AIAA JOURNAL
Vol. 37, No. 11, November 1999

Steady and Unsteady Flow Simulations Using the Hybrid Flow Solver AVBP

Thilo Schönfeld*
CERFACS, F-31057 Toulouse CEDEX 1, France
and
Michael Rudgyard†
Oxford University, Oxford, England OX1 3QD, United Kingdom

The unstructured flow solver AVBP of CERFACS is presented. The basic concepts of the program are described, and various computational results are presented for a large spectrum of applications ranging from steady-state external aerodynamics to unsteady turbulent flows with and without combustion. The code solves the compressible Navier–Stokes equations on hybrid grids of arbitrary cell type. The code is built on a modular software library and has been ported to a wide range of parallel computers.

Introduction

COMPUTATIONAL fluid dynamics (CFD) has reached maturity and today is increasingly incorporated into the industrial design process. However, (notably) aircraft manufactures claim that achieving the goal of a 24-h turnaround production cycle is no longer predominantly limited by hardware resources, but by the software related to numerical methods and the generation of computational grids. Many of the existing workhorse CFD codes were initially designed to provide solutions of high accuracy for laminar and turbulent stationary flows in configurations of limited complexity. For these limited requirements, structured multiblock codes provide satisfactory results, although gridding, for example, a wing–fuselage–nacelle configuration, may result in over 60–70 blocks of nonhomogeneous sizes. However, the current demand for solving unsteady flows for geometries of ever-increasing complexity is beyond the capabilities of purely structured methods. Chimera grid methods with overlapping domain interfaces seem to offer a promising remedy, but this approach is often rejected due to difficulties in defining conservative schemes. The unstructured grid methodology offers high flexibility but is potentially less accurate and requires higher computational costs per element when compared to structured methods. As a result, many national and multinational projects have currently undertaken to merge several codes and to create CFD libraries, with the aim to build CFD tools of high flexibility, efficiency, and modularity.

With these aims in mind, the AVBP project was started at the beginning of 1993 to create an unstructured solver capable of handling grids of any type of cell. The use of hybrid grids is motivated by the efficiency of unstructured grid generation, the accuracy of the final results (using regular structured elements), and the ability to use mesh adaptation. The philosophy of building a software library was adopted to best meet the requirement of modularity.

At present, AVBP is widely used by members of the CFD project at CERFACS, both for basic research and contractual work. Recently, the first external releases have been installed. The code is considered as a research code inasmuch as industrialization has yet to be undertaken. The growing activities in the field of unsteady (turbulent) flows motivate the presentation of the recent progress made on AVBP and the demonstration of the future potential of the code. In this paper we first present the basic concepts of the code, followed by a description of the numerical method that is designed for hybrid grids; we then show various examples of computational results to indicate the current capabilities of AVBP.

Other CFD solvers with similar three-dimensional hybrid grid capabilities are, for example, the solvers developed at University of Austin,[1,2] the SAUNA system of ARA in the United Kingdom,[3] the τ-code of the DLR, German Aerospace Research Center,[4] and commercial packages such as the HYBRID code of Amtec Engineering.[5]

Description of the Code

Basic Concepts

AVBP solves the laminar and turbulent compressible Navier–Stokes equations in two and three space dimensions. Steady-state or unsteady flows may be simulated. For stationary turbulent flows, two different turbulence models are available, whereas for the prediction of unsteady turbulent flows, large eddy simulation (LES) or direct numerical simulation (DNS) may be employed.

The capability to handle structured, unstructured, or hybrid grids is one key feature of AVBP. With the use of these hybrid grids, where a combination of several elements of different types is used within the same mesh, we aim to combine the advantages of the structured and unstructured grid methodologies. Although structured multiblock methods are generally more efficient for a given solution accuracy when compared to unstructured grid methods, they often need an impractical number of grid points because of the topology of hexahedrons or because they usually only permit length scale variations in one direction at a time. Structured methods are also less amenable to grid adaptation. Unstructured methods are distinguished by their flexibility for handling complicated geometries and their ability to permit varying length scales in all three directions independently. Drawbacks include the higher computational cost per element and potentially lower accuracy due to skewed elements. The use of hybrid grids, where cells of different types are used in the same mesh, offers a release from the constraints of structured and unstructured methods. Thus, one seeks a method that uses mostly hexahedral elements but that offers the flexibility of unstructured grids. Note that by unstructured we only refer to the data connectivity but not the element types; hence, quadrilaterals or hexahedrons can occur in an unstructured framework.

As a second key feature AVBP is built on the modular software library COUPL.[6] This subroutine library aims to free the nonspecialist user from the need to consider aspects of high-performance computing. COUPL uses a data parallel strategy, includes integrated domain partition and data reordering tools, handles message passing, and includes supporting routines for dynamic memory allocation, as well as routines for parallel I/O and iterative methods. The library makes use of a generalized data structure suitable for unstructured meshes of arbitrary elements. It aims to be portable and efficient on scalar and parallel computers.

Received 5 August 1997; revision received 5 August 1998; accepted for publication 28 April 1999.

*Permanent Research Scientist, Computational Fluid Dynamics Group, 42 Avenue Gaspard Coriolis. Member AIAA.

†Smith Institute Research Fellow, Computing Laboratory, Wolfson Building, Parks Road.

Data Structure

A generalized data structure allows us to consider both unstructured and structured (multiblock) meshes in two or three dimensions. We assume that the solution variables are stored on the nodes of the mesh and define generalized elements that may be real grid cells, such as tetrahedrons, prisms, pyramids or hexahedrons, or cell faces or edges. In the present discussion we consider cell-based algorithms, which are well suited for cell-vertex finite volume or finite element methods. Each type of generalized element has a number of vertices, and pointer arrays indicate the relationship between the element and the nodes, that is, cell-to-node connectivity. Gather and scatter techniques are then used throughout to pass information between the elements and the nodes.

The use of indirect addressing results in a loss of computational efficiency compared with structured grids methods. Because additional pointers are necessary for providing connectivity information, this also leads to increased memory requirements. To a large extent, these disadvantages are compensated for by the requirements for parallelizing unstructured methods. Because the nodal scatter operations impede optimal vectorization, a coloring option allows us to group the cells such that no cells with a common node belong to the same group.

Because a large part of the work in constructing most numerical discretizations takes place at the level of the elements, we have used a so-called grouping strategy. The element arrays are reordered into chunks of small groups that are typically composed of 80–100 elements: The number of elements is chosen to minimize memory requirements and to optimize efficient cache usage. The solver then operates on these groups of elements after gathering information from the nodes and then scattering the results back. For hybrid grids, each group contains a fixed type of element, and all groups of the same type within each partition are arranged consecutively in memory. The numerical scheme is then completely cell-type independent once the metric for the different elements is calculated.

The cell grouping approach drastically reduces the memory requirements of the underlying cell-vertex technique, where quantities are transferred between nodes and cells. A typical three-dimensional Euler calculation with 850,000 nodes (see the section on Falcon-type aircraft) requires 226 Mbytes of memory. For a three-dimensional Navier–Stokes calculation, a total of 59 real words per node is required, half of them for the storage of the solution gradients and the fluxes. Additionally, 41 integer words are allocated per hexahedral cell (with 24 words merely for the node-based face normal vectors).

