AIAA JOURNAL
Vol. 38, No. 6, June 2000

Modeling of Three-Dimensional Viscous Compressible Turbomachinery Flows Using Unstructured Hybrid Grids

A. I. Sayma,* M. Vahdati,† L. Sbardella,‡ and M. Imregun§
Imperial College of Science, Technology, and Medicine, London, England SW7 2BX, United Kingdom

An advanced numerical model for the simulation of steady and unsteady viscous compressible flows for turbomachinery applications is described. The compressible Favre-averaged Navier–Stokes equations are used together with a one-equation turbulence model. The flow domain is discretized using unstructured hybrid grids that can contain a mixture of hexahedral, pentahedral, tetrahedral, and triangular prismatic cells. The flow equations are discretized using a node-centered finite volume scheme that relies on representing the mesh using an edge-based data structure. A dual time stepping technique is applied to a point implicit formulation so that time accuracy can be maintained with large Courant–Friedrichs–Lewy numbers. Nonreflecting boundary conditions are applied at the inflow and outflow boundaries to prevent any spurious reflections of the outgoing waves. The model was validated against measured data for two cases. Radial profiles of pressure and temperature rise were determined from the steady flow analysis of a rig fan blade, and these were found to be in very good agreement with the measured quantities. A rotor/stator interaction was studied next. Detailed comparisons were carried out against measured steady and unsteady flow data and good agreement was obtained in all cases.

Introduction

THE nature of the flow in modern turbomachines is complicated due to the coexistence of subsonic, supersonic, and transonic regions, in addition to shock waves and shock–boundary-layer interactions. The situation is further compounded by the presence of acoustic waves that may cause acoustic resonances. From a design point of view, it is essential to be able to determine the steady loading on the blades accurately so that reliable performance parameters can be obtained. Of equal importance is the unsteady loading that might result in unacceptably high dynamic blade stresses. Because of such considerations, a realistic simulation of turbomachinery flows requires a time-accurate viscous representation of the unsteady compressible flow.

In recent years, the rapid development of numerical methods for the solution of the flow equations and the availability of powerful computers led to the emergence of various systems for the prediction of complex turbomachinery flows,[1,2] though most methods use structured grids.[1–4] On the other hand, unstructured grids received a great deal of research and development effort for external compressible flows.[5–10] It is only in recent years that unstructured tetrahedral grids found their way into turbomachinery applications.[11–14] Although unstructured grid methods provide flexibility for discretizing complex geometries, they have the drawback of requiring larger in-core memory and more CPU effort than their structured counterparts. Structured grids are considered to be the most suitable route to the discretization of turbomachinery blades because of their relatively simple shape. Design considerations increasingly require the inclusion of complex features such as tip gap leakage, cooling holes in turbine blades, snubbered fan blades, fan assemblies with intake ducts, struts, and various other structural elements. Because of the complexity of such geometries, the natural way forward is to use unstructured grids. Although tetrahedral grids, the choice of which appears obvious, are relatively easy to generate for capturing the inviscid features of the flow, the situation becomes more complicated in boundary layers, where large aspect ratio cells are required for computational efficiency. The gradients normal to the walls are several orders of magnitude larger than those along the walls; thus, more grid points are required in the former direction than the latter. Tetrahedral grids are not ideal for use in boundary layers where very large or very small cell angles degrade the accuracy of the solution.

Such considerations led to the development of hybrid grid models where hexahedral or prismatic cells can be used in the boundary layers and tetrahedral and prismatic cells can be used to fill the domain away from the walls. For turbomachinery blades, Sbardella et al.[15] presented a method to generate hybrid semistructured grids where the boundary layers are filled with hexahedral cells and the rest of the domain is filled with triangular prisms. Such a route not only provides a very efficient spatial discretization over standard unstructured grids, but it also provides, for a comparable number of points, a much better grid quality over its fully structured counterparts. The present work also uses semistructured grids for their computational efficiency, although the solver is written for general hybrid unstructured grids. To achieve further computational efficiency, the mesh is represented using an edge-based data structure. In this approach, the grid is presented to the solver as a set of node pairs connected by edges. The edge weights representing the intercell boundaries are computed in a separate preprocessor stage. Consequently, the solver has a unified data structure for which the nature of the hybrid mesh is concealed from the main calculation loops.

The system of equations is advanced in time using an implicit second-order time integration. A point relaxation procedure with Jacobi iteration is used for steady flows. Solution acceleration techniques, such as residual smoothing and local time stepping, are also employed. A dual time stepping procedure is used for unsteady flow computations: In addition to the internal Jacobi iterations with a pseudo-time step, time accuracy is ensured with external Newton iterations. Within each Newton iteration, a point Jacobi relaxation and its associated acceleration techniques are used to drive the solution to convergence. The flow model will now be explained in more detail.

Flow Model

The unsteady, compressible, Favre-averaged Navier–Stokes equations for a three-dimensional blade row can be cast in terms of absolute velocity $\boldsymbol{u}$ but solved in a relative non-Newtonian reference frame rotating with the blade about the x_1 axis with angular velocity ω. This system of equations, written in an arbitrary Eulerian Lagrangian (ALE) conservative form for a control volume Ω with boundary Γ, takes the following form:

Received 24 January 1999; revision received 25 October 1999; accepted for publication 8 November 1999.

*Research Fellow, Mechanical Engineering Department, Exhibition Road; a.sayma@ic.ac.uk.

†Research Fellow, Mechanical Engineering Department, Exhibition Road; m.vahdati@ic.ac.uk.

‡Research Student, Mechanical Engineering Department, Exhibition Road; l.sbardella@ic.ac.uk.

§Reader, Mechanical Engineering Department, Exhibition Road; m.imregun@ic.ac.uk.

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \boldsymbol{U}\,\mathrm{d}\Omega + \oint_{\partial\Omega}\left(\boldsymbol{F} - \frac{1}{Re}\boldsymbol{G}\right)\cdot\boldsymbol{n}\,\mathrm{d}\Gamma = \int_{\Omega}\boldsymbol{S}\,\mathrm{d}\Omega \tag{1}$$

where $\boldsymbol{n}$ represents the outward unit vector of the control volume boundary Γ. The viscous term $\boldsymbol{G}$ on the left-hand side of Eq. (1) has been scaled by the reference Reynolds number for nondimensionalization purposes. The solution vector of conservative variables $\boldsymbol{U}$ is given by

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho\boldsymbol{u} \\ \rho e \end{bmatrix} \tag{2}$$

The inviscid flux vector $\boldsymbol{F}$ has the following components:

$$\boldsymbol{F} = \boldsymbol{U}\boldsymbol{v} + \begin{bmatrix} 0 \\ p\delta_{ij} \\ u_j p \end{bmatrix} \tag{3}$$

where δ_{ij} represents the Kronecker delta function and $\boldsymbol{v}$ is the velocity in the relative frame of reference. The pressure p and the total enthalpy h are related to density ρ, absolute velocity $\boldsymbol{u}$, and internal energy e by two perfect gas equations:

$$p = (\gamma - 1)\rho[e - (|\boldsymbol{u}|^2/2)], \qquad h = e + (p/\rho) \tag{4}$$

where γ is the constant specific heat ratio. The viscous flux vector $\boldsymbol{G}$ has the following components:

$$\boldsymbol{G} = \begin{bmatrix} 0 \\ \sigma_{ij} \\ u_k\sigma_{ik} + \dfrac{\gamma}{\gamma-1}\left(\dfrac{\mu_l}{Pr_l} + \dfrac{\mu_t}{Pr_t}\right)\dfrac{\partial T}{\partial x_i} \end{bmatrix} \tag{5}$$

The viscous stress tensor σ_{ij} is expressed using the eddy viscosity concept, which assumes that, in analogy with viscous stresses in laminar flows, the turbulent stresses are proportional to the mean velocity gradients:

$$\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\delta_{ij}(\nabla\cdot\boldsymbol{u}) \tag{6}$$

where μ_l represents the molecular viscosity given by the Sutherland's formula and μ_t denotes the turbulent eddy viscosity, which must be determined by a suitable turbulence model. Therefore, $\mu = \mu_l + \mu_t$ is the total viscosity of the fluid. The value of λ is given by the Stokes relation $\lambda = -\frac{2}{3}\mu$ whereas the laminar Prandtl number Pr_l is taken as 0.7 for air. The turbulent Prandtl number Pr_t is taken as 0.9.

The term $\boldsymbol{S}$ in Eq. (1) is given by

$$\boldsymbol{S} = [0 \quad 0 \quad \rho\omega u_2 \quad \rho\omega u_3 \quad 0]^T \tag{7}$$

The eddy viscosity μ_t is calculated using the one-equation turbulence model of Baldwin and Barth,[16] though other one- and two-equation turbulence models such as that of Spalart and Allmaras[17] and $q-\zeta$ (Ref. 18) are under consideration.

