sum_{j=1}^{M} \vec{\mathbf{f}}_j(\mathbf{V}) b_j, \qquad \mathbf{S} \approx \sum_{j=1}^{M} \mathbf{s}_j b_j$$

When $\vec{\mathbf{F}}(\mathbf{Q})$ is a linear function of $\mathbf{Q}$, then $M = N$, and the expansion is trivial and exact. When the flux is a non-linear or a linear but non-constant coefficient function of $\mathbf{Q}$, then the degree of the flux expansion must be at least $p + 1$ and M will be greater than N. The same comment applies to the source term $\mathbf{S}$ except that the source expansion may be truncated to degree p.

Similar treatment of the approximate Riemann flux is only slightly more complex due to the fact that the solutions on either side of an interior edge, $\mathbf{V}$ and $\mathbf{W}$, are defined in terms different coordinate systems. The evaluation of the edge integral is simplified by rewriting the edge trace of $\mathbf{V}$ and $\mathbf{W}$ in terms of a lower dimensional $(\bar{\xi}, \bar{\eta})$ coordinate system associated with the edge: $\bar{\mathbf{V}} = \sum_{k=1}^{N} \bar{\mathbf{v}}_k \bar{b}_k$, and $\bar{\mathbf{W}} = \sum_{k=1}^{N} \bar{\mathbf{w}}_k \bar{b}_k$. $\bar{b}_k$ denotes a lower-dimensional basis associated with the edge coordinate system. This, of coarse, is just a coordinate transformation, and the coefficients $\bar{\mathbf{v}}$ are easily computed from $\mathbf{v}$ by a linear matrix operator $[\bar{\mathbf{v}}] = \mathbf{T}[\mathbf{v}]$. The trace of the flux can be computed either by taking the trace of the volume flux, $[\bar{\mathbf{f}}] = \mathbf{T}\left[\mathbf{J}^{-1}J\vec{\mathbf{f}} \cdot \vec{n}\right]$, or by recomputing the flux from the trace of the solution. Generally, the later is prefered for a linear problem. Now the approximate Riemann flux can be expanded in terms of $\bar{b}_k$ as

$$\begin{aligned} \mathbf{F}_e^R &\equiv \sum_{k=1}^{M} \mathbf{f}_{e,k}^R \bar{b}_k \\ &= \sum_{k=1}^{M} \left[\bar{\mathbf{f}}_{e,k}^R(\bar{\mathbf{v}}) + \bar{\mathbf{f}}_{e,k}^R(\bar{\mathbf{w}}) - \lambda\left(\bar{\mathbf{w}} - \bar{\mathbf{v}}\right)\right] \bar{b}_k / 2 \end{aligned}$$

without regard for the type of element or the orientation of the coordinate system of the adjacent elements.

As illustrated in figure 1, this approach allows elements of different type to be freely mixed in a calculation.
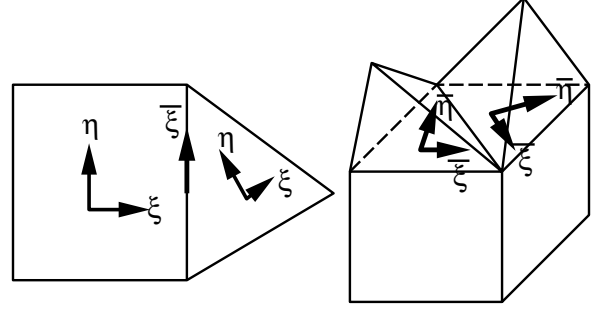

Fig. 1   Edge coordinate system $(\bar{\xi}, \bar{\eta})$ provides a common interface between dissimilar elements.

Now that the functional form of the solution, source, and fluxes is explicit, the integrals are analytically evaluated to give

$$\frac{\partial [\mathbf{v}_k]}{\partial t} + \vec{\mathbf{A}} \cdot \mathbf{J}^{-1} J \left[\vec{\mathbf{f}}_j\right] + \sum_{e=1}^{E} \mathbf{B}_e \left[\mathbf{f}_{e,j}^R\right] = [\mathbf{s}_j] . \quad (4)$$

where

$$\vec{\mathbf{A}} = \mathbf{M}^{-1}\left[\int_{\Omega} b_j \nabla b_k d\Omega\right], \qquad \mathbf{B} = \mathbf{M}^{-1}\left[\int_{\partial\Omega} b_k \bar{b}_j ds\right],$$

$$\text{and} \quad \mathbf{M} = \left[\int_{\Omega} b_k b_j d\Omega\right].$$

The matrices $\mathbf{M}$, $\vec{\mathbf{A}}$, and $\mathbf{B}$, depend only on the shape of the similarity element and the degree of the solution p. Thus, the set of matrices associated with a particular similarity element can be precomputed and applied to all elements that map to it at a considerable savings of both storage and computation. Finally, as a result of symmetry in the similarity elements, the matrices $\vec{\mathbf{A}}$, $\mathbf{B}_e$, and $\mathbf{T}_e$, are sparse in a predictable manner that is taken advantage of readily.