Parallel Implementation

At present, the parallel library COUPL allows two different parallelization strategies: a master–slave paradigm and the single process multiple data (SPMD) approach. In the former implementation, a unique master process is responsible for the I/O and grid partitioning, but is independent of the slave applications and is hidden to the user. The slaves are distributed across the parallel machine in question and perform all of the computations on a particular subset of the grid. In the alternative SPMD approach, which is arguably more appropriate for many newer parallel machines that only support single tasks per processor, the host process acts as both the master and slave. Through the standard COUPL subroutine interface, the swap between the master–slave and SPMD modes is transparent to AVBP (and the user).

Information is exchanged between the processors using the standard message passing libraries Parallel Virtual Machine (PVM) or Message Passing Interface (MPI) that are interfaced to AVBP through the integrated parallel macros (IPMs) developed at CERFACS.[7] IPM allows us to choose the parallel library at compile time by using the standard preprocessor M4.

Although unstructured data sets are irregular, they have the advantage of being nearly homogeneous. As a result, it is relatively easy to split an unstructured computational domain into an arbitrary number of subdomains. For this reason, any initial (structured) multiblock grid is cast into a single-domain grid, which then allows an optimal partitioning. In the block merging process of the original structured mesh, the creation of a small number of triangles (in two dimensions) or tetrahedrons, pyramids, or prisms can occur, which is tolerated without affecting the overall efficiency. Such elements can become necessary in special regions that have been meshed with degenerated elements in the original multiblock mesh.

The grid partitioning is done in a preprocessing step that is integrated into AVBP and allows the user to decide on the number of partitions at run time as a function of the number of processors available. This tool is a module of COUPL that automatically partitions and reorders the mesh using work estimates for each element type. The partitioning algorithms are based on standard recursive bisection methods such as recursive inertial bisection (RIB) or recursive graph bisection (RGB). These methods aim to load balance the calculation while minimizing interface lengths to reduce the communication between processors.

Usually, only unpartitioned single-domain grids are stored to file, to permit restarts on a (potentially) different number of processors from a previous run. As an exception, an option allows us to use the code as a preprocessor only and to partition the global grid on a large shared memory machine. The multidomain grid is then stored in separate files. These prepartitioned files may then be read and sent to the slaves directly without any further preprocessing. This allows us to circumvent memory limitations for large calculations on certain (massively) parallel architectures.

For the cell-vertex formulation, the subdivision of the global grid results in an increase of the total number of grid points due to a node doubling at the interface between two domains. Copies of these nodes are stored for each partition, although we distinguished between shared/owned or shared/unowned nodes by appropriate processors. Through COUPL a wide range of domain interface treatments is available that allows us to choose between nonoverlapping and overlapping partitions (using layers of additional dummy cells). It also offers the possibility of overlapping computations and communications. These options are incorporated into the cell grouping strategy, where the groups are distinguished between cells interior to a partition or adjacent to an interface.

The COUPL library provides routines for the parallel calculation of global quantities such as sums, norms, and other reduction operations. The distinction between owned and unowned subsets avoids redundant calculations.

Flow Solver

Numerical Method

The flow solver used for the discretization of the governing equations is based on the cell-vertex finite volume method with a central difference Lax–Wendroff discretization in combination with a linear preserving artificial viscosity model. The temporal integration is done by an explicit multistage Runge–Kutta scheme. For steady-state calculations, convergence is accelerated by the local time-stepping method, whereas the implementation of an edge-collapsing multigrid method is under development. This multigrid method will then naturally fit into the local grid refinement framework described next.

Cell-Vertex Method for Arbitrary Elements

As a starting point for the description of the cell-vertex method,[8] where the discrete values of the conserved variables are stored at the grid nodes, we consider the laminar Navier–Stokes equations written in the simple conservative form $(\partial \boldsymbol{w}/\partial t) + \nabla \cdot \boldsymbol{\mathcal{F}} = \boldsymbol{0}$, where $\boldsymbol{w}$ is the vector of conserved variables and $\boldsymbol{\mathcal{F}}$ is the corresponding flux tensor. For convenience, we divide the latter into a inviscid and a viscous part, $\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}^I(\boldsymbol{w}) + \boldsymbol{\mathcal{F}}^V(\boldsymbol{w}, \nabla \boldsymbol{w})$. The spatial terms of the equation are then approximated in each cell control volume Ω_j to give the steady-state cell residual

$$\boldsymbol{R}_{\Omega_j} \approx \frac{1}{V_{\Omega_j}} \int_{\partial \Omega_j} \boldsymbol{\mathcal{F}} \cdot \boldsymbol{n} \, \mathrm{d}S \tag{1}$$

where $\partial \Omega_j$ denotes the boundary of Ω_j with normal $\boldsymbol{n}$.

The present cell-vertex approximation is readily applicable to arbitrary cell types and is, hence, straightforward to apply for hybrid grids. The residual (1) is first computed for each element by making use of a simple integration rule applied to the faces. For triangular faces, we use a straightforward midpoint rule, which is equivalent to the assumption that the individual components of the flux vary

linearly on these faces. For quadrilateral faces, where the nodes may not be coplanar, to ensure that the integration is exact for arbitrary elements if the flux functions do indeed vary linearly, we divide each face into triangles and then integrate over the individual triangles. This so-called linear preservation property plays an important part in our algorithm for ensuring that accuracy is not lost on irregular meshes. Computationally, it is useful to write the discrete integration of Eq. (1) over an arbitrary cell as

$$\boldsymbol{R}_{\Omega_j} = \frac{1}{N_d V_{\Omega_j}} \sum_{i \in \Omega_j} \mathcal{F}_i \cdot \mathrm{d}\boldsymbol{S}_i \tag{2}$$

where $\mathcal{F}_i$ is an approximation of $\mathcal{F}$ at the nodes, N_d represents the number of space dimensions, and $\{i \in \Omega_j\}$ are the nodes of the cell. In this formulation the geometrical information has been factored into terms $\mathrm{d}\boldsymbol{S}_i$ that are associated with individual nodes of the cell but not faces; $\mathrm{d}\boldsymbol{S}_i$ is merely the average of the area-weighted normals for triangulated faces with a common node $i, i \in \Omega_j$. Note that for consistency we have

$$\sum_{i \in \Omega j} \mathrm{d}\boldsymbol{S}_i = \boldsymbol{0}$$

A linear preserving approximation of the divergence operator is obtained if we define the volume V_{Ω_j} consistently as

$$V_{\Omega_j} = \frac{1}{N_d^2} \sum_{i \in \Omega_j} \boldsymbol{x}_i \cdot \mathrm{d}\boldsymbol{S}_i \tag{3}$$

because $\nabla \cdot \boldsymbol{x} = N_d$. Viscous approximations require the solution gradient at the nodes to calculate the flux $\mathcal{F}^V$. A nodal approximation of the gradient is obtained as

$$(\nabla \boldsymbol{w})_k = \frac{1}{V_k} \sum_{j \mid k \in \Omega_j} V_{\Omega_j} (\nabla \boldsymbol{w})_{\Omega_j} \tag{4}$$

using a weighted average of the cell-based gradients $(\nabla \boldsymbol{w})_{\Omega_j}$ that may be defined in a manner similar to the divergence (2) so as to be transparent to linear solution variations.