Numerical Methodology

The three-dimensional spatial domain is discretized using unstructured grids that, in principle, can contain cells with any number of boundary faces. The solution vector is stored at the vertices of the cells.

For clarity, the numerical discretization of the flow equations will be illustrated on a two-dimensional mesh. However, the resulting formulation is equally applicable to three-dimensional cells. Using an edge-based scheme, the typical two-dimensional mesh of Fig. 1 can be discretized by connecting the median dual of the cells surrounding an internal node. For internal node I, the semidiscrete form can be written as

$$\frac{\mathrm{d}(\boldsymbol{\Omega}_I\boldsymbol{U}_I)}{\mathrm{d}t} + \sum_{s=1}^{n_s}\frac{1}{2}\left|\boldsymbol{\eta}_{IJ_s}\right|\left(\mathcal{F}_{IJ_s} - \mathcal{G}_{IJ_s}\right) + B_i = \boldsymbol{\Omega}_I\boldsymbol{S}_I \tag{8}$$

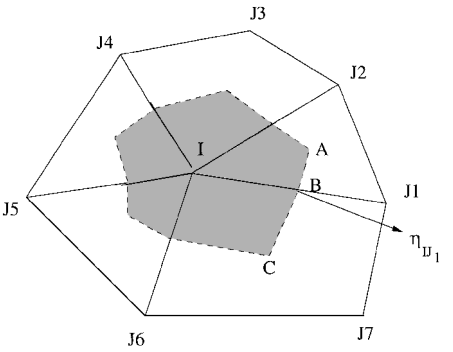


Fig. 1 Typical two-dimensional mixed-cell mesh.

where $\boldsymbol{\Omega}_I$ is the area of the control volume (shaded area in Fig. 1, $\boldsymbol{U}_I$ is the solution vector at node I, n_s is the number of sides connected to node I, $\mathcal{F}_{IJ_s}$ and $\mathcal{G}_{IJ_s}$ are the numerical inviscid and viscous fluxes along side IJ_s, and B_i is the boundary integral. The side weight $\boldsymbol{\eta}_{IJ_s}$ is given by the summation of the two dual median lengths around the side times their normals. For example, the weight of the side connecting nodes I and J_1 is given by

$$\boldsymbol{\eta}_{IJ_1} = -\boldsymbol{\eta}_{J_1I} = \boldsymbol{AB} + \boldsymbol{BC} \tag{9}$$

In three-dimensional discretization, the resulting weights are the summation of the areas times their normal over the cell faces resulting from connecting the centroids of the cells and the middle points of the sides. The four-sided areas, resulting in hexahedral and prismatic cells, are calculated by dividing them to triangular faces in a consistent manner for the neighboring cells such that the conservation property is assured. The resulting numerical scheme is second-order accurate in space for tetrahedral meshes. For prismatic and hexahedral cells, the scheme is still second-order accurate for regular cells with right angles. In the worst case of a highly skewed cell, the scheme will reduce to first-order accuracy.[19] However, hexahedral meshes are usually generated in boundary layers where the generation of regular cells to ensure orthogonality is relatively straightforward. Similarly, prismatic cells are usually generated in a structured manner by projecting triangular meshes on radial layers and then connecting them. Highly skewed meshes are unlikely to occur in such situations except when dealing with detailed geometric features such as the inclusion of fillet radii at blade roots, part-span shrouds, etc.

Inviscid Fluxes

The inviscid fluxes in Eq. (8) are expressed using a central difference formulation with a suitable artificial dissipation that is required to stabilize the scheme. Thus, the inviscid fluxes can be written as

$$\mathcal{F}_{IJ_s} = \frac{\boldsymbol{\eta}_{IJ_s}}{\left|\boldsymbol{\eta}_{IJ_s}\right|}\cdot\left(\boldsymbol{F}_I + \boldsymbol{F}_{J_s}\right) - \mathcal{D}_{IJ_s} \tag{10}$$

where $\mathcal{D}_{IJ_s}$ is the artificial dissipation along the side IJ_s. The artificial dissipation is based on an upwind scheme developed by Swanson and Turkel[20] and Jorgenson and Turkel.[21] The scheme consists of a mixture of second- and fourth-order artificial viscosity. The fourth-order terms ensure the stability of the scheme in smooth flow regions. The second-order terms are required to damp numerical oscillations in the vicinity of discontinuities where the scheme reverts to first order using a pressure-based sensor. The artificial dissipation can be written as

$$\mathcal{D}_{IJ_s} = \left|A_{IJ_s}\right|\left[\phi\left(\boldsymbol{U}_{J_s} - \boldsymbol{U}_I\right) - \epsilon_4(1-\phi)\left(\mathcal{L}_{J_s} - \mathcal{L}_I\right)\right] \tag{11}$$

where A_{IJ_s} is the standard Roe matrix,[22] $\epsilon_4 \approx \frac{1}{8}$ is the fourth-order artificial dissipation coefficient, and $\mathcal{L}$ is a pseudo-Laplacian operator:

$$\mathcal{L}(\boldsymbol{U}_I) = \left(\sum_{s=1}^{n_s}\frac{\boldsymbol{U}_{J_s} - \boldsymbol{U}_I}{S_{IJ_s}}\right)\left(\sum_{s=1}^{n_s}\frac{1}{S_{IJ_s}}\right)^{-1} \tag{12}$$

where $S_{IJ_s} = |\boldsymbol{x}_{J_s} - \boldsymbol{x}_I|$. Moreover, ϕ is the second-order flux limiter that is required to enforce the monotonicity of the scheme given by

$$\phi = \max\left(\phi_I, \phi_{J_s}\right) \tag{13}$$

where

$$\phi_I = \frac{2\left| p_{J_s} - p_I - \boldsymbol{S}_{IJ_s}(\nabla p)_{J_s} \right|}{(1-\omega)\left[\left| p_{J_s} - p_I \right| + \left| p_I - p_{J_s} + 2(\nabla p)_{J_s}\boldsymbol{S}_{IJ_s} \right|\right] + 2\omega\left[p_{J_s} + p_I + \boldsymbol{S}_{IJ_s}(\nabla p)\right]} \tag{14}$$

where $\omega \approx 0.5$. Here, ϕ_{J_s} can similarly be obtained from Eq. (14) by substituting J_s for I.

Viscous Fluxes

The viscous fluxes are treated within the same edge-based data structure framework, provided the gradients of the primitive variables are known at the mesh nodes. Using the edge weights of Eq. (9), these gradients can be calculated from the formula

$$\left(\boldsymbol{\Omega}\frac{\partial \theta}{\partial x_j}\right)_I = \sum_{s=1}^{n_s} \frac{1}{2}\eta_{IJ_s}\left(\theta_I + \theta_{J_s}\right) + B_i \tag{15}$$

where θ represents a generic primitive variable and B_i is the boundary integral arising from the contributions of the boundary faces at the domain boundaries. This formulation results in a viscous flux scheme that uses information from two layers of points surrounding the point under consideration. The choice of this scheme is purely for computational efficiency and storage economy. The use of a finite element approach will require the storage of nine additional quantities per side. In any case, numerical experiments show that there is negligible difference between the two approaches.