Arbitrary, flat-sided, triangular and tetrahedral elements and a subset of other element types can be linearly mapped into a small set of similarity elements. A boundary element in which one or more sides are curved is a case in which the physical element cannot be linearly mapped to a similarity element. Illustrated in figure 2 for a triangle, the physical element can still be linearly mapped to a computational element, but now its curved sides are described by a polynomial. The integrals in equation (4) can still be evaluated analytically; however, the matrices are distinct for each element, and in general, the matrices are full. This wall treatment allows the elements to be sized as required to resolve acoustic waves and not be restricted by the need to resolve wall curvature. Experience thus far indicates that only a few such elements are required in most cases, so the additional storage and computation does not impact the overall calculation.
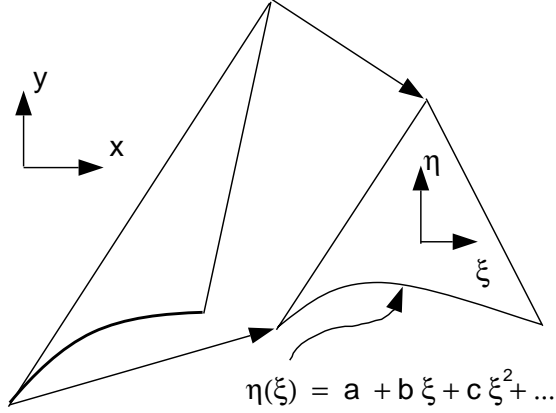
Fig. 2   Mapping for curved wall element.

$$\eta(\xi) = a + b\,\xi + c\,\xi^2 + \dots$$

p	Maximum $\Delta t/\Delta x$		
	Hu scheme	5-stage scheme	6 stage scheme
1	1.70	1.61	1.77
2	0.567	0.543	0.572
3	0.287	0.271	0.300
4	0.177	0.167	0.185
5	0.122	0.106	0.127
6	0.089	0.073	0.930

Table 1   Stability constraints for 5 and 6 stage linear Runge-Kutta schemes, and Hu's[10] low-dispersion Runge-Kutta scheme.

The solution is advanced in time using a Runge-Kutta scheme. Earlier work used the three-stage third-order TVD Runge-Kutta scheme of Shu and Osher.[9] A stability analysis establishing the the time step constraints has been previously published.[3] The low-dissipation and low-dispersion Runge-Kutta scheme of Hu[10] is used in the present work. This scheme provides fourth-order accuracy by an alternating sequence of 5- and 6-stage Runge-Kutta schemes with modified lower-order coefficients. A stability analysis of this scheme applied to a one-dimensional discontinuous Galerkin method, given in table 1, indicates that the time step constraint falls between that of 5- and 6-stage linear Runge-Kutta schemes.

Physical Modeling

The scatter of acoustic waves is well represented by the linearized Euler equations. In two-dimensions, the $\mathbf{Q}$, $\vec{\mathbf{F}}$, and $\mathbf{S}$ of equation (1) are given by

$$\mathbf{Q} = \begin{pmatrix} \rho \\ p \\ u \\ v \end{pmatrix}, \quad \mathbf{F}_1 = \begin{pmatrix} \overline{U} & 0 & \overline{\rho} & 0 \\ 0 & \overline{U} & \gamma\overline{P} & 0 \\ 0 & 1/\overline{\rho} & \overline{U} & 0 \\ 0 & 0 & 0 & \overline{U} \end{pmatrix} \mathbf{Q},$$

$$\mathbf{F}_2 = \begin{pmatrix} \overline{V} & 0 & 0 & \overline{\rho} \\ 0 & \overline{V} & 0 & \gamma\overline{P} \\ 0 & 0 & \overline{V} & 0 \\ 0 & 1/\overline{\rho} & 0 & \overline{V} \end{pmatrix} \mathbf{Q},$$

$$\mathbf{S} = \begin{pmatrix} 0 & 0 \\ 0 & (1-\gamma)(\overline{U}_x + \overline{V}_y) \\ -(\overline{UU}_x + \overline{VU}_y)/\overline{\rho} & 1/\overline{\rho}_x \\ -(\overline{UV}_x + \overline{VV}_y)/\overline{\rho} & 1/\overline{\rho}_y \\ 0 & 0 \\ (\gamma-1)\overline{P}_x & (\gamma-1)\overline{P}_y \\ \overline{V}_y & -\overline{U}_y \\ -\overline{V}_x & \overline{U}_x \end{pmatrix} \mathbf{Q}. \tag{5}$$

An overline has been used to denote local temporal-mean quantities, and subscripted values denote differentiation. ρ and p are the density and pressure, and u and v are the x and y directed velocities, respectively. γ is the ratio of specific heats. The equations have been made dimensionless using the ambient speed of sound c_o as the reference velocity. The density and the pressure have been normalized by ρ_o and $\rho_o c_o^2$, respectively. Time t is normalized by l_r/c_o where the reference length scale l_r is case specific. In the absence of a mean flow, an acoustic wave will propagate a unit in the non-dimensional distance in each unit of non-dimensional time.

When the mean flow quantities are spatially constant, the flux expansion in terms of the basis functions b_k is a trivial operation. However, when the mean flow quantities are non-uniform, their spatial variation must be accounted for in the flux expansion. A general approach is to approximate the mean flow by polynomials and then derive the flux expansion in terms of polynomial products. To ensure the formal order properties of the discontinuous Galerkin method, the mean flow quantities should be represented to the same degree as the solution (degree p). The polynomial products can be truncated to degree $p + 1$. However, this general approach requires considerable storage for the mean flow quantities. Fortunately, in wave scatter simulations, the assumption of linearity is based on the premise that the perturbation amplitude is small relative to its wavelength (to neglect non-linear steepening), and the wavelength is small relative to variations in the mean flow (to neglect feedback from the acoustics to the mean flow). Thus, for the purposes of simulating the scatter of acoustic waves, it is sufficient to represent the mean flow by a lower-order polynomial such as a piecewise linear representation.

Results

Parallel Computations

The program has been ported to parallel computing platforms using MPI calls. The initial port was performed on the two-dimensional version of the code. The port required only minimal changes to the code and was performed in only a few weeks.[11] Aside from the mesh partitioning step, the port involved creating and inserting three functions containing about 120 lines of code: `InitPid`, `BeginSendRecv`, and

4

EndSendRecv. The InitPid function allocates the send and receive buffers and initializes domain connectivity data structures. The BeginSendRecv function loads the send buffer and posts both the MPI sends and receives, and the EndSendRecv synchronizes the calculation. The latter two functions are simply inserted into the Runge-Kutta scheme at the appropriate points. The three-dimensional version of the code, which was created sometime later, ran in parallel with no changes to the above three functions. The speed and flexibility of the port is attributed to the modular nature of the code and the language in which it is written (C++).