Once the cell residuals are calculated, we may then define the semidiscrete scheme

$$\frac{\mathrm{d}\boldsymbol{w}_k}{\mathrm{d}t} = -\frac{1}{V_k} \sum_{j \mid k \in \Omega_j} D^k_{\Omega_j} V_{\Omega_j} R_{\Omega_j} \tag{5}$$

where $D^k_{\Omega_j}$ is a distribution matrix that weights the cell residual from cell Ω_j to node k and V_k is a control volume associated with each node. We note that conservation is guaranteed if

$$\sum_{k \in \Omega_j} D^k_{\Omega_j} = I$$

In the present context, Eq. (5) is solved to obtain the steady-state solution using explicit Euler or Runge–Kutta time stepping.

The family of schemes of interest to us makes use of the following definition of the distribution matrix:

$$D^k_{\Omega_j} = 1/n_n [I + C(\delta t_{\Omega_j} / V_{\Omega_j}) \mathcal{A}_{\Omega_j} \cdot \mathrm{d}\boldsymbol{S}_k] \tag{6}$$

where n_n is the number of nodes of Ω_j and $\mathcal{A}$ is the Jacobian of the flux tensor. The simplest central difference scheme is obtained by choosing $C = 0$ and is neutrally stable when combined with Runge–Kutta time stepping. A Lax–Wendroff type scheme may also be formulated, in which case C is chosen to be a constant that depends on the number of space dimensions and the type of cells used: It may be shown that this takes the simple form $C = n_v^2/2N_d$. By replacing the cell time step δt_{Ω_j} by a matrix Φ_{Ω_j} with suitable properties, one may also obtain an SUPG-like scheme that has slightly better convergence and shock-capturing behavior at some extra computational cost.

To develop third-order finite element Taylor–Galerkin-type schemes to improve the modeling of unsteady LES simulations, an effort to measure the accuracy of the present and this new class of schemes has been undertaken.[9] As a result, the order of the standard Law–Wendroff scheme when applied to both regular and perturbed elements (triangles, quadrilaterals, and hexahedrons) ranges from $\mathcal{O}(1.5)$ to $\mathcal{O}(1.6)$ measured in the L_2 norm of the error.

Artificial Viscosity

To avoid spurious small-scale oscillations and poor solutions near shock waves, it is usual to add artificial viscosity terms to the right-hand side of Eq. (5). Here we describe modifications to a model[10] that has the property of preserving a linear solution on any mesh. As with most models, it is a combination of a shock-capturing term and a background dissipation term. For general cells, it takes the form

$$S^k_{\Omega_j} = (\epsilon^{(2)}_{\Omega_j} + \epsilon^{(4)}_{\Omega_j}) \left(\left\{ \frac{1}{n} \sum_{i \in \Omega_j} \boldsymbol{w}_i \right\} - \boldsymbol{w}_k \right) - \epsilon^{(4)}_{\Omega_j} \left\{ \frac{1}{n} \sum_{i \in \Omega_j} \nabla \boldsymbol{w}_i \right\} \cdot \left(\left\{ \frac{1}{n} \sum_{i \in \Omega_j} \boldsymbol{x}_i \right\} - \boldsymbol{x}_k \right) \tag{7}$$

where $\epsilon^{(4)}_{\Omega_j} = \max(0, \kappa^{(4)} - \kappa^{(2)} \epsilon^{(2)}_{\Omega_j})$ and $\kappa^{(2)}$ and $\kappa^{(4)}$ are suitable parameters that take the values $0.5 \le \kappa^{(2)} \le 1.0$ and $0.05 \le \kappa^{(4)} \le 0.2$ for a typical transonic flow calculation. The quantity $\epsilon^{(2)}_{\Omega_j}$ is proportional to a limiter-type function ζ_{Ω_j} that operates on some scalar quantity, such as the pressure:

$$\zeta_{\Omega_j} = \max_{k \in \Omega_j} \frac{\left| \left(\left\{ (1/n) \sum_{i \in \Omega_j} P_i \right\} - P_k \right) - \nabla P_k \cdot \left(\left\{ (1/n) \sum_{i \in \Omega_j} \boldsymbol{x}_i \right\} - \boldsymbol{x}_k \right) \right|}{\left| \left\{ (1/n) \sum_{i \in \Omega_j} P_i \right\} - P_k \right| + \left| \nabla P_k \cdot \left(\left\{ (1/n) \sum_{i \in \Omega_j} \boldsymbol{x}_i \right\} - \boldsymbol{x}_k \right) \right| + P_k} \tag{8}$$

This function (8) is zero if the pressure varies linearly.

Boundary Conditions

The correct formulation of the boundary conditions has been the subject of an extensive investigation.[11] Because of the wide range of applications of AVBP, a large set of boundary conditions is necessary, notably for internal flow configurations. Characteristic boundary conditions may be applied at inlet and outlet boundaries. At run time the user may select either the widely used full residual or the normal approach: In the normal approach (following Ref. 12) the characteristic boundary conditions are applied to the normal flux derivatives of the residual, which is in general more accurate and implies a decomposition of variations in the conservative variables into a set of ingoing and outgoing waves. This condition is mainly applied for internal viscous flows, notably at nonreflecting boundaries, but also to treat corner points, whereas the (less expensive) full-residual approach is satisfactory for far-field boundaries.

Examples of Applications

In this section we will show a representative set of results that provide an overview of the current capabilities of AVBP. Because space is limited, hereinafter we concentrate on applications where either unstructured or hybrid grids were used. Given that most of the steady-state calculations have been published previously, we concentrate on the newly developed local grid refinement tool and to a larger extent on the unsteady flow calculations. All steady-state computations and most of the unsteady cases were carried out with the Lax–Wendroff scheme. The use of central differences is indicated where appropriate.