Implicit Temporal Discretization

Equation (8) can be expressed in the form

$$\frac{\mathrm{d}(\boldsymbol{\Omega}_I \boldsymbol{U}_I)}{\mathrm{d}t} = \boldsymbol{R}(\boldsymbol{U}) \tag{16}$$

A second-order implicit backward time integration of Eq. (16) can be expressed as

$$\frac{3(\Omega \boldsymbol{U})_I^{n+1} - 4(\Omega \boldsymbol{U})_I^n + (\Omega \boldsymbol{U})_I^{n-1}}{2\Delta t} = \boldsymbol{R}(\boldsymbol{U}^{n+1}) \tag{17}$$

where n denotes the time level. The implicit nonlinear system of equations given by Eq. (17) needs to be solved every time step. An iterative equation is constructed from Eq. (17) by simply adding a pseudo-time derivative term $\boldsymbol{U}_\tau$ to the left-hand side:

$$\boldsymbol{U}_\tau + \frac{3(\Omega \boldsymbol{U})_I^{n+1} - 4(\Omega \boldsymbol{U})_I^n + (\Omega \boldsymbol{U})_I^{n-1}}{2\Delta t} = \boldsymbol{R}(\boldsymbol{U}^{n+1}) \tag{18}$$

Indicating with $\boldsymbol{U}^m$ the mth approximation to $\boldsymbol{U}^{n+1}$, Eq. (19) can be rewritten as

$$\Omega_I^{n+1}\left(\frac{1}{\Delta\tau} + \frac{3}{2}\frac{1}{\Delta t}\right)\Delta \boldsymbol{U}_I + \frac{3\Omega_I^{n+1}\boldsymbol{U}_I^m - 4(\Omega \boldsymbol{U})_I^n + (\Omega \boldsymbol{U})_I^{n-1}}{2\Delta t} = \boldsymbol{R}^{m+1} \tag{19}$$

where $\Delta \boldsymbol{U}_I = \boldsymbol{U}_I^{m+1} - \boldsymbol{U}_I^m$, with $\Delta\tau$ representing the pseudo-time step. Linearizing the right-hand side of Eq. (19) around $\boldsymbol{U}_I^m$, the pseudo-time integration, which advances the solution from t^n to t^{n+1}, becomes

$$\left(\frac{\Omega_I^{n+1}}{\Delta\tau} + \frac{3}{2}\frac{\Omega_I^{n+1}}{\Delta t} - \boldsymbol{J}^m\right)\Delta \boldsymbol{U}_I = \boldsymbol{R}^m - \frac{3}{2}\frac{\Omega_I^{n+1}}{\Delta t}\boldsymbol{U}_I^m - \boldsymbol{E}_I^n \tag{20}$$

where $\boldsymbol{E}_I^n$ involves the portion of the physical time derivative at previous time steps and is invariant during the iteration process:

$$\boldsymbol{E}_I^n = \frac{4(\Omega \boldsymbol{U})_I^n - (\Omega \boldsymbol{U})_I^{n-1}}{2\Delta t} \tag{21}$$

The left-hand side of Eq. (20) contains a portion of the physical-time derivative to reduce the pseudo-time step in regions of the flow where the ratio pseudo-/physical-time step $\Delta\tau/\Delta t$ becomes large.[23] This equation can be solved using conventional algorithms. Here, the procedure used consists of a two-point backward implicit difference and an expansion of the residual about the nth time level.[24] Where $\Delta \boldsymbol{U}_I^{n+1} = \boldsymbol{U}_I^{n+1} - \boldsymbol{U}_I^n$, a two-point backward implicit difference of Eq. (20) can be written as

$$\bar{\boldsymbol{J}}_D \Delta \boldsymbol{U}_I^{n+1} + \sum_{s=1}^{m_I} \bar{\boldsymbol{J}}_{\mathrm{OD}_s} \Delta \boldsymbol{U}_{J_s}^{n+1} = \boldsymbol{R}_I^n \tag{22}$$

where $\bar{\boldsymbol{J}}_D$ is the block diagonal contribution to the Jacobian matrix and $\bar{\boldsymbol{J}}_{\mathrm{OD}_s}$ is the off-diagonal contribution of node J_s. Equation (22) represents a sparse linear system that needs to be solved at each time level n. A defect-correction procedure is used to solve Eq. (22), where the left-hand side sparse matrix is discretized using a lower-order approximation (i.e., first order in space) because of storage considerations and computational complexity and also because the resulting lower-order matrix is better conditioned than the higher order matrix. Thus, the resulting method is only moderately implicit. The sparse linear system (22) is solved using a point Jacobi iterative method. Where $\boldsymbol{\Phi}_I^l$ is the lth approximation to $\Delta \boldsymbol{U}_I^{n+1}$, the iterative method can be written as

$$\bar{\boldsymbol{J}}_D \boldsymbol{\Phi}_I^{l+1} = \boldsymbol{Q}_I^l \tag{23}$$

$$\boldsymbol{Q}_I^l = \boldsymbol{R}_I^n - \sum_{s=1}^{m_I} \bar{\boldsymbol{J}}_{\mathrm{OD}_s} \boldsymbol{\Phi}_{J_s}^l \tag{24}$$

$$\lim_{l \to \infty} \Phi^l = \Delta \boldsymbol{U}^{n+1} \tag{25}$$

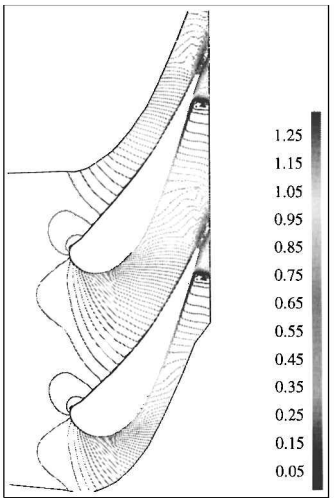
This formulation requires the storage of a 5×5 block diagonal matrix only. The matrix is computed once only, at the beginning of the numerical integration. Local time stepping is also employed to speed up the convergence of inner iterations.

Time accuracy is guaranteed by the outer iteration level where the time step is fixed throughout the solution domain, whereas the inner iteration procedure can be performed using traditional acceleration techniques such as local time stepping and residual smoothing. The same time stepping procedure is applied to the turbulence model equation, which is solved separately every time step.

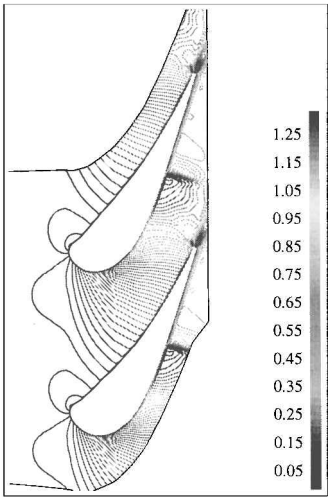
Boundary Conditions

This section describes the numerical treatment of the inflow and outflow boundaries for turbomachinery flow calculations. Whereas the far-field boundaries for isolated airfoils can be taken many chords away, the boundaries are typically less than one chord away for most turbomachinery applications. This situation may lead to computational inaccuracies if the boundary conditions are not suitably formulated. Various techniques have been developed to minimize the reflection of the outgoing waves[25–28] and an overview is given by Givoli.[29] Here two different set of treatments, one for steady-state computations and the other for unsteady computations, are used. The steady-state boundary treatment is based on the characteristics of the Euler equations. In particular, the steady-state boundary conditions are obtained from the linear, harmonic unsteady nonreflecting boundary conditions, as a limiting case of zero-frequency unsteadiness. The resulting nonreflecting boundary conditions are exact for linear solutions at the far-field boundary. The treatment of such boundary conditions for two-dimensional turbomachinery applications can be found in Ref. 30. An extension to three dimensions is reported in Ref. 28.

In this work, the effectiveness of the boundary treatment will be demonstrated for the steady-state flow past a high-pressure nozzle-guide vane (NGV). Figure 2a shows the Mach number contours for



a) Quasi-three-dimensional nonreflective boundary conditions



b) Standard one-dimensional boundary conditions

Fig. 2 Comparison of steady-state Mach number contours at 20% span.

a radial section at 20% span using the quasi-three-dimensional nonreflective boundary conditions of Ref. 31, whereas Fig. 2b shows the flow obtained with the standard one-dimensional boundary conditions of Refs. 32 and 33. The contours look very similar before the throat region of the passage because the information coming from the outflow plane does not affect this part of the domain. The two solutions differ considerably on the diverging part of the passage, especially around the shock position. This noticeable mismatch is caused because the one-dimensional boundary conditions generate nonphysical reflected acoustic waves that arise from imposing a constant exit pressure along the circumferential direction. The approach used in Fig. 2a imposes an average value of the exit pressure only, letting the other circumferential harmonics obey the nonreflective equations.

Two further types of boundary conditions are needed for turbomachinery calculations: solid wall and periodicity. On solid walls, the pressure is extrapolated from the interior points and the slip (inviscid) or no-slip (viscous) conditions are used to compute the other quantities. An extra boundary condition for the heat flux is employed for viscous flow calculations. Using the edge-based data structure, the periodicity is handled in a straightforward way as long as the points in the two periodic boundaries are located at same axial and radial coordinates.

In general, for turbomachinery applications, the physical boundary conditions employed at the inflow boundary are total pressure, total temperature, and flow angle, whereas static pressure is used at the outflow. On solid walls, two possible approaches are implemented: either no-slip boundary conditions, where the turbulence model and flow equations are solved up to the wall, or slip boundary conditions, where the standard wall functions are used. An adiabatic wall is used as a thermal boundary condition.