In the two-dimensional case, the mesh is partitioned using the PARMETIS[12] software package. The three-dimensional results are preliminary, and the mesh is partitioned simply by distributing the list of elements as they are read in from a mesh file.

Figure 3 gives the speed-up for the SP2 and clusters of workstations for a small case (800 third-order triangular elements). As the domain is divided over more processors, the number of elements assigned to each processor becomes too small, and the communication overhead becomes apparent. Also, the workstations were not isolated or dedicated, so the performance figures are well below ideal; however, four to eight distributed workstations still provide a valuable performance improvement.
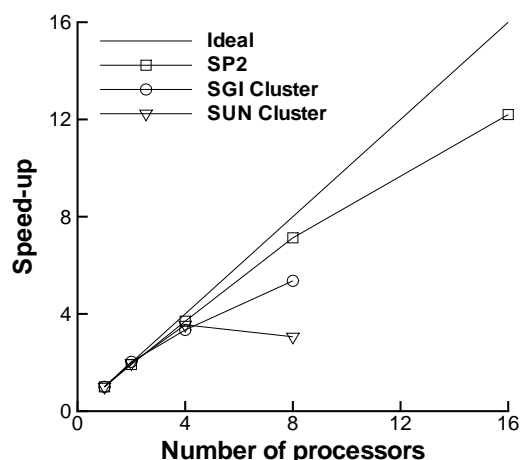
Fig. 3 Speed-up on SP2 and workstations for a domain with 800 elements.

Figure 4 gives the speed-up on the Origin 2000 (O2K) using an improved version of the program that has been optimized for cache usage on SGI machines. The parallel implementation has not been modified except for the partitioning and initialization procedures. Results are given for two domain sizes using fifth-order triangular elements ($p = 4$). The superlinear speed-up observed on the O2K is attributed to the cache performance of the code. As the number of processors is increased, the domain is broken into smaller parts

that such that, at some point, all of the data fits in cache. This is clearly evident in figure 5 which shows the computational rate as a function of the number of elements on each processor. For comparison, the figure also shows results for a range of problem sizes run on a single processor. Although there is a consistent degradation in performance between the single-processor and multi-processor cases, the computational rate of the multi-processor cases are similar indicating good scalability. Thus, computational rate may be a better indicator of scalability then the usual "speedup" measure. The scatter in the data gives an indication as to the quality of the load balancing provided by the PARMETIS mesh partitioner.
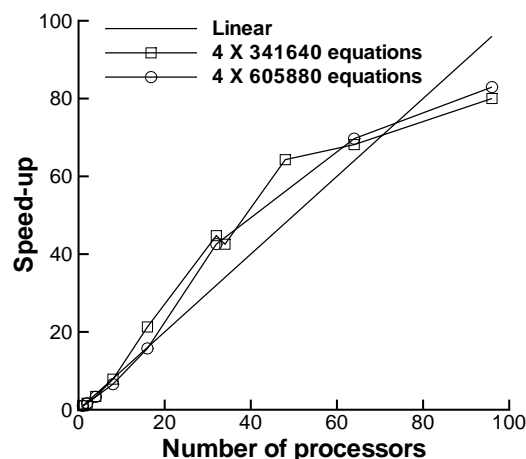

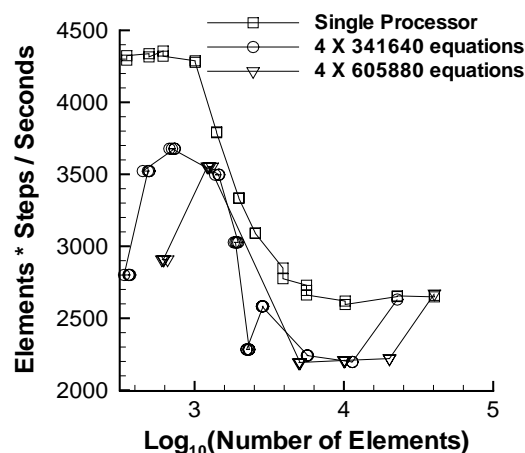Fig. 4 Speed-up for large domain on Origin 2000.

Fig. 5 Computational rate gives a clear indication of the cache performance.

Preliminary performance results for the three-dimensional code are shown in figure 6 for two large domains. The performance is encouraging considering the simplistic method by which the mesh is partitioned.
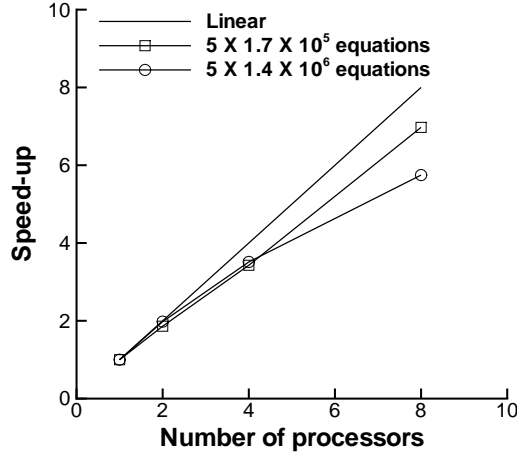
Fig. 6   Speed-up of three-dimensional code on Origin 2000.

Fig. 7   Instantaneous u velocity contours for a temporal instability.

