AIAA JOURNAL
Vol. 49, No. 1, January 2011

Efficient Edge-Based Rotor/Stator Interaction Method

Manuel A. Burgos*
Universidad Politécnica de Cartagena. 30006 Murcia, Spain
and
Jesus Contreras† and Roque Corral‡
Industria de Turbopropulsores, S. A., 28830 Madrid, Spain

DOI: 10.2514/1.44512

A method to address the rotor/stator interaction problem on an edge-based Reynolds-averaged Navier–Stokes solver is presented. The implementation of the phase-lagged boundary conditions and a new original conservative discretization of the sliding plane in an unstructured solver is discussed in detail. The numerical method is implicit in time, and convergence is accelerated using a multigrid technique where the coarser grids are obtained by agglomeration. Results of the present method are compared with the experimental results obtained at the National Aerospace Laboratory (The Netherlands) for a subsonic fan.

Introduction

PROBABLY one of the largest problems related with the high-fidelity simulation of turbomachinery is the existence of a large number of highly packed blade rows. The axial distance between neighboring rows varies between 5 and 50% of the axial chord; therefore, the interaction among them may be very important. The acoustic field generated in this interaction is relevant from the point of view of noise emission [1] and high-cycle fatigue of the turbomachines [2]. Moreover, a number of complex aerodynamic phenomena, such as the laminar to turbulent transition at low Reynolds number regimes, depend as well on these unsteady interactions [3].

From an aerodynamic perspective, turbomachinery design is conducted following a hierarchical approach to take into account the interaction among different rows. First, throughflow methods that assume a circumferentially averaged flow are used to set up the basic aerodynamic parameters [4,5]. Then, steady radial distributions of boundary conditions are derived to set up three-dimensional (3-D) single-row simulations. Nowadays, multistage steady simulations, where single-row computations are coupled through circumferentially averaged boundary conditions, are often carried out [6,7]. Fully unsteady 3-D stator/rotor simulations are conducted in special situations but by no means may they be considered a routine design practice.

The simulation of stator/rotor interactions using nonlinear methods requires three main building blocks: an accurate and efficient unsteady solver, a rotor/stator interface plane, and the use of a method to deal with the dissimilar blade count of the rotor and stator rows. This paper deals with the last two problems in the context of unstructured grids. Although unstructured grids have received a great deal of research in external flows, it is only very recently that they have been employed in unsteady turbomachinery applications [8,9]. However, the treatment of the stator/rotor interface plane is poorly reported, especially in unstructured meshes. Phase-lagged boundary conditions are better documented [10–13], but their detailed validation against experimental data is not available or scarce. In this article, first the baseline numerical algorithm is outlined, then the treatment of the rotor/stator interface plane is discussed in detail. We finally close the paper briefly, describing the implementation of the phase-lagged boundary conditions and comparing the results obtained for a subsonic fan tested at the National Aerospace Laboratory (The Netherlands) (NLR) with the experimental data.

Presented as Paper 2007-4474 at the 18th AIAA Fluid Dynamics Conference,, Miami, FL, 25–28 June 2007; received 21 March 2009; revision received 1 May 2010; accepted for publication 9 August 2010.

*Department of Fluid Dynamics.
†Technology and Methods Department.
‡Technology and Methods Department; Associate Professor of Aerospace Engineering, School of Aeronautics, Universidad Politécnica de Madrid, 28040 Madrid, Spain.

Baseline Solver

This section describes the unstructured baseline solver, known as Mu^2s^2T, used in the present work as the departing point for the implementation of the rotor/stator interface plane and the phase-lagged boundary conditions.

The Navier–Stokes equations in conservative form for an arbitrary control volume may be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\mathbf{U}\,\mathbf{d}\Omega+\int_{\Sigma}\mathbf{F}(\mathbf{U})\cdot\mathrm{d}\mathbf{A}=0 \tag{1}$$

where $\mathbf{U}$ is the vector of conservative variables, $\mathbf{F}$ is the sum of the inviscid and viscous fluxes, Ω is the flow domain, Σ is its boundary, and $\mathrm{d}\mathbf{A}$ is the differential area pointing outward to the boundary.