Case Studies

As part of a long-term turbomachinery aeroelasticity program, a Favre-averaged Navier–Stokes code, AU3D, was developed using the methodology of the second and third sections. The code deals with steady and unsteady flows and can also accommodate blade vibration through moving meshes. The AU3D code was validated for a wide range of steady and unsteady flows against available experimental and analytical data. The overall agreement was considered to be good. Two test cases will be presented here. The first one is a steady flow analysis for a rig fan blade, and the other one is rotor/stator interaction study. Experimental data were available in

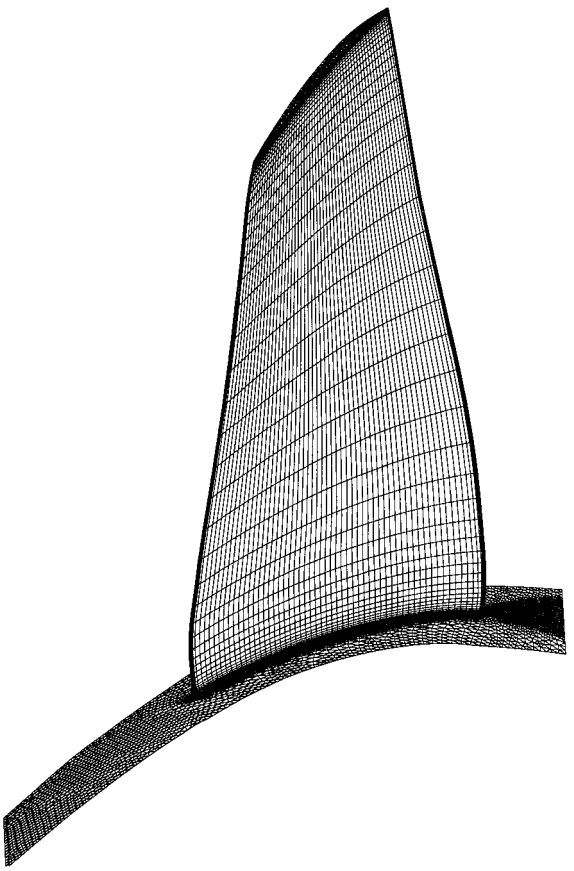


Fig. 3 Fan blade mesh with 250,000 points.

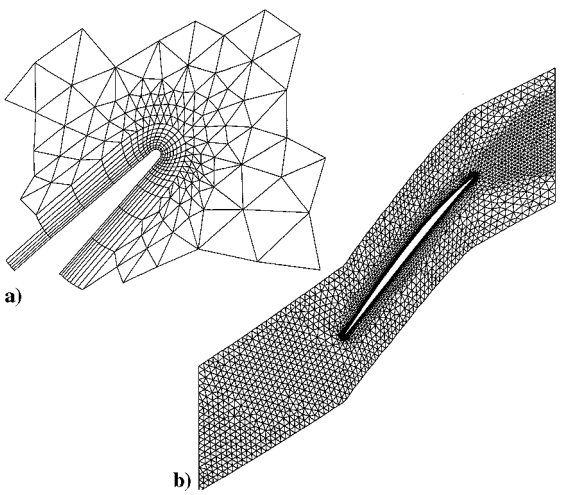


Fig. 4 a) Zoom at trailing edge; b) fan blade mesh at midsection.

both cases, and various comparisons will be made with the predicted results.

Fan Blade

A rig fan blade running at its rated speed on the nominal working line will be studied first. As discussed, a semistructured mesh was generated for flow discretization, and a mesh smoothing procedure was used for each of the mapped radial layers to minimize grid skewness. The smoothing procedure includes the movement of the points around leading and trailing edges to preserve a good resolution of the blade curvature in those regions. Three meshes, with 250,000, 500,000, and 750,000 points, were used in the steady flow analysis of the blade. Views of the first mesh on the blade surface and at the hub section are shown in Fig. 3. Figure 4 shows the same mesh at midsection. The advantages of the semistructured meshing strategy can be seen from a study of Figs. 3 and 4. A good resolution of the boundary layer is obtained by generating an O-type grid that blends smoothly to the unstructured part away from the boundary layer. In the radial direction, where the flow variation is much less than that in the axial and circumferential directions, the mesh density can be adjusted easily by virtue of being structured in that direction while still keeping a good resolution of the leading and trailing edges. Furthermore, the approach naturally allows the addition of a tip gap mesh by triangulating the blades tip and mapping extra layers over the tip. Figure 5 shows a cut across the blade showing the tip gap mesh. Typically, 5–10 layers of cells, for example, about 4,000 points for the second mesh, are used to discretize the tip gap region.

The solution for the first grid was obtained using wall functions, whereas the flow equations were solved up to the wall for the second and third meshes. It was found out that changes in the solution were negligible for meshes finer than the second mesh. Thus, the second and third solutions can be considered to be mesh converged.

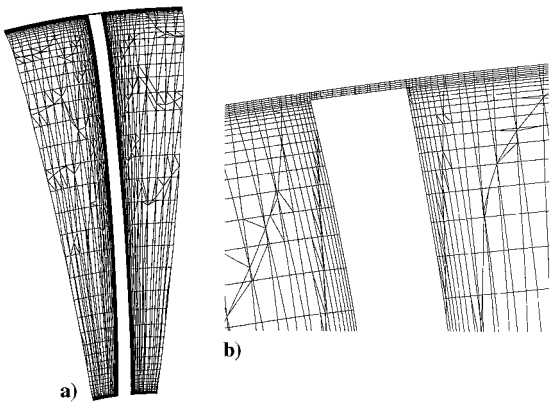
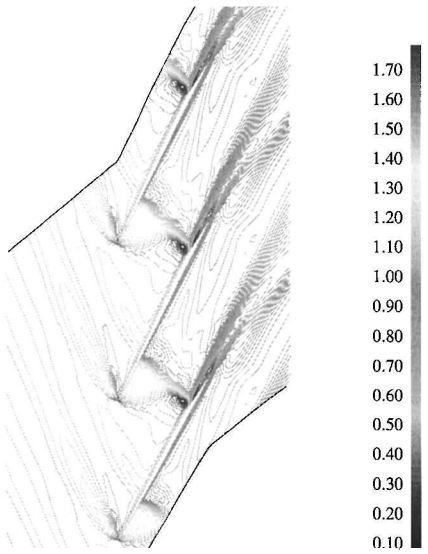
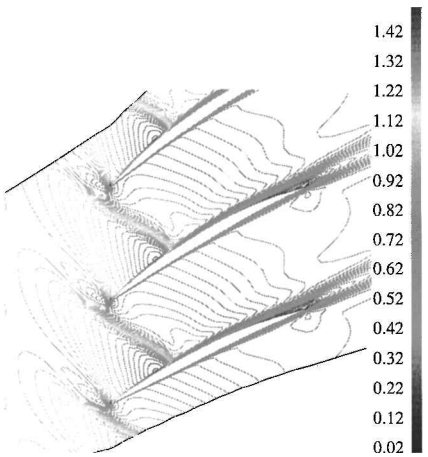


Fig. 5 a) Vertical cut in the fan blade mesh; b) zoom showing tip gap mesh.



a) 90% span



b) Midsection

Fig. 6 Mach number contours.

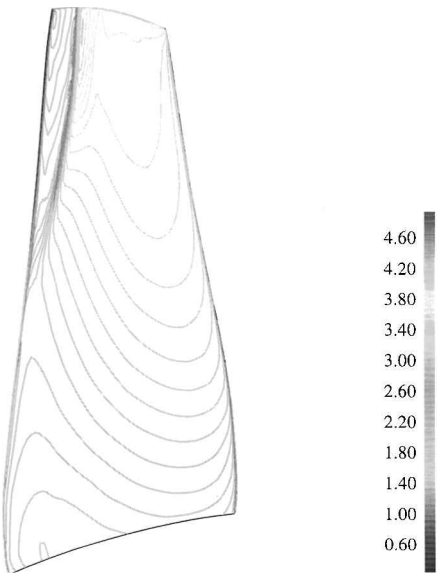


Fig. 7 Pressure contours at pressure side of the blade.

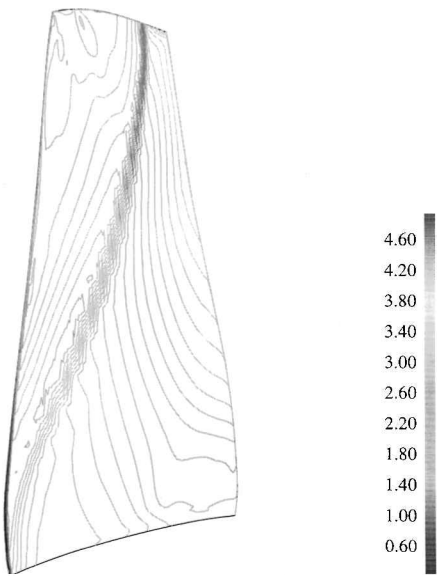


Fig. 8 Pressure contours at suction side of the blade.