Steady-State External Aerodynamics

Unstructured Grid of Tetrahedrons: Falcon-Type Aircraft

The generic twin-engine airplane Falcon-JJ is an example of a large-scale aerodynamics calculation on an unstructured grid. The computational grid is composed of 847,000 tetrahedral elements

Table 1 Speedup on SGI Power Challenge for the unstructured Falcon aircraft

Processors	Machine	Total nodes	Cells/proc.	Speedup
1	SGI	155,932	847,361	1.00
2	SGI	159,098	423,680	1.98
4	SGI	163,932	211,840	2.70
8	T3D	169,575	105,920	1.00
16	T3D	177,917	52,960	1.96
32	T3D	188,785	26,480	3.63
64	T3D	201,832	13,240	6.69
128	T3D	220,817	6,620	12.42

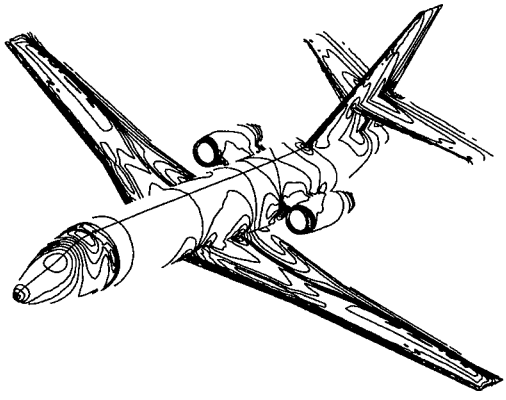

Fig. 1 Pressure contours for Falcon aircraft.

and is a courtesy of Imperial College and EPF Lausanne. The free-stream conditions are the transonic flow at $M_\infty = 0.85$ and $\alpha = 2$ deg. Figure 1 shows the calculated pressure contours that reveal shocks at the upper wing surface and behind the canopy.

The global grid has been partitioned using the RIB algorithm and was mapped for 1–4 processors on an SGI Power Challenge and 8–128 processors of a Cray T3D. The corresponding speedup table is given in Table 1.

Local Grid Refinement

In general, the use of mesh refinement techniques is motivated by the demand to optimize the resolution of flow calculations. Often one does not know a priori exactly where to cluster additional points to better resolve the physical phenomena of interest. It is therefore necessary to determine this automatically during the calculation. Feedback about the flow is obtained in the form of some error indicator. Of the two principal strategies for mesh adaptation, namely, the grid point distribution approach and the local grid refinement method, only the latter is considered within AVBP. A preprocessing tool has been developed that allows isotropic and anisotropic refinement recursively on several levels.[13]

One of the most serious problems with using structured refinement of hexahedrons is the appearance of internal interfaces between coarse and fine cells that are characterized by discontinuous grid lines and hanging nodes. Two basic approaches have emerged to address this problem. In the buffer cell approach,[14] the hanging nodes are removed by the creation of unstructured transition cells at the interface that fill out the element that is only partially refined. The handling of these buffer cells is straightforward for a hybrid solver that can deal with simplex elements. The shortcoming of this approach lies in the creation of a nonnegligible overhead due to the generation of the additional buffer elements. The alternative derived cell approach (adopted by Ref. 15, for example) allows nonsimplex elements with nodes on the edges or the faces of a refinement interface. No extra buffer cells are created at the interfaces. Within AVBP the various types of derived elements are added to the list of basic elements and fit well into the hybrid mesh environment.

We give two examples for three-dimensional isotropic local grid refinement applied to steady-state problems for a hexahedral grid (Fig. 2) and a tetrahedral grid (Fig. 3). Figure 4 shows the resulting C_p contours for inviscid transonic flow over the ONERA M6 wing, with $M_\infty = 0.84$ and $\alpha = 3.06$ deg. In this case, the refined

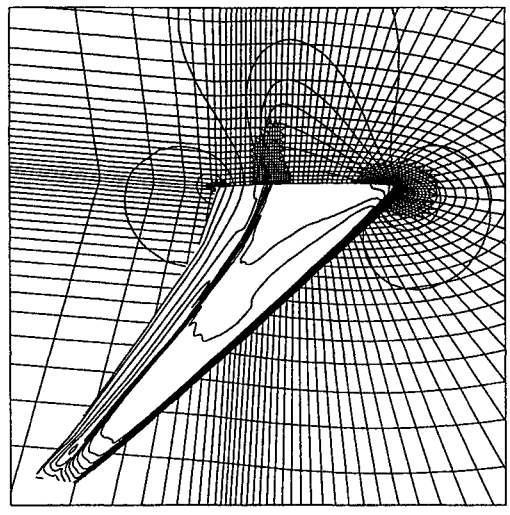

Fig. 2 Locally refined grid and pressure contours for transport aircraft wing.

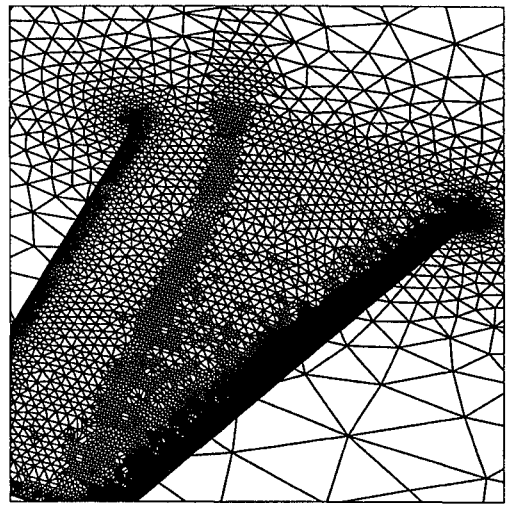

Fig. 3 Locally refined grid for ONERA M6 wing.

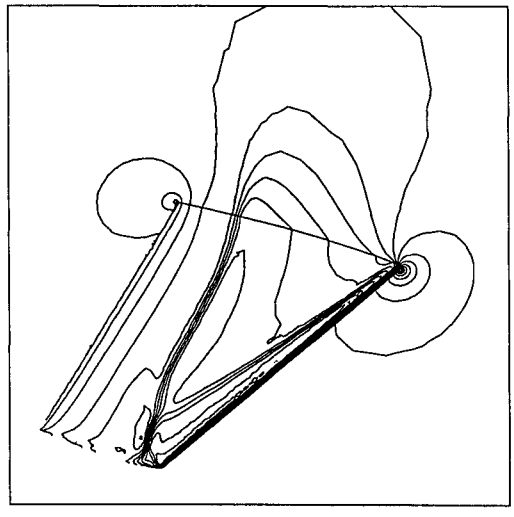

Fig. 4 Pressure contours of ONERA M6 wing.

grid with hanging nodes contains 703,196 tetrahedral elements, with 1,004,919 elements in the buffered grid. The corresponding convergency histories are shown in Fig. 5.

These stationary simulations require refinement only a few times during a typical calculation, and static load balancing for parallel computing can be done after each refinement. Because for large-scale applications storage is exhausted after three or four levels of adaptation, the overhead of static as opposed to dynamic balancing

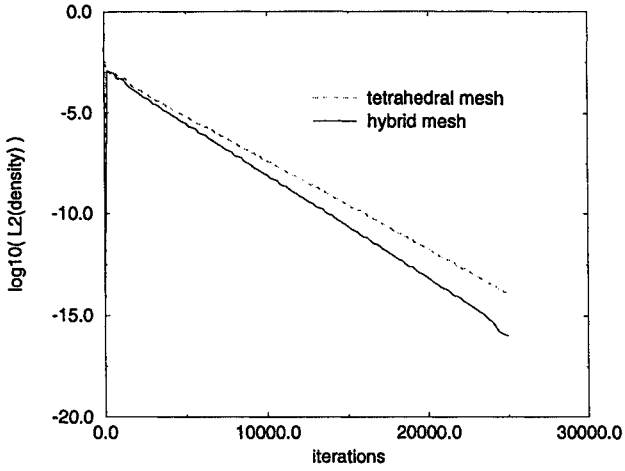

Fig. 5 Convergence history for ONERA M6 wing.