The baseline solver [14], known as Mu^2s^2T, uses hybrid unstructured grids to discretize the spatial domain and may contain cells with an arbitrary number of faces. The solution vector is stored at the vertices of the cells. The control volume associated to a node is formed by connecting the median dual of the cells surrounding it using an edge-based data structure (see Fig. 1). For the internal node i, the semidiscrete form of the system of nonlinear Eq. (1) can be written using a finite volume approach as

$$\frac{\mathrm{d}(\Omega_i\mathbf{U}_i)}{\mathrm{d}t}+\sum_{j=1}^{n_{\text{edges}}}\mathbf{F}_{ij}\cdot\mathbf{A}_{ij}-\mathbf{D}_{ij}=\mathbf{S}(\mathbf{U}_i) \tag{2}$$

where Ω_i is the control volume, $\mathbf{A}_{ij}$ is the area associated to the edge ij, $\mathbf{F}_{ij}=\frac{1}{2}(\mathbf{F}_i+\mathbf{F}_j)$ represents the inviscid and viscous fluxes through area A_{ij}, $\mathbf{D}_{ij}$ are the artificial dissipation terms, and n_{edges} is the number of edges that surround node i. The resulting spatially discretized equations can be recast as a summation at each vertex of contributions along all edges meeting at that vertex. The resulting numerical scheme is cell centered in the dual mesh (dashed line in Fig. 1) and second-order accurate. It may be shown that, for triangular grids, the scheme is equivalent to a cell vertex finite volume scheme.

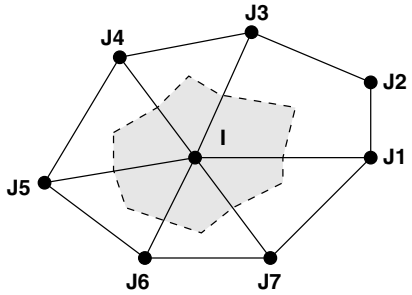

Fig. 1 Typical hybrid-cell grid and associated dual mesh.

Numerical Diffusion

A blend of second- and fourth-order artificial dissipation terms, $\mathbf{D}_{ij}$, is added to capture shock waves and prevent the appearance of high-frequency modes in smooth flow regions, respectively. The second-order terms are activated in the vicinity of shock waves by means of a pressure-based sensor and, locally, the scheme reverts to first order in these regions. The artificial dissipation terms can be written as

$$\mathbf{D}_{ij} = |S_{ij}|A_{ij}[\mu_{ij}^{(2)}(\mathbf{U}_j - \mathbf{U}_i) - \mu_{ij}^{(4)}(L_j - L_i)] \tag{3}$$

where $\mu_{ij}^{(2)} = 0.5(\mu_i^{(2)} + \mu_j^{(2)})$ and $\mu_{ij}^{(4)} = 0.5(\mu_i^{(4)} + \mu_j^{(4)})$ are the averages of the artificial viscosity coefficients in the nodes i and j, which are given by

$$\mu_i^{(2)} = \min(\varepsilon_2, k_2\delta_i), \qquad \mu_i^{(4)} = \max(0, \quad \varepsilon_4 - k_4\delta_i) \tag{4}$$

where δ_i is a pressure-based sensor,

$$\delta_i = \frac{\left|\sum_{j=1}^{n_{\text{edges}}}(p_j - p_i)\right|}{\sum_{j=1}^{n_{\text{edges}}}(p_j + p_i)} \tag{5}$$

and ε_2, k_2, ε_4, and k_4 are constants. Typically, $\varepsilon_2 = 0.5$ and $\varepsilon_4 = \frac{1}{128}$. L is a pseudo-Laplacian operator:

$$L(\mathbf{U}_i) = \sum_{j=1}^{n_{\text{edges}}}(\mathbf{U}_j - \mathbf{U}_i) \tag{6}$$

Finally, $|S_{ij}|$, a 5×5 matrix that plays the role of a scaling factor, is the Roe matrix [15].

Viscous Terms

The viscous fluxes can be evaluated in a number of ways. The gradients of the flow variables are approximated at the nodes using the divergence theorem computed in the same way as the convective fluxes:

$$(\Omega\nabla U)_i = \sum_{j=1}^{n_{\text{edges}}}\frac{1}{2}A_{ij}(U_i + U_j) \tag{7}$$

Then, an approximation of the gradients at the midpoint of the edge is obtained by a simple average:

$$\overline{\nabla U}_{ij} = \frac{1}{2}(\nabla U_i + \nabla U_j) \tag{8}$$

Potentially, the previous expression may be used to compute the flux of the viscous terms; however, to reduce the stencil of the resulting scheme and to mimic the discretization that is obtained in structured grids [16], Eq. (8) is replaced by

$$\nabla U_{ij} = \overline{\nabla U}_{ij} - \left(\overline{\nabla U}_{ij}.\delta\mathbf{s}_{ij} - \frac{(U_i - U_j)}{|\mathbf{x}_i - \mathbf{x}_j|}\right)\delta\mathbf{s}_{ij} \tag{9}$$

where $\delta\mathbf{s}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)/|\mathbf{x}_i - \mathbf{x}_j|$.