Nonuniform Mean Flows

To validate the code for nonuniform mean flows, the growth rate of a temporal instability in a shear layer is calculated. The mean velocity profile is given by $M(1 + \tanh(y))/2$ where the maximum Mach number is $M = 0.2$. The computational domain is comprised of triangles and extends from -4.25 to 4.25 in y and from -10 to 10 in x. The triangular mesh is generated by dividing cells of a Cartesian mesh with equal spacing in x and sinh stretching in y. All of the calculations are performed with degree $p = 4$ polynomials which provides fifth-order accuracy. The spatial scales of the mean flow and the instability wave are similar so it is also necessary to approximate the mean flow with degree four polynomials. The eigenfunction for a particular instability mode, calculated by the stability code developed by Macaraeg et al.,[13] is used to provide initial profiles for all of the variables. Figure 7 shows instantaneous contours for the u velocity after several periods of oscillation. During the simulation, the pressure is sampled at ten points in the vicinity of the shear layer. Figure 8 shows the evolution of the pressure at the center of the shear layer over four periods. The growth rate from the stability code is used to determine the amplitude envelope which compares very well with the computed solution. The average growth rate of the pressure over the second two periods is compared with the result from the stability code in figure 9. As the mesh is refined, the solution quickly attains the value from the stability code. A second mesh refinement is performed by randomly perturbing the initial mesh by 20% in both the x and y directions at each triangle vertex. Except for the coarsest meshes, the solution on the perturbed mesh is nearly as good as that on the original mesh.

Deployed Slat

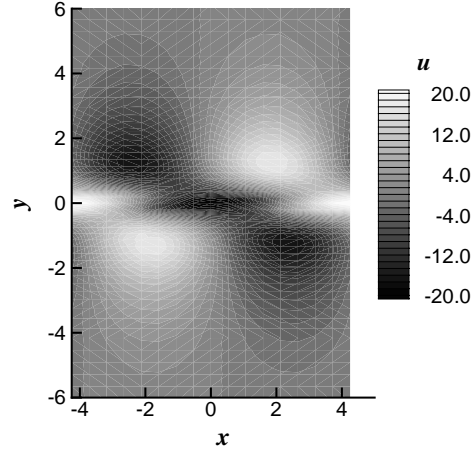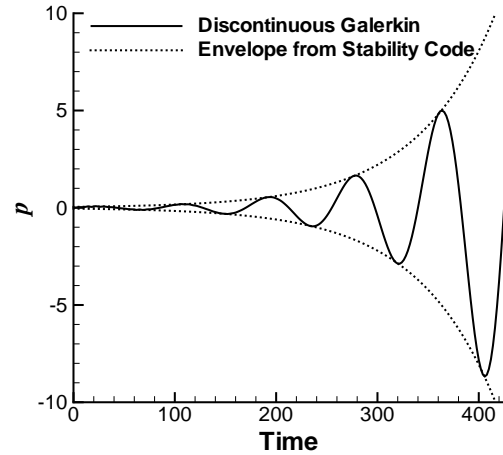Computations are being performed on the scattering

Fig. 8   Temporal evolution of the pressure at the center of the shear layer for a temporal instability.

and resonance characteristics of a slat geometry using the discontinuous Galerkin method. An example grid is shown in figure 10. The actual grid used in the simulations has 6.2 times as many elements. Acoustic sources at various frequencies and with different orientations have been placed at several locations around the slat. The region of primary interest is the trailing edge of the slat where unsteady Reynolds-averaged Navier-Stokes (RANS) calculations[14] have shown significant vortex shedding. Figures 11 and 12 show the scattered fields for dipole sources oriented normal and parallel to the slat edge. A frequency of 50 kHz is used that corresponds to the dominant frequency observed in the RANS calculations. There is significant beaming and a strong dependence on the source orientation.

These calculations can be performed at a rate of 62 seconds/period using 6 O2K processors and 32 seconds/period on 10 O2K processors. With this level of performance, it is practical to rapidly investigate the
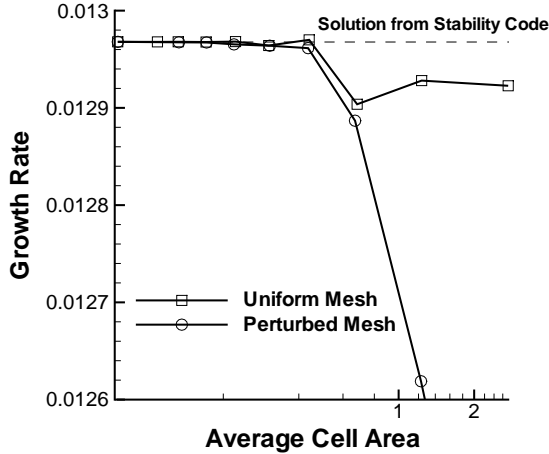
Fig. 9   Growth rate comparison for a temporal instability using uniform and perturbed meshes.

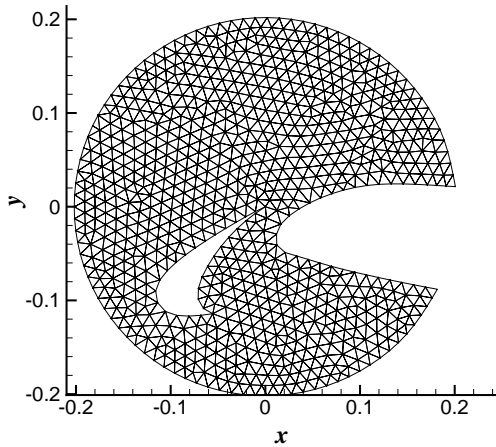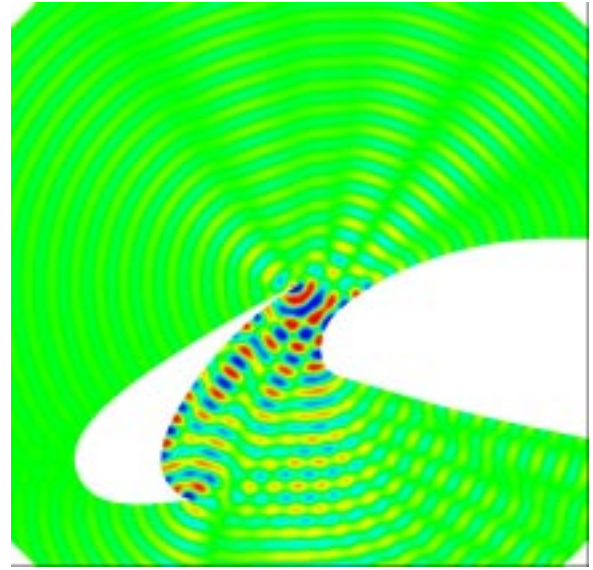effects of variations in the acoustic source and geometry.