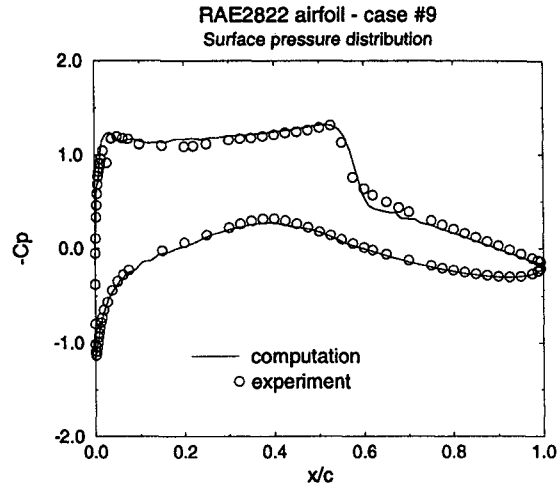

Fig. 6 Comparison of experiment and turbulent flow calculation with Spalart–Allmaras model.

in parallel is negligible when compared to the overall time required for the solver. However, local refinement for unsteady flows requires by far more adaptation cycles and, hence, fast refinement algorithms. Then the issue of the nontrivial dynamic load balancing needs to be addressed as well.

Turbulence Modeling

Although unstructured grids are widely used for inviscid compressible flow calculations, their application to laminar viscous and turbulent flows is still limited. One important problem associated with turbulent flow calculations on unstructured grids is the need of highly stretched cells for the accurate resolution of the viscous fluxes in boundary layers or wakes. For high-Reynolds-number flows, the viscous layer becomes very thin, and a turbulent closure suitable for unstructured grids is required. Unfortunately, simple algebraic turbulent models are often difficult to implement into an unstructured environment, as many require algorithms to determine global function maxima across boundary layers.

One-equation Spalart–Allmaras model. For the simulation of turbulent external flows, the one-equation Spalart–Allmaras model represents a good compromise between modeling complexity and computational costs.[16] This popular model has been designed for external flows around wings and is well suited to an parallel unstructured grid framework due to its local formulation.

With this model the eddy viscosity is obtained by solving an additional transport equation for a quantity similar to the turbulent viscosity. Unlike the nonconservative formulation of the original Spalart–Allmaras model, in our approach a conservative formulation has been adopted that leads to the turbulent variable $\tilde{\mu}_t = \rho\tilde{\nu}$. In our implementation an implicit formulation of the source term can be used to accelerate convergence. The calculation of the wall distances is done in parallel as a preprocessing step, whereby each slave sends a list of its wall boundary nodes to the root slave; the accumulated list of all wall nodes is broadcasted back to the slaves that then compute the distance of the nodes to any of the wall nodes in parallel.

A result is shown for the standard workshop case 9 of the RAE2822 airfoil under transonic flow conditions at $M_\infty = 0.73$, $Re_c = 6.5 \times 10^6$, and $\alpha = 2.8$ deg. An unstructured grid is employed with 24,000 nodes and 47,700 triangular elements. Figure 6 shows the surface pressure distribution, which agrees well with the experimental data except for a small zone downstream the shock. The coefficients for lift and drag are $c_L = 0.831$ and $c_D = 0.0181$, whereas in the experiment $c_L = 0.803$ and $c_D = 0.0168$.

Two-equation k–ϵ model with wall functions. A two-equation k–ϵ model has been primarily developed for internal flows.[17] The coupled transport equations for the turbulent kinetic energy k and its dissipation rate ϵ are added to the mean flow equations and solved in a manner similar to that used for the laminar Navier–Stokes equations.

For the simulation of turbulent high-Reynolds-number flows that require a very fine resolution of the strong velocity gradients near walls, the standard k–ϵ model is modified in near-wall regions and replaced by so-called wall functions. The resolution of the turbulent transport equations in the boundary layer is then replaced by a semilogarithmic law of the wall.

Unsteady Flows

Whereas our flow solver was initially conceived for stationary (external) flows, the current trend of applications is toward the modeling of unsteady (internal) flows. These increasing activities are strongly related to the rising importance of understanding the flow structure and the mechanisms leading to turbulence. By providing new tools to investigate the instantaneous three-dimensional structure of turbulent flows, DNS and LES have allowed significant advances both in understanding and modeling of turbulence. Although an incompressible module is planned, at present AVBP solves compressible turbulent flows. Most of these studies on unsteady turbulent flows have been carried out by Nicoud,[18] Nicoud et al.,[19] and Ducros et al.[20] at CERFACS.

The required high spatial resolution often limits the range of applicability of the DNS/LES approach to simple model configurations such as isotropic homogeneous turbulence, channel flows, or boundary layers. Large computational domains are necessary to minimize the influence of inflow/outflow boundary conditions. The next step for DNS, as well as LES, is to address more complex flows. This task requires the development of suitable numerical schemes providing high accuracy on arbitrary meshes, new treatments to handle inflow and outflow boundary conditions (including acoustic waves for compressible cases), and efficient algorithms to limit the computational costs. The use of hybrid grids is an attempt to tackle the grid point distribution problem.

LES

The LES approach appears as an attractive compromise between Reynolds-averaged Navier–Stokes turbulence models (which often fail to well predict the unsteady flow behavior) and the DNS (which is restricted to very low Reynolds numbers and small configurations and, hence, is not directly applicable for practical industrial needs). The underlying idea behind the LES approach is to filter the complete system of the compressible Navier–Stokes equations, which thus exhibits subgrid scale tensors and vectors describing the interaction between nonresolved and resolved motion. The influence of nonresolved scales on the resolved motion is taken into account by means of a subgrid-scale model based on the introduction of a turbulent viscosity ν_t.

Two models have been developed for unstructured meshes: The first is a classical Smagorinsky-type formulation, and the second model is based on a high-pass filtered velocity field. The accuracy of these models has been demonstrated for freely decaying isotropic homogeneous turbulence.[20] Many difficulties related to the use of a subgrid model in a unstructured mesh have been investigated. Some ideas initially developed for a structured environment were generalized to a more general hybrid formulation. Notably,

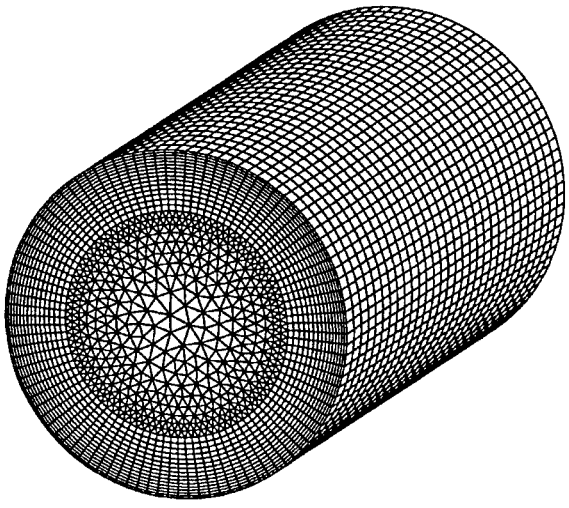

Fig. 7 Closeup of the hybrid mesh for the pipe flow with 240,000 nodes.