The results will be presented for the second mesh and compared to those of the first mesh in some cases. Figure 6 shows the Mach number contours at 90% and 50% blade span, whereas Figs. 7 and 8 show the pressure contours on the suction and pressure sides, respectively. The flow features are typical of a fan operating close to its rated speed.[2] Near the tip section, a lambda shock structure exists where the bow shock at the leading edge interacts with the passage shock on the pressure side. The shock on the suction side is curved toward the trailing edge and interacts with the boundary layer, a feature resulting in a much thicker boundary layer behind the shock. Farther down the blade, the suction side shock moves forward until it gets expelled outside the passage near the midsection. Figure 9a shows a velocity vector plot on the pressure side of the blade. It can be seen that there is a flow migration up the blade behind the shock. Figure 9b shows the details of the flow near the tip. There is a change of direction behind the shock that is due to the flow migration. Figure 9c shows the details of the trailing-edge secondary flow near the hub section, another typical feature of a fan blade at its rated speed. Additionally, Fig. 10 shows the Mach number contours at the midlayer in the tip gap. It can be seen that both the tip vortex and the leading-edge vortex are resolved resonably well.

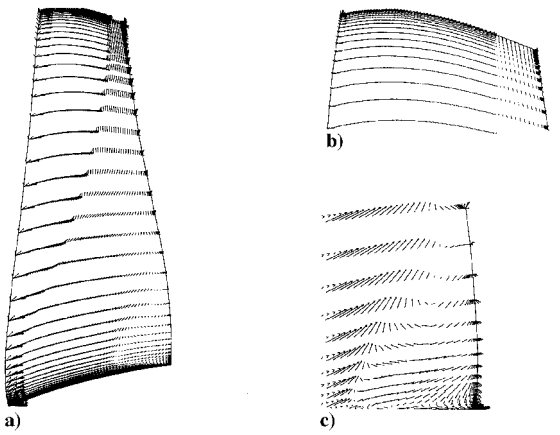


Fig. 9 Vector plots at suction side.

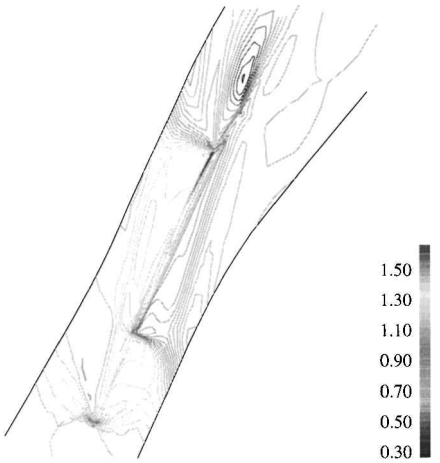


Fig. 10 Mach number contours in the tip gap region.

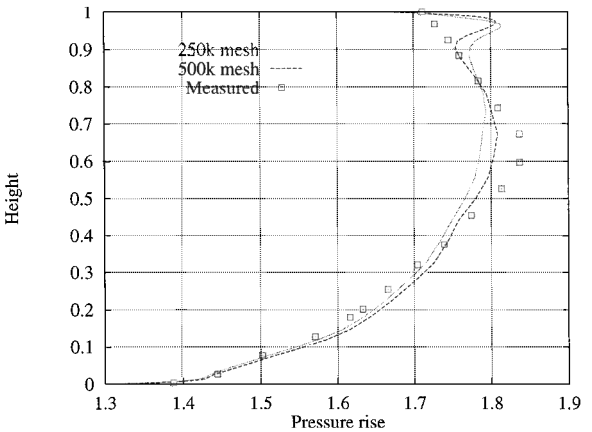


Fig. 11 Predicted and measured radial profiles of pressure rise at outflow.

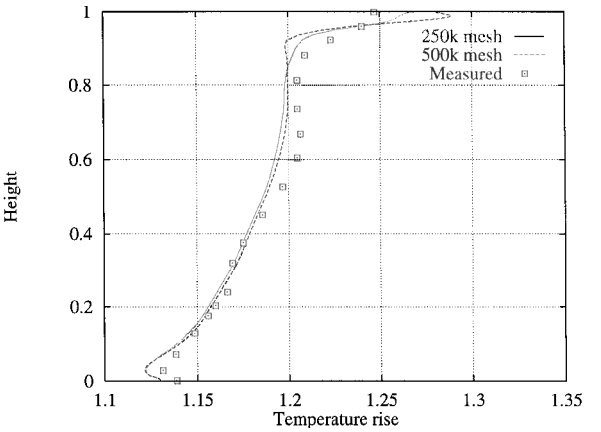


Fig. 12 Predicted and measured radial profiles or temperature rise at outflow.

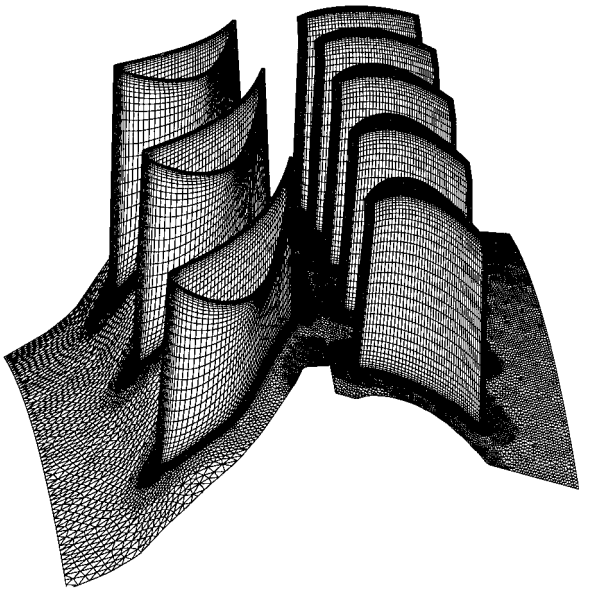


Fig. 13 Sector mesh for rotor/stator interaction calculations.

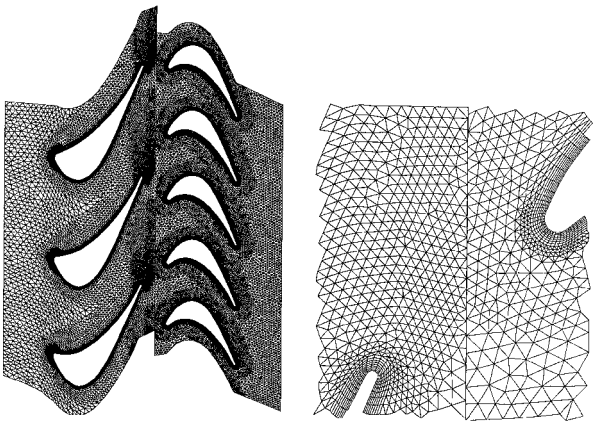


Fig. 14 Rotor/stator mesh at the hub section.

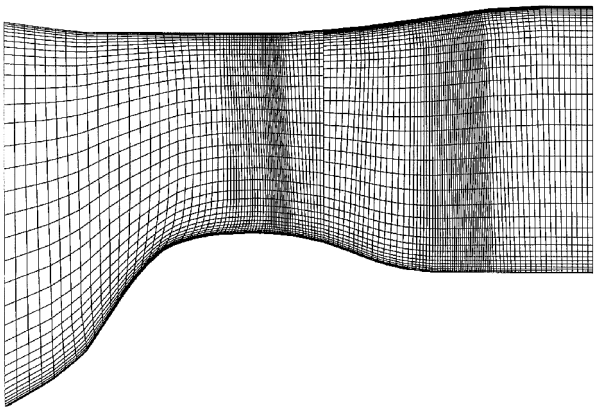


Fig. 15 Rotor/stator mesh at the periodic boundary.