Fig. 11   Scattering of a dipole source at the trailing edge of the slat. The source is oriented normal to the streamlines at the trailing edge.

Fig. 10   Example mesh for a slat geometry.

Three-Dimensional Simulations

Three dimensional mesh refinement studies are performed to verify that the code produces the expected convergence rate as well as to determine the absolute levels of error one can expect for a given resolution. With methods for structured grids, the latter studies can be performed exclusively in one-dimension with reasonable expectations that the same error behavior will be observed in two or three dimensions. When using unstructured grids, however, the topology of the element changes dramatically, and it is not at all clear that the studies in one dimension will carry over to two or three dimensions. All of the three-dimensional results shown here are for a fifth-order method ($p = 4$).

The mesh refinement tests are performed on a unit cube using the linear Euler equations with no mean flow, and an initial solution of the form $p = \cos(2\pi(x + y))$, and $u = v = p/\sqrt{2}$. A degree four polynomial
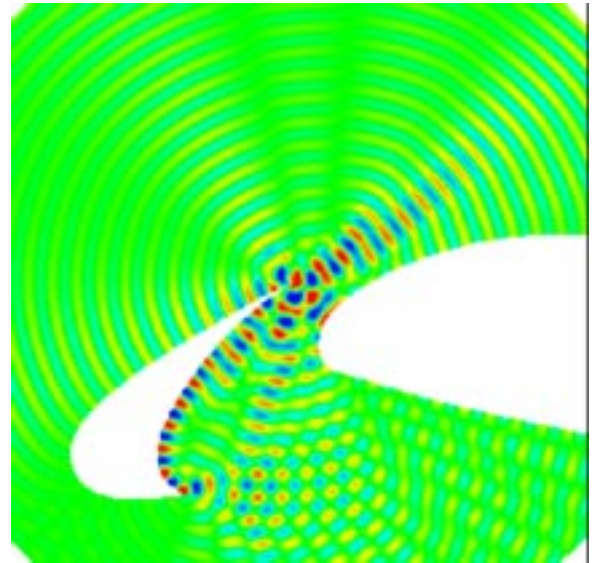

Fig. 12   Scattering of a dipole source at the trailing edge of the slat. The source is oriented parallel to the streamlines at the trailing edge.

approximation to the initial solution is obtained by fitting a polynomial at a collection of points within each element. A tetrahedral mesh is constructed from an $N_x \times N_y \times N_z$ Cartesian mesh in which each cube is subdivided into five tetrahedrons (N_x denotes the number of elements in the x direction). With this construction, the distance between vertices varies by a factor of $\sqrt{2}$. The initial test propagates the wave for 10 periods on an $N \times N \times N$ mesh for $N = 2, 4$, and 8. A second test propagates the wave for 100 periods on an $N \times N \times 2$ mesh for $N = 2, 4$, and 8. The reported error is the L_2 norm of the local error in the solution sampled at 2500 points in the domain. Figure 13(a) gives the convergence rate at times = 1, 10, and 100. Between the

AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS PAPER

two finer grids ($N = 4$ and 8), all cases converge at a rate of about 5.5. Figure 13(b) gives the growth in the error as a function of time, which is equivalent to distance propagated. With as little at two elements per wavelength the error is still less than 3% at time $= 100$.
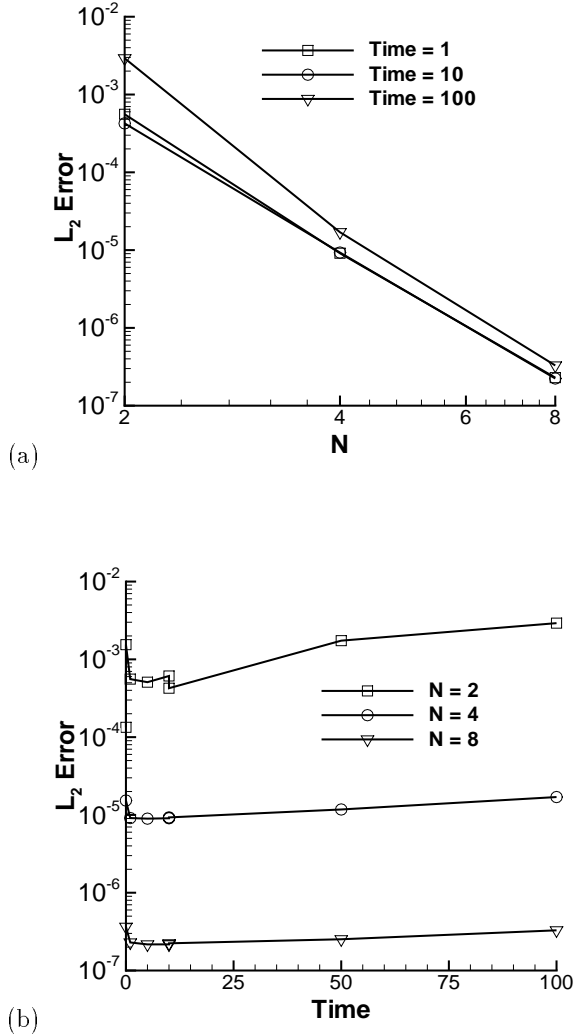

(a)

(b)

Fig. 13 Propagation error as a function of (a) mesh size, (b) distance propagated.