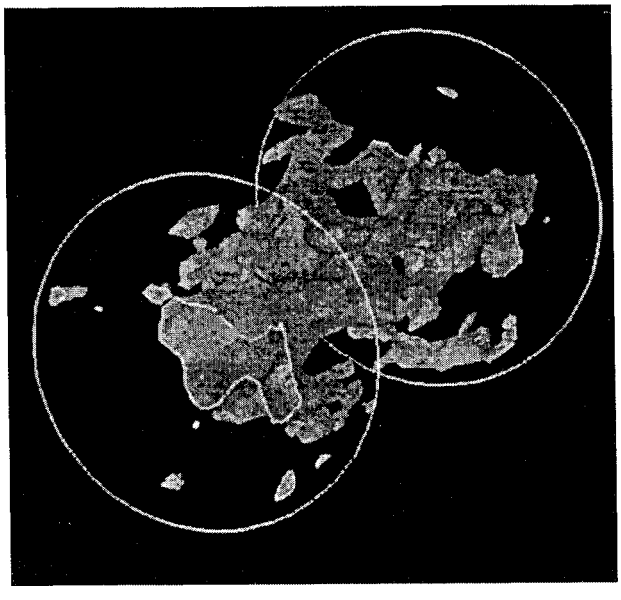

Fig. 8 Isosurface of streamwise velocity after the transitional stage.

the original dissipative operator has been modified to enhance the stability for high-frequency modes[18] because the standard centered scheme is only neutrally stable for high-frequency modes and the highest mode is not damped at all.

We apply the filtered LES model for the simulation of a turbulent pipe flow. The Reynolds number is $Re_{U_b} = 1 \times 10^4$, based on the pipe diameter and the bulk velocity. The centerline Mach number is of order 0.25. The initial flowfield is a Poiseuille flow together with a white noise of small amplitude to trigger the transition to turbulence. A source term is added to simulate a pressure gradient that corresponds to the fully turbulent state. A hybrid mesh is used with hexahedrons near the wall and prisms in the core region (Fig. 7). The total number of nodes is 240,000. It has been designed to reproduce the main features of the fully developed turbulent flow. In Fig. 8 we observe classical parietal turbulent motion with ejections of relative low-speed fluid from the wall to the upper part of the layer. These coherent structures compare well with the DNS results.[21]

Active Flow Control

The active control of unsteady flows is of high interest for practical applications such as the control of boundary-layer separation as means for drag reduction for airplane wings. The present example of a two-dimensional cavity geometry represents an aeroacoustics coupling problem, where flow control is applied to determine the mechanisms that lead to the creation of noise and its reduction.[22] These studies exhibit a certain complexity due to the coupling of different physical aspects such as hydrodynamic instabilities, acoustic waves, and fluid–wall interactions.

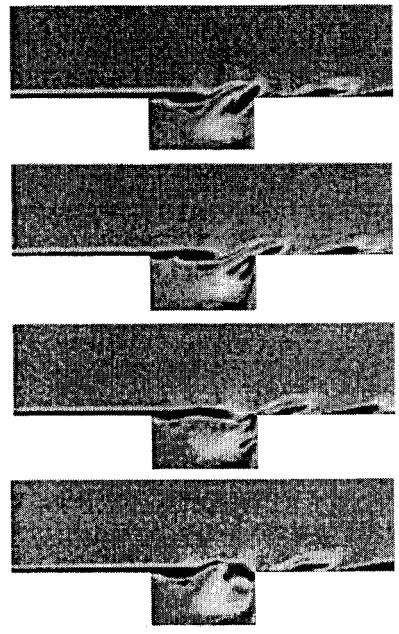

Fig. 9 Vorticity isolines at instants 0, $\frac{1}{4}T$, $\frac{1}{2}T$, and $\frac{3}{4}T$ over a period of the instability of the cavity flow.

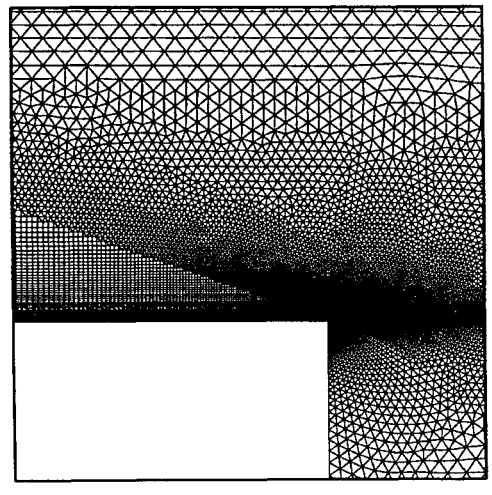

Fig. 10 Closeup at left corner of a hybrid grid for a cavity.

Figure 9 shows the flow pattern at different instants over a pseudoperiodical phase of the instability. At the left corner a mixing layer forms followed by a periodic vortex shedding. These vortices are convected downstream, where they impinge on the right corner and split into two parts. While one part dissipates as it is convected to the exit boundary, a second subvortex stays inside the cavity where it is caught by a large recirculation zone.

The vortex burst at the right corner is accompanied by the formation of various acoustic waves that lead to an important creation of noise inside the cavity. The goal of this work is to diminish these noise levels by actively influencing the flow characteristics. In the open-loop control approach, a synthetic jet acts as actuator. Unsteady suction and blowing is imposed just upstream of the point of creation of the mixing layer in the vicinity of the left corner. A specific frequency of this pulsed jet can be found that significantly reduces the pressure levels. Ongoing studies investigate the closed-loop mode, where the mass flow rate of the actuator is controlled by a sensor placed in the vicinity of the downstream corner.

A hybrid grid best meets the resolution requirements for this configuration. The mesh is composed of 35,000 points with quadrilateral elements in the near-wall regions and of triangles in the outer domain (Fig. 10). The boundary layer is resolved by at least 20 nodes

that follow the boundary-layer thickness. The flow conditions are the unsteady laminar flow at $M_\infty = 0.2$ and $Re_\delta = 220$ based on the inflow boundary-layer thickness.

Although currently we examine the active control of laminar flows, the near-future objective is to control three-dimensional turbulent flow configurations based on the LES approach described in the preceding section.

Chemistry Modeling

The unsteady flow capabilities of AVBP open the field to other applications such as laminar and turbulent combustion. For this, a simple chemistry model has been developed that allows us to specify additional species following the Arrhenius law. The objective of these investigations is to simulate reactive turbulent flows with heterogeneous species concentrations in complex geometries.