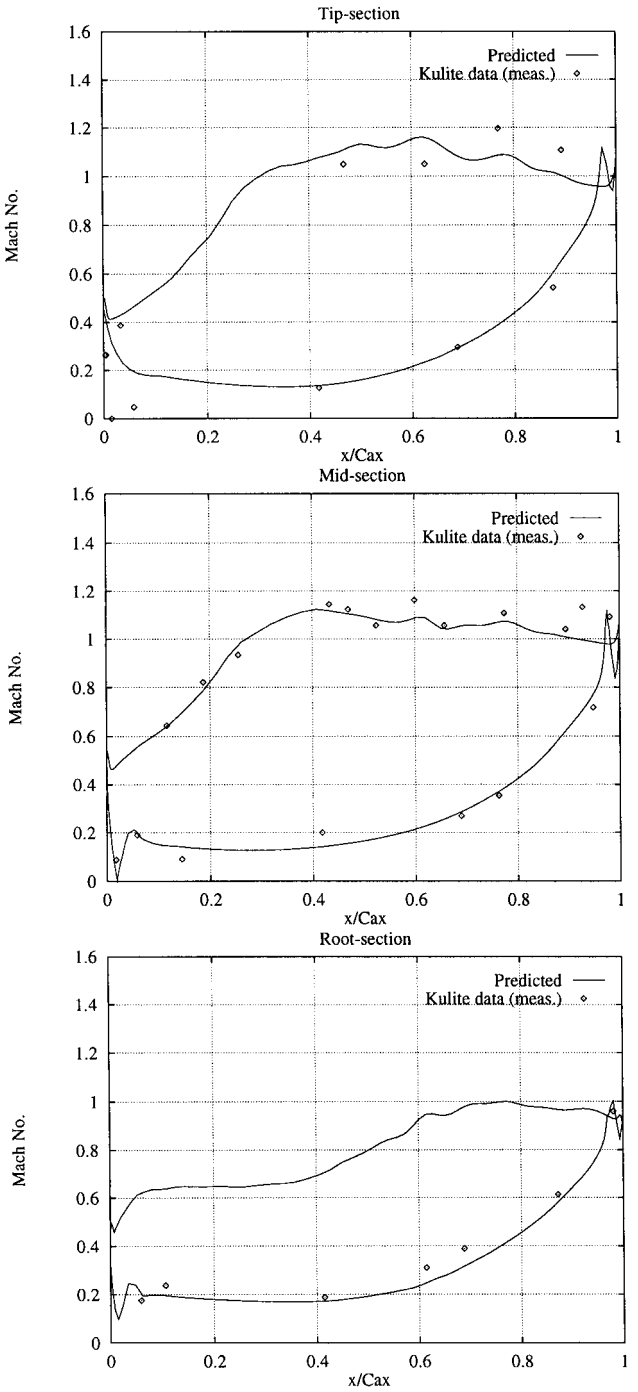


Fig. 16 Predicted and measured steady pressure profiles along the rotor blade chord.

The comparison with the experimental data was done by computing the standard performance parameters at the outflow section. The predicted and measured flow quantities were circumferentially averaged and plotted as a function of the radius. Two such quantities, pressure rise and temperature rise, are plotted in Figs. 11 and 12. The agreement with the experimental data is generally very good, the discrepancy near the tip section probably being due to the tip gap size, the value of which is difficult to estimate during the operation of the fan. Note that typically the grids used result in typical y^+ values of around 150, 100, and 70, repcetviely, for the three meshes used. The computer time required to obtain a steady-state flow solution for the 500,000 point mesh is about 8 CPU h on a 266 MHz Linux personal computer. Convergence was achieved by a drop of 5 orders of magnitude in the residual values.

Rotor/Stator Interaction

A coupled rotor/stator interaction case was studied next using a well-known turbine test rig geometry, RT27, for which extensive steady and unsteady flow measurements were conducted at the Osney Laboratory, Oxford.[34,35] Because there are 36 stator and 60 rotor blades, a periodic sector of 2 stator and 5 rotor passages was selected without having to change the blade numbers or to conduct a whole-annulus calculation.

Again a semistructured mesh was generated for one stator and one rotor passage, and the sector mesh was assembled by duplicating these passages. A view of the three-dimensional mesh, which contains about 10^6 points, is shown in Fig. 13. The mesh at the hub section is shown in Fig. 14 and that at the periodic boundary is shown in Fig. 15. The rotor blades are unshrouded, and, thus, a tip gap mesh was also included. A steady-flow analysis was performed for isolated stator and a rotor passages at first. The boundary conditions were provided by the circumferentially averaged measured values. The boundary conditions for the rotor inflow

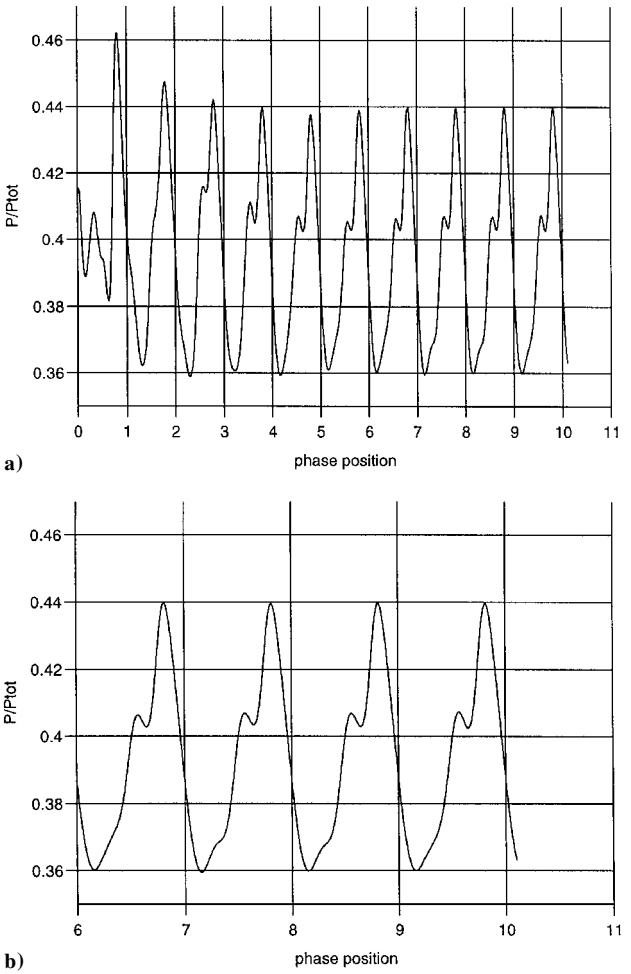


Fig. 17 a) Convergence to periodic state; b) zoom of the last part of panel a.

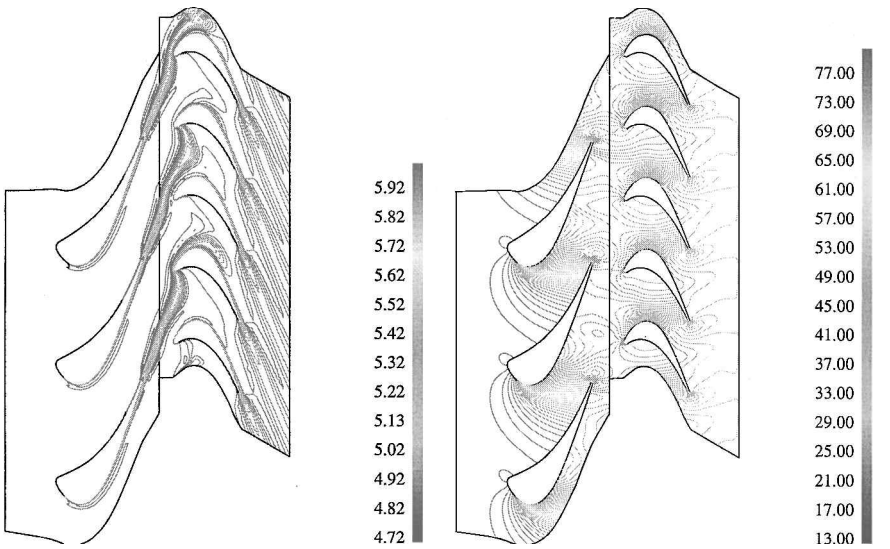


Fig. 18 Entropy and pressure contours at midsection.

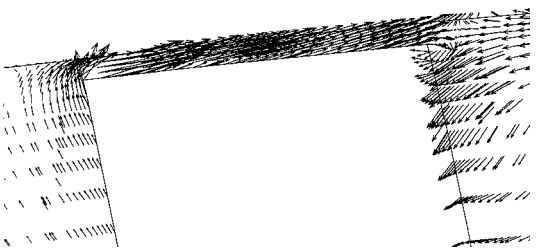


Fig. 19 Vector plot across the rotor blade normal to axial direction.

steady-state calculations were obtained by averaging the values obtained from the steady-state solution for the stator. Figure 16 shows a comparison between the predicted and measured steady pressure profiles around the rotor blade at three different positions: near the hub, midspan, and near the tip. Good agreement is observed for all three cases.

The steady flow solution was used as a starting point for the unsteady computations. At the interface boundary, the information was exchanged between the stator and rotor domains using an interpolation procedure that preserves linearity in a simillar fashion to the multigrid interpolation algorithm of Ref. 36. Unlike for their structured counterparts,[3] the implementation of a conservative exchange of information between two general unstructured grids can be difficult and expensive. However, for the type of semistructured grids used here, it is possible to perform a pseudo-three-dimensional conservative exchange of information by considering data from each radial layer separately. Such an approach can be justified for most turbine flows that are predominantly in the axial direction.