It has been claimed that the discontinuous Galerkin method is insensitive to mesh smoothness. This property has been demonstrated for two-dimensional scalar advection[4] and for a two dimensional shear layer earlier. It is demonstrated here for three-dimensional wave propagation. A Gaussian pressure pulse centered $(x, y, z) = (0.75, 0.5, 0.5)$ generates a spherical acoustic wave within a unit cube. The mean flow is uniform but with $\bar{U} = 0.5$, $\bar{V} = \bar{W} = 0$. A tetrahedral mesh with 6000 elements is generated from a $10x10x10$ Cartesian mesh in which each cube is divided into six tetrahedrons. The solution is approximated with degree four polynomials. Figure 14(a) shows the perturbation pressure on three planes of the box at time $=$

0.25. Next, each grid point is randomly perturbed by 20% of the mean mesh size and the case is rerun. Figure 14(b) shows very good agreement between the two solutions on the line $y = z = 0.5$. The uniform case is shown as a solid line, and the case with the randomly perturbed mesh is shown with symbols. Contour plots of the solutions are indistinguishable.
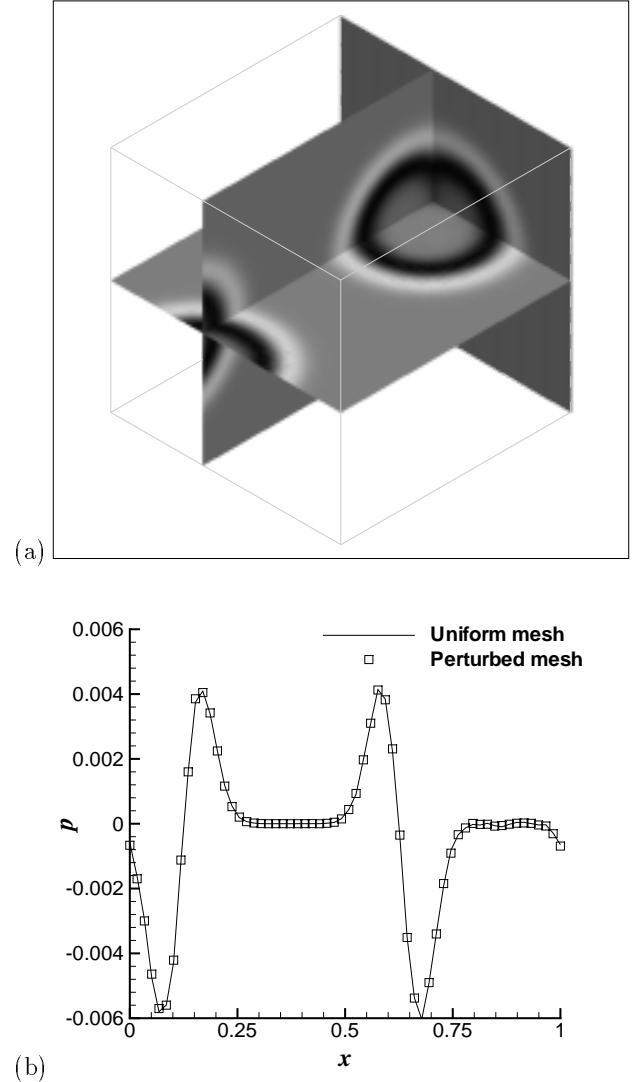

(a)

(b)

Fig. 14 Perturbation pressure of an acoustic wave initiated by a Gaussian pulse. a) solution from uniform mesh on three planes b) solution on line $y = z = 0.5$ with smooth and randomly perturbed meshes.

Finally, a computation is presented on a generic blended-wing-body configuration in which the mesh is generated by commonly available tetrahedral mesh generation software. Figure 15 shows the surface mesh and several planes and a line on which results are shown. The aircraft has a length of 150 ft. and a simi-span of 145 ft. to be representative of a configuration currently under study by NASA and industry. The outer boundary of the computational domain is

a box defined by $-50 < x < 200$, $-200 < y < 0$, and $-50 < z < 50$. The geometry and surface
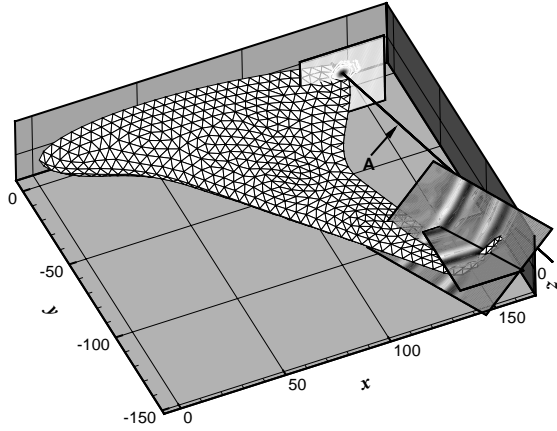

Fig. 15 Generic blended-wing-body configuration with planes and lines on which data is examined.

mesh are generated using the commercial software MSC/Patran, and volume meshes are generated using VGRID[15] and AFLR3.[16] These volume meshers were developed for aerodynamic performance calculations in which the mesh size is expected to grow away from the body. Both meshers have difficulty generating meshes with uniformly sized tetrahedrons. To ease the problem, the aircraft surface is meshed with an average edge length of five, and the bounding box is meshed with an average edge length of seven. The results shown here are for a mesh with 78048 tetrahedrons generated by AFLR3. In general, meshes generated by AFLR3 are more uniform; however even the mesh used here has volumes ranging from 2.2 to 243.

An initial pressure disturbance of the form

$$p = \begin{cases} \cos^2(\pi R/8) & \text{for } R \le 4 \\ 0 & \text{for } R > 4 \end{cases},$$

where $R = \sqrt{((x-140)^2 + (y+10)^2 + (z-5)^2)}$, creates a transient wave that is propagated to a time of 160. The calculation required approximately 48 hours using eight processors of an Origin 2000. Based on the parallel performance shown earlier, it is reasonable to expect that such a calculation can be performed in only a few hours on a large array of 80 to 100 processors.