Pulse Detonation Engine

The studies on a pulse detonation engine applied to missiles represents a work of O. Colin undertaken in collaboration with Aérospatiale Missile Division. The goal of these initial investigations is to predict the propagation speed of the detonation shock wave. This speed is primarily governed by the heat release, the polytropic coefficient gamma, the molar weight, and the equivalence ratio, while viscous and turbulent effects are negligible in the propagation phase of the detonation. A Roe total variation diminishing scheme with limiters following Barth has been implemented to remedy the numerical oscillations due to the centered Lax–Wendroff scheme. We show the Euler results for a simplified two-dimensional detonation engine configuration of ENSMA at Poitiers, France, for which experimental data are available.[23] At initial time the chamber is filled with a stoichiometric mixture C_2H_4/O_2 and air outside. The detonation is initiated by an analytical one-dimensional profile at the chamber entrance. The pressure at the thrust wall of the chamber shows good agreement between the numerical simulations and experimental measurements, shown in Fig. 11, where the variation of the pressure is plotted as a function of time at the detonation ignition position and for a chamber lengths ratio $l/d = 2.52$. Figure 12 shows the temporal evolution of the Mach number at two different instants. This configuration has an extremely fine grid in the small chamber section (78,750 quadrilaterals). To avoid spreading of these fine elements into the large outer part, the latter is filled with 14,224 triangles.

LES in a Premixed Burner

The configuration of a two-dimensional premixed burner is an example illustrating the use of AVBP for LES computations of reactive flows on hybrid meshes. A mixture of methane and air is fed to a dump combustor. Under certain conditions, this burner exhibits strong instabilities with a very characteristic frequency. The purpose of these computations is to reproduce these instabilities and to compare the flame response to that measured experimentally by Poinsot et al.[24]

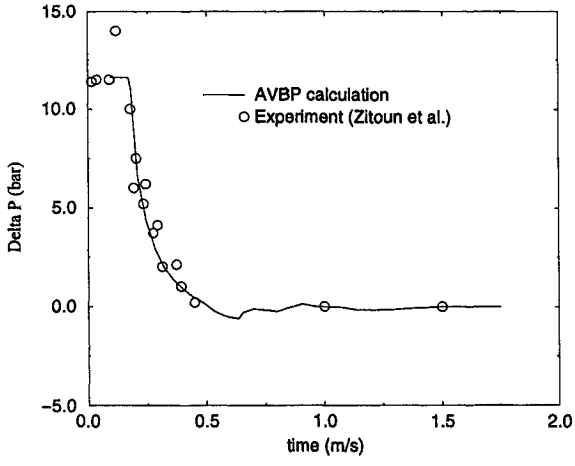

Fig. 11 Variation of pressure as a function of time at detonation initiation position.

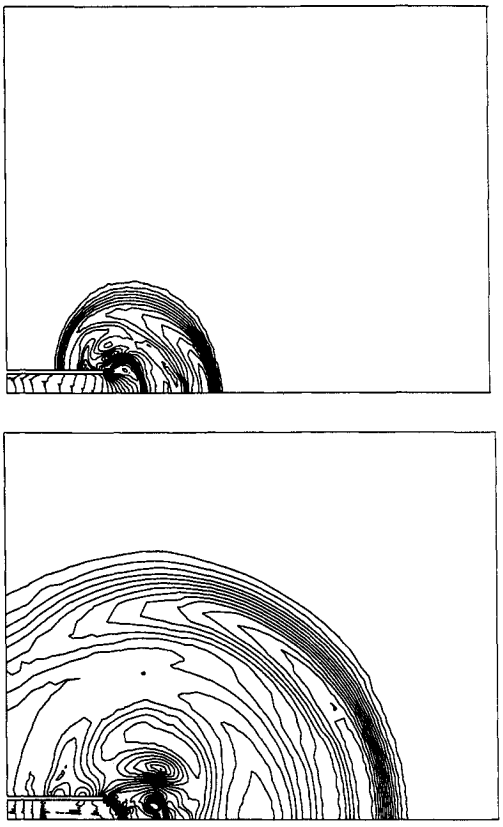

Fig. 12 Mach number isolines at $T = 0.196$ and $T = 0.618$ ms for a detonation wave leaving a pulse detonation engine.

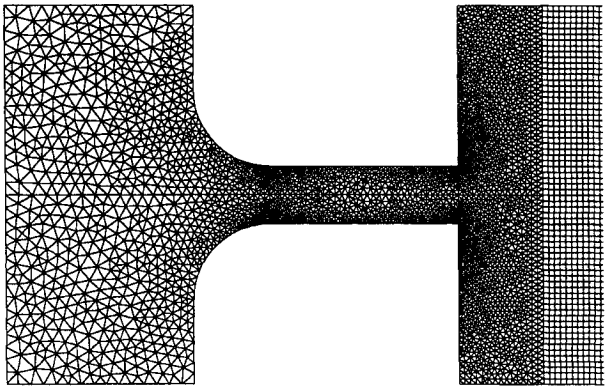

Fig. 13 Closeup of the computational mesh for the LES studies of combustion instabilities in a premixed burner.

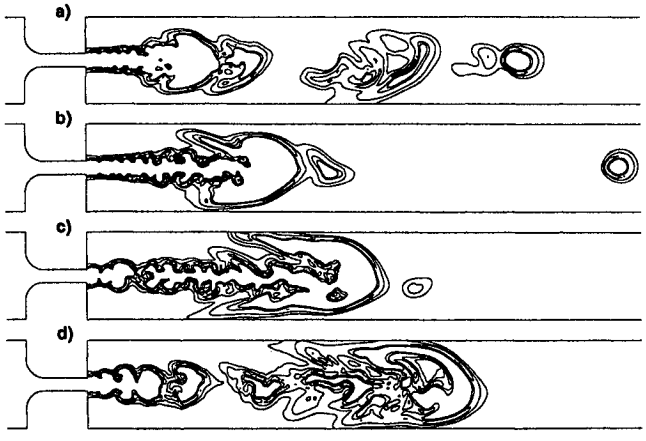

Fig. 14 LES of a premixed burner: reaction rate fields during one cycle of acoustic forcing.

A close-up of the mesh is shown in Fig. 13. The part upstream and near downstream of the dump is meshed using 6500 triangular elements, while the far downstream part of the dump is composed of 42,000 evenly spaced quadrilateral elements. This hybrid mesh offers a good compromise between mesh size and resolution requirements in the vicinity of the dump. Figure 14 shows a typical result. An acoustic excitation is fed to the inlet of the domain and generates a mushroom vortex at the dump. This vortex increases the flame surface and, thus, the volumetric heat generation in the combustor. The time delay between the excitation (corresponding to the experimentally known instability frequency of 530 Hz) and the increase in heat generation is very close to experimental findings. These results are discussed in full detail by Angelberger et al.[25]

Conclusions and Future Work

The flow solver AVBP developed at CERFACS for the three-dimensional laminar and turbulent Navier–Stokes equations is presented. The code is built on a parallel library and handles hybrid grids of arbitrary element types. Although initially conceived for steady-state flows, the current main trend of activity is toward the development of tools for the simulation of unsteady compressible flows. Results for both stationary aerodynamical applications and unsteady turbulent flows are presented. Studies on active flow control and turbulent combustion are further topics of interest.