Turbulence effects are accounted for using the Wilcox 1998 k-ω model with realizability [17], which is integrated to the wall.

Implicit Temporal Discretization

Equation (2) can be expressed in the form

$$\frac{\mathrm{d}(\Omega_i\mathbf{U}_i)}{\mathrm{d}t} = \mathbf{R}(\mathbf{U}) \tag{10}$$

It is a common practice [18] to use a second-order-implicit backward time integration of Eq. (10) that can be expressed as

$$\frac{3(\Omega_i\mathbf{U}_i)^{n+1} - 4(\Omega_i\mathbf{U}_i)^n + (\Omega_i\mathbf{U}_i)^{n-1}}{2\Delta t} = \mathbf{R}(\mathbf{U}^{n+1}) \tag{11}$$

where n denotes the time-step level. The nonlinear system of equations represented by Eq. (11) needs to be solved at every time step. An iterative method is constructed by adding a pseudotime derivative $\mathbf{U}_\tau$ in the left-hand side:

$$P\frac{\partial\mathbf{U}_i}{\partial\tau} + \frac{3(\Omega_i\mathbf{U}_i) - 4(\Omega_i\mathbf{U}_i)^n + (\Omega_i\mathbf{U}_i)^{n-1}}{2\Delta t} = \mathbf{R}(\mathbf{U}) \tag{12}$$

Between each time step, the solution is advanced in the nonphysical time τ using an explicit five-stage Runge–Kutta scheme where the artificial viscosity terms are evaluated only in three stages of the Runge–Kutta [19]. Acceleration strategies, represented by the preconditioning matrix P, such as local time stepping, Jacobi preconditioning, and multigridding are used to speed up the convergence of Eq. (11). In this paper, we have modified Eq. (11) by discretizing $(\Omega\mathbf{U})_i$ using a four-point stencil second-order optimized scheme [20] that further reduces the dissipation error of the standard second-order backward discretization.

Slide Plane Treatment

The information transfer between two nonconformal grids may locally alter the solution and degrade the numerical accuracy in the vicinity of the interface. The origin of the problem is the difference between the inner and boundary numerical schemes; for this reason, nonconformal interfaces are usually avoided in standard computations. However, in rotor/stator configurations, the relative motion between the rotor and stator is accounted for using body-fitted grids

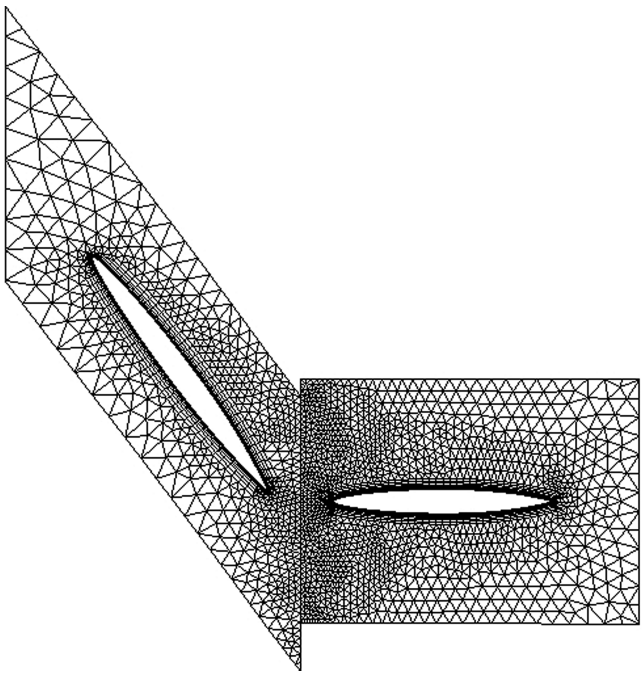

Fig. 2 Coarse hybrid grid of the NLR fan midsection used for the verification of the phase-lagged boundary conditions.