Figure 16(a) shows the initial evolution of this pulse along line "A" shown in figure 15. The pulse should to decay at a rate proportional to $1/R$. Figure 16(b) shows the solution along line "A" at time = 50, 100, and 150 with an appropriate scaling. The final two sample times are in close agreement which indicates the wave is decaying at the correct rate. Figures 17(a-c) show the pressure on a horizontal cut through the winglet at time = 130, 140, and 150. Figures 18(a-c) show a similar set of solutions on a spanwise cut
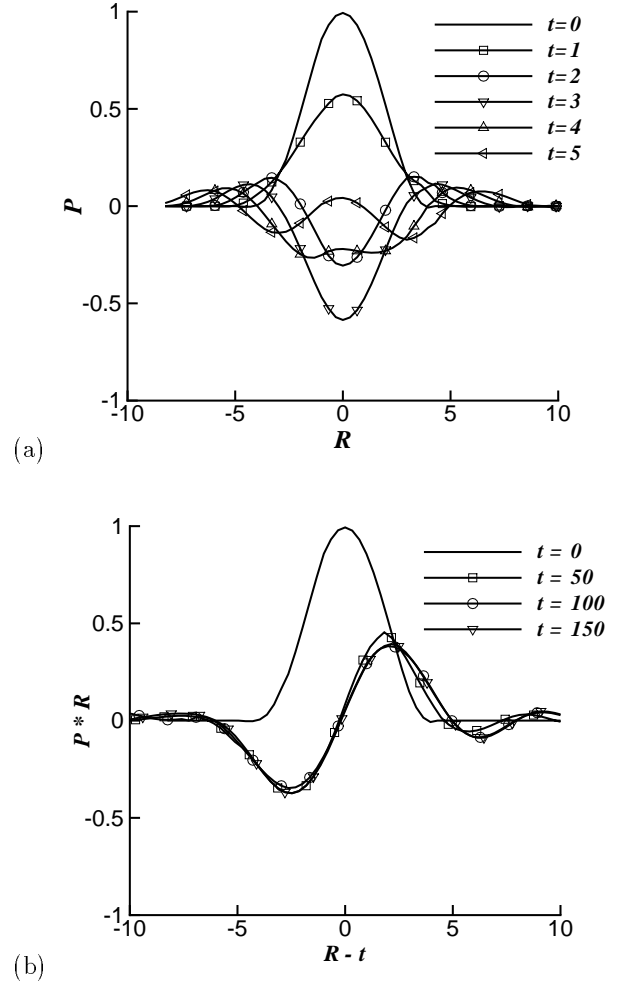

(a)

(b)

Fig. 16 Perturbation pressure along line "A" shown in figure 15: (a) initial transient (b) later solutions scaled to show the $1/r$ decay.

through the wing and winglet. Though qualitative, these figures show the propagation and reflection of waves in the expected manner.

Conclusions

Several improvements to the high-order discontinuous Galerkin program have been validated. The capability to treat non-uniform mean flows was verified by comparison with an instability wave on both smooth and random meshes. The accuracy of the solution is not sensitive to the smoothness of the mesh. The compact form of the discontinuous Galerkin method allows easy and efficient ports to parallel platforms. The parallel implementation gives superlinear speed-up for large problems. This is attributed to the compact form of the discontinuous Galerkin method which allows useful work to be overlapped with communication as well as cache accelerations that occur when a large problem is broken into smaller components. The extension to three dimensions has been performed and

verified for smooth and non-smooth meshes. Solutions converge to the exact solutions at the expected rate, and the accuracy is not greatly effected by mesh smoothness. Results are presented establishing the absolute levels of error one can expect for a given resolution and distance of propagation when using tetrahedral elements. It is demonstrated that the three-dimensional fifth-order method can propagate a wave for 100 wavelengths with less than 3% error while using only two elements per wavelength. Finally, application of the program to the analysis of a two-dimensional slat and a generic blended-wing-body configuration demonstrates that the method and program can rapidly provide analysis of real geometries.

References

[1] Johnson, C. and Pitkärata, J., "An Analysis of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation," *Mathematics of Computation*, Vol. 46, No. 176, 1986, pp. 1–26.

[2] Cockburn, B. and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework," *Mathematics of Computation*, Vol. 52, No. 186, 1989, pp. 411–435.

[3] Atkins, H. L. and Shu, C. W., "Quadrature-Free Implementation of Discontinuous Galerkin Method for Hyperbolic Equations," *AIAA Journal*, Vol. 36, No. 5, 1997, pp. 775–782.

[4] Atkins, H. L., "Continued Development of the Discontinuous Galerkin Method for Computational Aeroacoustic Applications," AIAA Paper-97-1581, 1997, Third Joint CEAS/AIAA Aeroacoustics Conference, May 12-14.

[5] Atkins, H. L., "Local Analysis of Shock Capturing Using Discontinuous Galerkin Methodology," AIAA Paper 97-2032, 1997, 13th AIAA Computational Fluid Dynamics Conference, Snowmass Village, Colorado, June 29-July 2.

[6] Tam, C. K. W. and Hardin, J. C., editors, *Second Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*. NASA CP-3352, 1997, Proceedings of a workshop sponsored by NASA and Florida State University in Tallahassee, Florida on Nov. 4-5, 1996.

[7] Cockburn, B., Hou, S., and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws IV: The MultiDimensional Case," *Mathematics of Computation*, Vol. 54, No. 190, 1990, pp. 545–581.

[8] Dubiner, M., "Spectral Methods on Triangles and Other Domains," *Journal of Scientific Computing*, Vol. 6, 1991, pp. 345.

[9] Shu, C.-W. and Osher, S., "Efficient Implementation of Esentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 77, No. 77, 1988, pp. 439–471.

[10] Hu, F. Q., Hussaini, M. Y., and Manthey, J., "Low-dissipation and -dispersion Runge-Kutta schemes for Computational aeroacoustics," *Journal of Computational Physics*, Vol. 124, No. 1, 1995, pp. 177–191.