The future development will be driven by these activities and consists, in addition to improving the physical models and their application to complex geometries, of the development of efficient numerical techniques for hybrid grids. These comprise robust higher-order methods for the unsteady turbulent simulations as well as edge-collapsing multigrid techniques. Work on fully automatic mesh adaptation techniques is continuing and will include aspects such as parallel load balancing. Multiphysical applications issuing from problems related to aeroelasticity represent another potential area for future research. A (pre)industrialization of AVBP is under way.

Acknowledgments

Although the authors of this survey paper are the initiators and main developers of the AVBP solver, covering the large spectrum of applications was only possible thanks to the work and results obtained by current and former members of the computational fluid dynamics team at CERFACS. We particularly acknowledge the scientific contributions of Christian Angelberger, Olivier Colin, Frédéric Ducros, Sébastien Ducruix, Grégory Hernandez, Tim Kestens, Jean-Philippe Légier, Jens-Dominik Müller, Franck Nicoud, Christophe Nottin, Jörg Schlüter, Robert Struijs, and Thierry Poinsot.

References

[1]Ward, S., and Kallinderis, Y., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex 3-D Geometries," AIAA Paper 93-0669, Jan. 1993.

[2]Kallinderis, Y., "Discretization of Complex 3-D Flow Domains with Adaptive Hybrid Grids," *Proceedings of the Fifth International Conference on Numerical Grid Generation*, National Science Foundation Engineering Center, Mississippi State Univ., Mississippi State, MS, 1996, pp. 505–515.

[3]Chappell, J., Shaw, J., and Leatham, M., "The Generation of Hybrid Grids Incorporating Prismatic Regions for Viscous Flow Calculations," *Proceedings of the Fifth International Conference on Numerical Grid Generation*, National Science Foundation Engineering Center, Mississippi State Univ., Mississippi State, MS, 1996, pp. 537–546.

[4]Gerhold, T., Friedrich, O., Evans, J., and Galle, M., "Calculation of Complex Three-Dimensional Configurations Employing the DLR-τ-Code," AIAA Paper 97-0167, Jan. 1997.

[5]Soetrisno, M., Imlay, S., Roberts, D., and Taflin, D., "Computations of Viscous Flows for Multi-Element Wings Using Hybrid Structured-Unstructured Grids," AIAA Paper 97-0623, Jan. 1997.

[6]Schönfeld, T., and Rudgyard, M., "COUPL and Its Use Within Hybrid Mesh CFD Applications," *Proceedings of the 10th International Conference on Parallel CFD 98*, edited by A. Ecer, D. Emerson, J. Periaux, and N. Satofuka, Elsevier, Amsterdam, 1998.

[7]Giraud, L., Noyret, P., Sevault, E., and van Kemenade, V., *IPM 2.0 User's Guide and Reference Manual*, CERFACS, Toulouse, France, 1994.

[8]Rudgyard, M., "Cell Vertex Methods for Steady Inviscid Flow," VKI Lecture Series 1993-04, von Kármán Inst. for Fluid Dynamics, Rhode St. Genese, Belgium, 1993.

[9]Colin, O., and Rudgyard, M., "Tayler Galerkin Schemes in AVBP," *Journal of Computational Physics* (submitted for publication).

[10]Morgan, K., Peraire, J., and Peiro, J., "Unstructured Grid Methods for Compressible Flows," AGARD R-787, 1992.

[11]Nicoud, F., "Defining Wave Amplitude in Characteristic Boundary Conditions," *Journal of Computational Physics*, Vol. 149, No. 2, 1999, pp. 418–422.

[12]Poinsot, T., and Lele, S., "Boundary Conditions for Direct Simulations of Compressible Viscous Flows," *Journal of Computational Physics*, Vol. 101, No. 1, 1992, pp. 104–129.

[13]Müller, J.-D., Schönfeld, T., and Rudgyard, M., "A Comparison of the Treatment of Hanging Nodes for Hybrid Grid Refinement," AIAA Paper 97-1859, June 1997.

[14]Biswas, R., and Strawn, R., "A Dynamic Mesh Adaptation Procedure for Unstructured Hexahedral Grids," AIAA Paper 86-0027, Jan. 1996.

[15]Mavriplis, D., "Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed-Element Unstructured Meshes," AIAA Paper 97-0857, Jan. 1997.

[16]Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1992.

[17]Schönfeld, T., Colin, O., and Rudgyard, M., "Parallel Implementation of a $k-\epsilon$ Turbulence Model and Wall-Functions for Unstructured Grids," AIAA Paper 96-2061, June 1996.

[18]Nicoud, F., "Effects of Strong Wall Injection on the Structure of a Low-Reynolds Turbulent Flow," CERFACS, CFD-TR/97-7, Toulouse, France, 1997.

[19]Nicoud, F., Ducros, F., and Schönfeld, T., "Towards Direct and Large Eddy Simulations of Turbulent Flows in Complex Geometries," *Computation and Visualization of Three-Dimensional Vortical and Turbulent Flows: Proceedings of the Fifth CNRS/DFG Workshop on Numerical Flow Simulation*, edited by R. Friedrich and P. Bontoux, Vol. 64, Notes on Numerical Fluid Mechanics, Vieweg, Brunswick, Germany, 1998, pp. 157–171.

[20]Ducros, F., Nicoud, F., and Schönfeld, T., "Large Eddy Simulations of Compressible Flows on Hybrid Meshes," *Proceedings of the 11th Symposium on Turbulent Shear Flows*, Institut National Polytechnique Grenoble, Grenoble, France, 1997, pp. 28-1–28-6.

[21]Eggels, J., Unger, F., Weiss, M., Westerweel, J., Adrian, R., Friedrich, R., and Nieuwstadt, F., "Fully Developed Turbulent Pipe Flow: A Comparison Between Direct Numerical Simulation and Experiment," *Journal of Fluid Mechanics*, Vol. 268, 1994, pp. 175–209.

[22]Kestens, T., and Nicoud, F., "Active Control of an Unsteady Flow over a Rectangular Cavity," AIAA Paper 98-2348, June 1998.

[23]Zitoun, R., Gamezo, V., and Desbordes, D., "Experimental Optimization of a Single Operating Cycle of Pulsed Detonation Engine," *Proceedings of the Second International Meeting on Properties of Reactive Fluids and Their Application to Propulsion*, Ecole Nationale Superieure de Mecanique et d'Aerotechnique, Futuroscope, France, 1996.

[24]Poinsot, T., Trouvé, A., Veynante, D., Candel, S., and Esposito, E., "Vortex-Driven Acoustically Coupled Combustion Instabilities," *Journal of Fluid Mechanics*, Vol. 177, 1987, pp. 265–292.

[25]Angelberger, C., Veynante, D., Egolfopoulos, F., and Poinsot, T., "Large Eddy Simulations of Combustion Instabilities in Premixed Flames," *Proceedings of the CTR Summer Program*, Stanford Univ./NASA Ames Research Center, Palo Alto, CA, 1998, pp. 67–82.

J. Kallinderis
Associate Editor