to the airfoils, and the transfer of information between nonconformal grids is mandatory (see Fig. 2). The approach adopted to solve this interaction must take into account the numerical algorithms themselves and the data structures that support them. In particular, the use of unstructured grids, multigridding, or parallelization techniques may impose severe restrictions on the proposed solution. Sayma et al. [9] made use of the semistructured nature of their grid system to construct a pseudo-3-D conservative exchange of information for their point-implicit numerical method by considering data from each radial layer separately. Hills [21] performed a more generic approach by first extending the original grid of the stator and rotor to ensure that both grids overlapped in a single row of elements and then interpolating the variables in the cells of the neighboring row. His numerical method [22] is implicit, but the convergence to each physical time step is accelerated using an edge-collapsing multigrid algorithm that allows the use of an interpolation approach even in the coarser levels of the multigrid. In our case, the coarsening strategy is based on agglomeration, and the grid topology of the coarser grids does not exist, preventing the use of interpolation in the volumetric cells to transfer information in the slide plane.

Several approaches exist to transfer information between nonconformal grids. Depending on the spatial discretization scheme, two options are at hand. The most common method is to close the control volumes with the interface plane [23] because its implementation is simple, at least in structured grids (Fig. 3, left). This is the best approach when the spatial discretization is cell centered. When dealing with unstructured codes that use an edge-based data structure, the unknowns associated with the control volume lie on the interface nodes. The use of half volumes that only retain the contribution of one of the sides of the interface plane locally degrades the accuracy of the spatial discretization. In this work, it has been decided to implement a numerical treatment at the interface that takes into account flux contributions from cells at both sides of the interface (Fig. 3, right) to retain a second-order-accurate scheme at the interface.

Both approaches require the calculation of the interface areas: the former to transfer the fluxes between the cells (Fig. 3, left) and the latter (Fig. 3, right) to choose representative factors to merge the control volumes located at both sides of the interface.

The method adopted in the present work begins by computing the fluxes across all the faces of the control volumes abutting the sliding plane. Then, these fluxes are added with the contributions coming from the other side of the interface plane to form a fictitious control volume that extends at both sides of the interface plane (Fig. 3, right). The current approach uses an overlapping method in the interface in the sense that the control volumes associated with the nodes in the left-hand side of the interface (Fig. 4, right) overlap, both in the normal and tangential directions of the interface, with the control volumes associated with the nodes of the right-hand side (Fig. 4, middle). The information associated with the opposite side of the interface is conceptually completed, creating ghost control volumes (dashed lines in Fig. 4); however, these control volumes are not actually created in practice. The fluxes associated with the semivolume of the side of the interface to which the node belongs $\mathbf{R}_k^+$

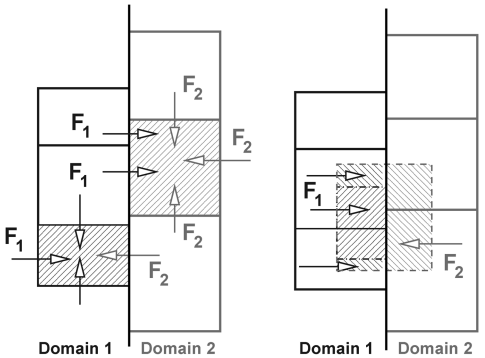

Fig. 3 One-sided (left) and two-sided (right) control volumes in a nonconformal plane (solid line: mesh; dashed line: control volume).

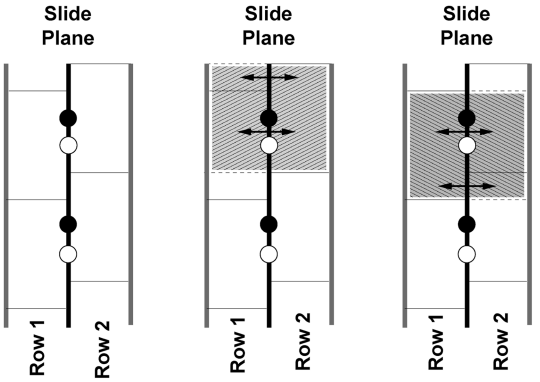

Fig. 4 Scheme of the overlapping control volume system at the sliding plane (black and white nodes are associated with row 1 and row 2, respectively): control volumes at the slide plane boundaries (top), control volume associated with the second row (middle), and control volume associated with the first row (bottom).