[11] Atkins, H. L., Baggag, A., Ozturan, C., and Keyes, D., "Parallelization of an Object-Oriented Unstructured Aeroacoustics Solver," ICASE Report No. 99-11, 1999, Ninth SIAM Conference on Parallel Processing for Scientific Computing, San Antonio, TX, March 22-24.

[12] Karypis, G. and Kumar, V., "Parallel multilevel k-way partitioning scheme for irregular graphs," Technical Report 96-036, Department of Computer Science, University of Minnesota, 1996.

[13] Macaraeg, M. G., Streett, C. L., and Hussaini, M. Y., "A spectral collocation solution to the compressible stability eigenvalue problem," NASA-TP-2858, 1988.

[14] Khorrami, M. D., Berkman, M. E., Choudharri, M., Singer, B. A., Lockard, D. P., and Brentner, K. S., "Unsteady flow compuations of a slat with a blunt trailing edge," AIAA Paper 99-1805, 1999, Presented at the 5th AIAA/CEAS Aeroacoustics Conference, Seattle, WA, May 10-12.

[15] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by Advancing-Layer Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.

[16] Marcum, D. L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper-95-0212, 1995.
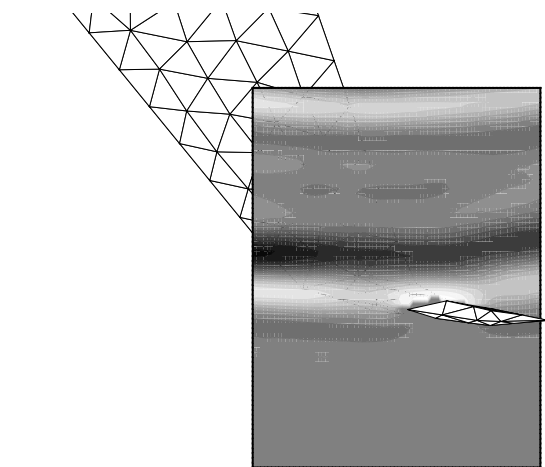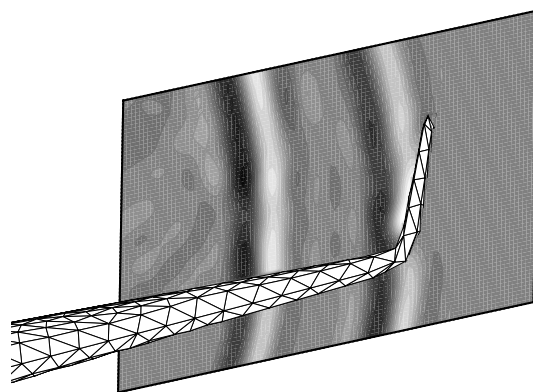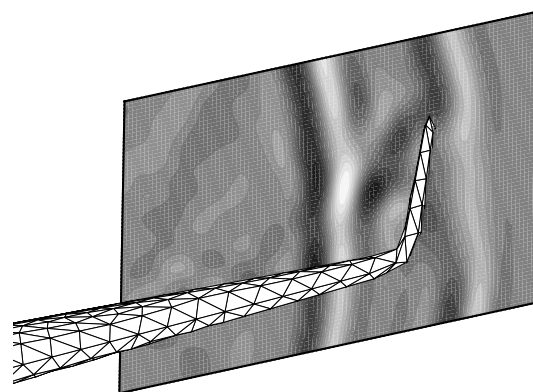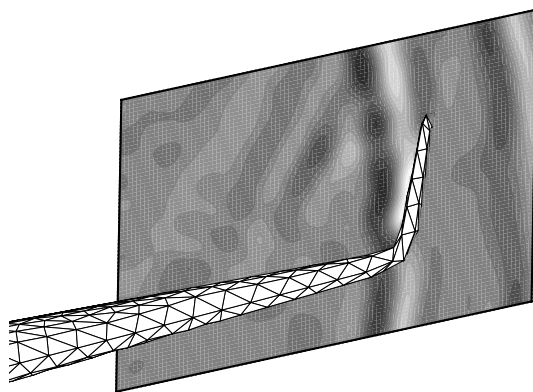
(a)
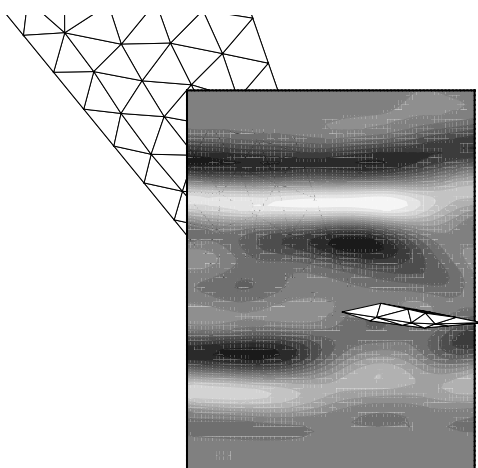
(b)
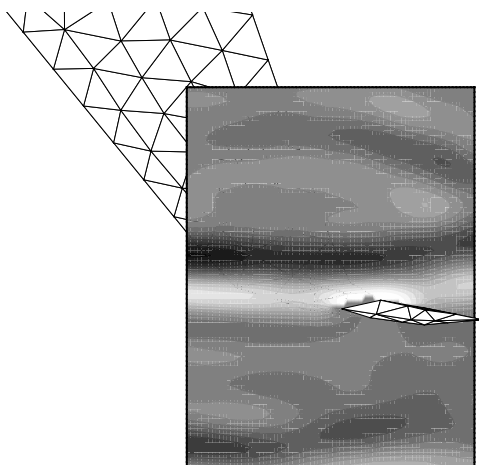
(c)

Fig. 17 Perturbation pressure on a horizontal cut through the winglet at time = (a) 130, (b) 140, (c) 150.

(a)

(b)

(c)

Fig. 18 Perturbation pressure on a spanwise cut through the wing and winglet at time = (a) 130, (b) 140, (c) 150.