To ensure convergence to periodic state, the unsteady flow analysis was run for 12 cycles (Fig. 17). Figure 18 shows instantaneous entropy and pressure contours at the midsection, from which it can be seen that the flow is subsonic. The localized effect of the tip gap can be seen from Fig. 19, where the secondary flow vectors are plotted at sections normal to the axial direction. It can be seen that the leakage flow causes a small secondary flow recirculation bubble that extends to a small fraction of the blade span only. The predicted and measured unsteady time histories are shown in Fig. 20. The Fourier transform of the time history of Fig. 19 is given in Figs. 21 and 22, and the predicted and measured unsteady pressure time histories are compared in Figs. 19, 23, and 24. The predicted and measured unsteady time histories near the hub section and pressure surface are shown in Fig. 25. It can be seen that there is good overall agreement with the experimental data. The computer time required to obtain a periodically converged solution was about 120 CPU h on a 266 MHz Linux personal computer. The single passage steady-state solutions were obtained in about 4 CPU h for both the stator and rotor blades using single-passage meshes.

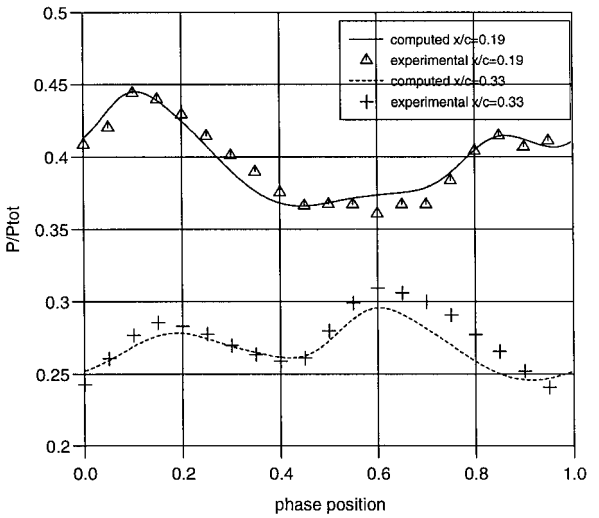


Fig. 20 Predicted and measured unsteady time histories: midheight suction surface.

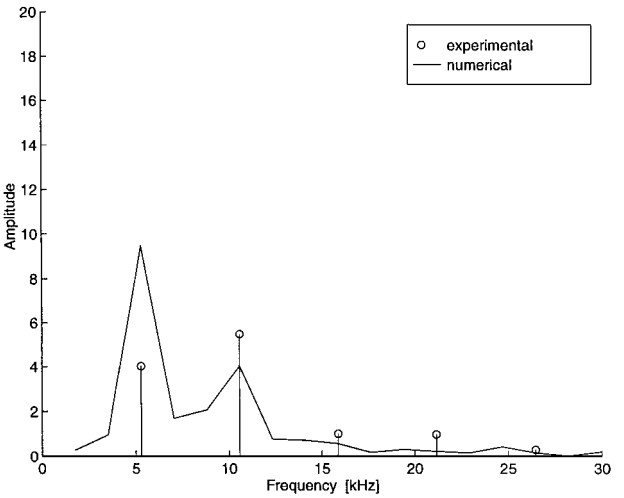


Fig. 21 Harmonic content of the unsteady pressure in Fig. 20: fast Fourier transform (FFT) of unsteady total pressure, $x/c = 0.19$.

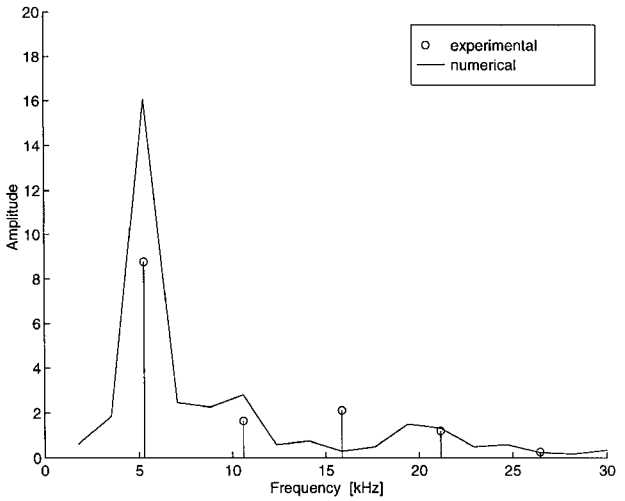


Fig. 22 Harmonic content of the unsteady pressure in Fig. 20: FFT of unsteady total pressure, $x/c = 0.33$.

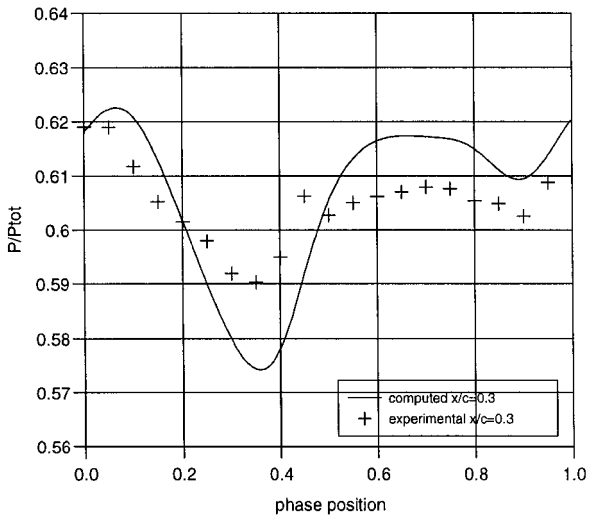


Fig. 23 Predicted and measured unsteady time histories: midheight pressure surface.

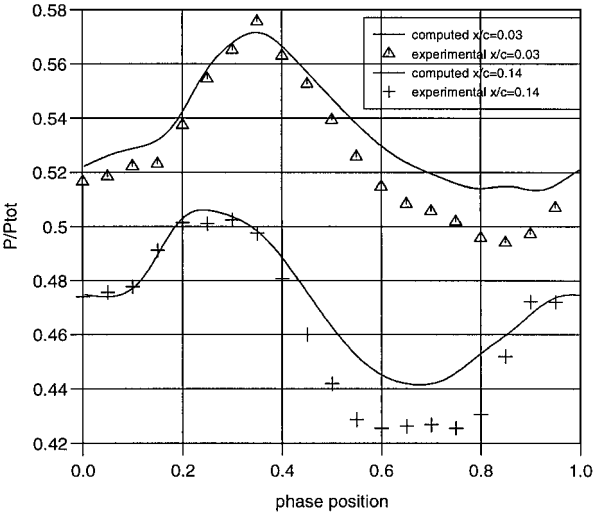


Fig. 24 Predicted and measured unsteady time histories: near tip, suction surface.

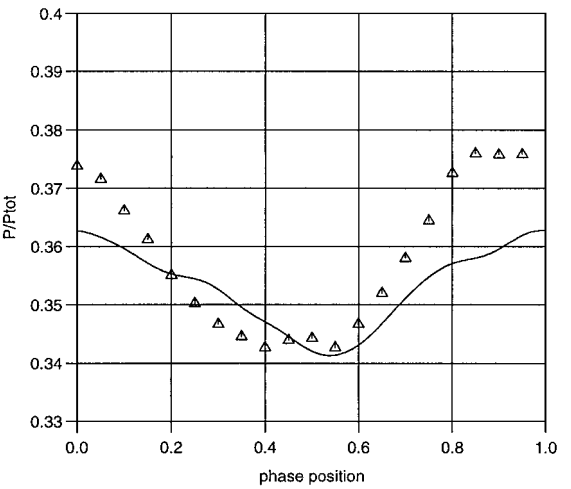


Fig. 25 Predicted and measured unsteady time histories near hub section and pressure surface: ——, computed $x/c = 0.8$, and ▵, experimental $x/c = 0.8$.

Conclusion

1) A numerical model for the steady and unsteady flow analyses of turbomachinery blades was presented. The model uses unstructured hybrid grids and has an edge-based data structure. Implicit dual time stepping is employed during the unsteady computations. A further possibility is the implementation of a multigrid scheme, but more work is needed to devise efficient and robust implementations to unsteady viscous flows with deforming grids required for aeroelasticity computations.

2) Two test cases are presented to validate the model against experimental data: steady-state flow in transonic fan and unsteady flow in turbine rotor/stator interaction. In both cases, good agreement with experimental data was obtained.