are computed using the baseline edge-based numerical scheme. However, the fluxes associated with the opposite side $\mathbf{R}_k^-$ are calculated by means of a weighted area average of the contribution of the cells at the other side of the interface that share a part of the projection of the face of the control volume of node k abutting on the interface a_{kj}, which in practice is a polygon (Fig. 5 outlines the scheme and the nomenclature). In other words,

$$\mathbf{R}_k^{\text{Tot}} = \mathbf{R}_k^+ + \mathbf{R}_k^- = \mathbf{R}_k^+ + \sum_{j=1}^{\text{Nneig·cells}} \frac{a_{kj}}{A_{kj}} \mathbf{R}_j^- \tag{13}$$

where $\mathbf{R}_j^-$ is the total flux associate to the cell j; A_{kj} is the area of the face of the cell j located at the interface, which is neighbor of volume k; and a_{kj} is the intersection of A_{kj} with the face of the cell on the other side of the interface that we are considering (see Fig. 5). The volume associated to cell k is computed in an analogous way as $\Omega_k = \Omega_k^+ + \Omega_k^-$.

The same scheme could be used as well for the computation of the gradients in the interface nodes using the divergence theorem; however, although the extra cost would be negligible, this approach has not been followed in the present work, and all the fluxes are treated in the same way. This means that the computation of the viscous fluxes at the interface could be further improved.

The aforementioned algorithm is implemented, expanding the list of edges of the baseline algorithm in such a way that, for every node in the sliding plane k, there are as many edges added as dual cells of the opposite side of the interface abutting in the cell associated to node k, $N_{\text{neigh·cells}}$ (see Fig. 5). This is a dynamic list that is updated at every time step using an alternate digital tree (ADT). Each edge of this list has an associated factor, a_{kj}/A_{kj}, that is also updated every time step. This expanded list of edges has the form: ednod_new [k, index_node (j)], where k is the global index of the node in the interface and index_node (j) is the global index of the nodes at the other side of the sliding plane for which the associated cells are abutting on the cell associated to node k and where

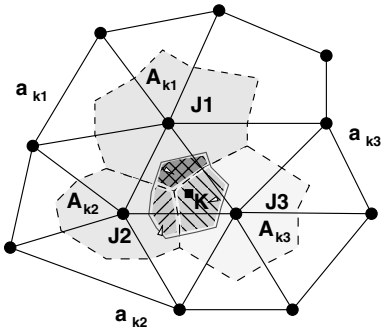

Fig. 5 Flux transfer across the interface plane in an edge-based solver.

$j = 1, \ldots,$ Nneigh·cells. $\mathbf{R}_j^-$ is computed using the standard algorithm. In practice, a new loop over the new edges performs a slightly modified version of the baseline algorithm.

It is important to recall that the only connectivity used by the algorithm is the edge connectivity. The cell connectivity of the volumetric grid is just used in a preprocessing step to compute the dual areas and volumes. The face connectivity of the sliding plane is, however, used to dynamically compute a_{kj}. All these areas could be computed in a preprocessing step and stored and, in this case, the only information needed by the algorithm would be the edge connectivity, but its computation in every time step is preferred to reduce memory usage.

The multigrid is an integral part of the numerical method, and the slide plane treatment needs to take this fact into account. The current approach is transparent in the coarser grids. Once the fluxes, the projected areas, and the extended list of edges are computed in the finest grid, the agglomeration process is carried out in a consistent way in the coarser grids. This means that the sliding plane treatment described in the following section is only conducted in the finest grid.

Although the distribution of the conservative variables along the two sides of the interface is slightly discontinuous, by construction, the fluxes at both sides of the interface are the same; hence, the scheme is conservative. It is worth noting that, if the surface meshes at both sides of the interface are identical, the flux transfer provided by the present approach and that obtained by merging the grids to obtain a conformal mesh are exactly the same. This approach is believed to be very similar to that outlined in [24].

Simplified Analysis of the Discretization

With the aim of clarifying the nature of the proposed scheme, we include the discretization that would produce the current approach in a two-dimensional (2-D) structured mesh with uniform grid spacings, Δx^- and Δx^+, at both sides of the interface (see Fig. 6). Let us consider Eqs. (2) and (13); in this case,

$$\frac{\mathrm{d}(\Omega_k \mathbf{U}_k)}{\mathrm{d}t} + \mathbf{R}_k^+ + \mathbf{R}_k^- = 0 \tag{14}$$