3) The model is found to be computationally efficient, and it will be possible to undertake large unsteady viscous flow simulations with current computing power. A natural progression of the work is the consideration of moving meshes so that aeroelasticity computations can also be undertaken. Parallelization on distributed memory machines is currently in progress, and it is expected that whole-annulus multibladerow computations will become commonplace before the end of the century. Such issues will be discussed in a forthcoming paper.

Acknowledgments

The authors would like to thank Rolls–Royce plc for both sponsoring this work and allowing its publication. Thanks are also due to the European Commission for a research studentship under the Training and Mobility of Researchers (Marie Curie) scheme. They also acknowledge the support of the Defence Research Agency (Pyestock), the Ministry of Defence and the Department of Trade and Industry for the measurements of Rig RT27 data. They also thank R. Ainsworth and R. Moss for providing the RT27 test data.

References

[1] Dawes, W. N., "Development of a Three-Dimensional Navier–Stokes Solver for Application to all Types of Turbomachinery," American Society of Mechanical Engineers, Paper 88-GT-70, 1988.

[2] Arnone, A., "Multigrid Methods for Turbomachinery Navier–Stokes Calculations," *Solution Techniques for Large-Scale CFD Problems*, edited by W. G. Habashi, Wiley, London, 1995, pp. 293–332.

[3] Rai, M. M., "Navier–Stokes Simulations of Rotor/Stator Interaction Using Patched and Overlaid Grids," *Journal of Propulsion and Power*, Vol. 3, No. 5, 1987, pp. 387–396.

[4] Rai, M. M., and Madavan, N. K., "Multi-Airfoil Navier–Stokes Simulations of Turbine Rotor–Stator Interaction," *Journal of Turbomachinery*, Vol. 112, 1990, pp. 377–384.

[5] Barth, T. J., "On Unstructured Grid Solvers," TR Lecture Series, Von Kármán Inst., VKI-LS 1990-03, 1990.

[6] Barth, T. J., "Numerical Aspects of Computing High-Reynolds-Number Flows on Unstructured Meshes," AIAA Paper 91-0721, Jan. 1991.

[7]Frink, N. T., "Assessment of an Unstructured-Grid Method for Predicting Three-Dimensional Turbulent Viscous Flows," AIAA Paper 96-0292, Jan. 1996.

[8]Mavriplis, D. J., "Three-Dimensional Multigrid Reynolds-Averaged Navier–Stokes Solver for Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 3, 1995, pp. 445–453.

[9]Peraire, J., Morgan, K., Vahdati, M., and Peiró, J., "The Construction and Behavior of Some Unstructured Grid Algorithms for Compressible Flows," *ICFD Conference on Numerical Methods for Fluid Dynamics*, Oxford Univ. Press, Oxford, 1992, pp. 221–229.

[10]Venkatakrishnan, V., "A Perspective on Unstructured Grid Flow Solvers," Inst. for Computer Applications in Science and Engineering, TR 95-3, Hampton, VA, 1995.

[11]Holmes, D. G., and Chuang, H. A., "2D Linearised Harmonic Euler Flow Analysis for Flutter and Forced Response," *Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines and Propellers*, edited by H. M. Atassi, Springer–Verlag, New York, 1993, pp. 341–353.

[12]Dawes, W. N., "The Extension of a Solution-Adaptive 3D Navier–Stokes Solver Towards Geometries of Arbitrary Complexity," American Society of Mechanical Engineers, Paper 92-GT-363, June 1992.

[13]Dawes, W. N., "Simulating Unsteady Turbomachinery Flows on Unstructured Meshes Which Adapt Both in Time and Space," American Society of Mechanical Engineers, Paper 93-GT-104, May 1993.

[14]Vahdati, M., and Imregun, M., "Non-Linear Aeroelasticity Analyses Using Unstructured Dynamic Meshes," *18th International Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, edited by Y. Tanida and M. Namba. Elsevier, New York, 1995, pp. 177–190.

[15]Sbardella, L., Sayma, A. I., and Imregun, M., "Semi-Unstructured Mesh Generator for Flow Calculations in Axial Turbomachinery Blading," *18th International Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, edited by T. Franson, Kluwer, Dordrecht, The Netherlands, 1997, pp. 541–554.

[16]Baldwin, B. S., and Barth, T. J., "One-Equation Turbulence Transport Model for High-Reynolds-Number Wall-Bounded Flows," AIAA Paper 91-0610, 1991.

[17]Spalart, P. R., and Allmaras, S. R., "One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1991.

[18]Gibson, M. M., and Dafa'Alla, A. A., "Two-Equation Model for Turbulent Flow," *AIAA Journal*, Vol. 33, No. 8, 1994, pp. 1514–1522.

[19]Essers, J. A., Delanaye, M., and Rogiest, P., "Upwind-Biased Finite Volume Technique Solving Navier–Stokes Equations on Irregular Meshes," *AIAA Journal*, Vol. 33, No. 5, 1995, pp. 833–842.

[20]Swanson, R. C., and Turkel, E., "On Central-Difference and Upwind Schemes," *Journal of Computational Physics*, Vol. 101, No. 2, 1992, pp. 292–306.

[21]Jorgenson, P. C., and Turkel, E., "Central Difference TVD Schemes for Time Dependent and Steady State Problems," *Journal of Computational Physics*, Vol. 107, No. 2, 1993, pp. 297–308.

[22]Roe, P., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[23]Melson, N. D., Sanetrik, M. D., and Atkins, H. L., "Time-Accurate Navier–Stokes Calculations with Multigrid Acceleration," *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, Univ. of Colorado, Boulder, CO, 1993.

[24]Vahdati, M., Morgan, K., and Praire, J., "The Computation of Viscous Compressible Flows Using an Upwind Algorithm and Unstructured Meshes," *Computational Nonlinear Mechanics in Aerospace Engineering*, edited by W. Habashi, Progress in Aeronautics and Astonautics, AIAA, Washington, DC, 1992, pp. 479–505.

[25]Engquist, B., and Majda, A., "Absorbing Boundary Conditions for the Numerical Simulation of Waves," *Mathematics of Computation*, Vol. 31, No. 139, 1977, pp. 629–651.

[26]Higdon, R. L., "Absorbing Boundary Conditions for Difference Approximations to the Multi-Dimensional Wave Equation," *Mathematics of Computation*, Vol. 47, No. 176, 1986, pp. 437–459.

[27]Ferm, L., "Non-Reflecting Boundary Conditions for the Steady Euler Equations," *Journal of Computational Physics*, Vol. 122, No. 2, 1995, pp. 307–316.

[28]Giles, M. B., "Nonreflecting Boundary Conditions for the Euler Equations Calculations," *AIAA Journal*, Vol. 28, No. 12, 1990, pp. 2050–2058.

[29]Givoli, D., "Non-Reflecting Boundary Conditions," *Journal of Computational Physics*, Vol. 94, No. 1, 1991, pp. 1–29.

[30]Giles, M. B., "Non-Reflecting Boundary Conditions for the Euler Equations," Computational Fluid Dynamics Lab., TR TR-88-1, Massachusetts Inst. of Technology, Cambridge, MA, 1988.

[31]Saxer, A. P., and Giles, M. B., "Quasi-Three-Dimensional Nonreflecting Boundary Conditions for Euler Equations Calculations," *Journal of Propulsion and Power*, Vol. 9, No. 2, 1993, pp. 263–271.

[32]Thompson, K. W., "Time Dependent Boundary Conditions for Hyperbolic Systems," *Journal of Computational Physics*, Vol. 68, No. 1, 1987, pp. 1–24.

[33]Thompson, K. W., "Time Dependent Boundary Conditions for Hyperbolic Systems, II," *Journal of Computational Physics*, Vol. 89, 1990, pp. 439–461.

[34]Moss, R., Ainsworth, R. W., Sheldrake, C. D., and Miller R., "The Unsteady Pressure Field Over a Turbine Blade Surface: Visualisation and Interpretation of Experimental Data," American Society of Mechanical Engineers, Paper 97-GT-474, June 1997.

[35]Moss, R., Sheldrake, C. D., Ainsworth, R. W., Smith, A. D., and Dancer, D., "Unsteady Pressure Transfer Measurements on a Rotating Blade Surface in a Transient Flow Facility," CP-571, AGARD, TR 22, 1995.

[36]Peraire, J., Peiró, J., and Morgan, K., "Finite Element Multigrid Solution of Euler Flows Past Installed Aero-Engines," *Computational Mechanics*, Vol. 11, No. 2, 1993, pp. 433–451.

J. Kallinderis
Associate Editor