with

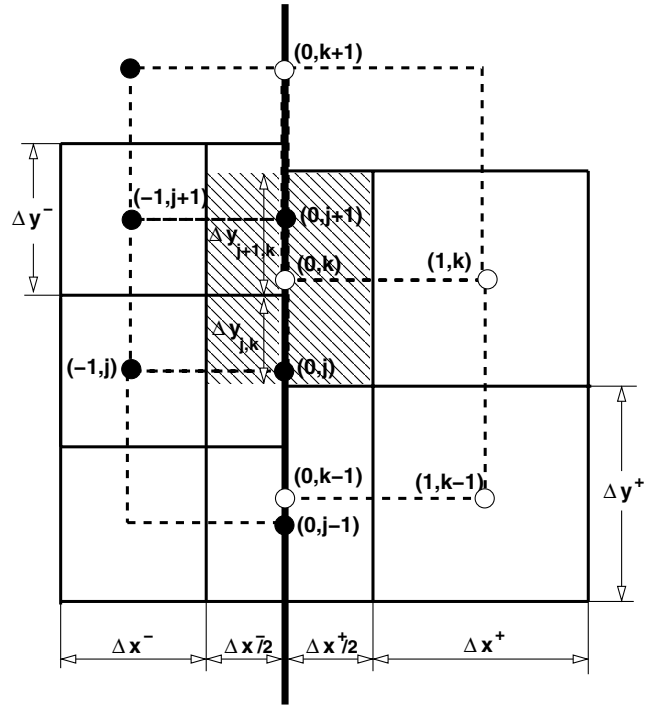

Fig. 6 Scheme of the discretization at the sliding plane in a 2-D structured grid (dashed line: actual grid; solid line: dual mesh).

$$\mathbf{R}_k^- = \frac{\Delta y_{j,k}}{\Delta y^-}\mathbf{R}_j^- + \frac{\Delta y_{j+1,k}}{\Delta y^-}\mathbf{R}_{j+1}^- \tag{15}$$

The flux contribution from the left-hand side of the interface to cell k is a weighted sum of the fluxes at cells $(0, j)$ and $(0, j+1)$, where

$$\mathbf{R}_j^- = (F_{\mathrm{sp}} - F_{-1/2,j})\Delta y^-\mathbf{i} + (G_{0,j+1/2} - G_{0,j-1/2})\Delta x^-/2\mathbf{j} \tag{16}$$

$\mathbf{F} = F\mathbf{i} + G\mathbf{j}$ and F_{sp} are the flux across the sliding plane. Since $G_{0,j+1/2} = 0.5(G_{0,j+1} + G_{0,j})$, then

$$\mathbf{R}_j^- = (F_{\mathrm{sp}} - 0.5[F_{0,j} + F_{-1,j})]\Delta y^-\mathbf{i} + 0.5(G_{j+1} - G_{j-1})\Delta x^-/2\mathbf{j} \tag{17}$$

The ghost cell volume Ω_k may be expressed as

$$\Omega_k = \Omega_k^- + \Omega_k^+ = \Delta y^+\Delta x^-/2 + \Delta y^+\Delta x^+/2$$

It is important to recall that $\Delta y^+ = \Delta y_{j,k} + \Delta y_{j+1,k}$. Assuming that $\Delta x^+ = \Delta x^-$, for the sake of simplicity, we obtain that

$$\begin{aligned}\frac{\mathrm{d}\mathbf{U}_k}{\mathrm{d}t} = \frac{1}{2}\Bigg[&\frac{G_{0,k+1}^+ - G_{0,k-1}^+}{2\Delta y^+} + \alpha\frac{G_{0,j+1}^- - G_{0,j-1}^-}{2\Delta y^-} \\ &+ (1-\alpha)\frac{G_{0,j+2}^- - G_{0,j}^-}{2\Delta y^-}\Bigg]\mathbf{j} + \Bigg[\frac{F_{1,k}^+ + F_{0,k}^+}{2\Delta x} - \alpha\frac{F_{0,j}^- + F_{-1,j}^-}{2\Delta x} \\ &- (1-\alpha)\frac{F_{0,j+1}^- + F_{-1,j+1}^-}{2\Delta x}\Bigg]\mathbf{i}\end{aligned} \tag{18}$$

where $\alpha = \Delta y_{j,k}/\Delta y^+$, and the superscripts $-$ and $+$ have been added to highlight the contributions from the left- and right-hand sides of the sliding plane, respectively. The first observation is that the flux across the sliding plane does not appear in the formulation. It may be noticed as well that $\partial G/\partial y$ is a weighted average of second-order derivatives at the left- and right-hand sides of the interface, the former being a combination of the derivatives at nodes j and $j+1$. On the other hand, the second row of Eq. (18) may be rearranged to express $\partial F/\partial x$ as

$$\begin{aligned}\frac{\partial F}{\partial x} = &\frac{F_{1,k}^+ - [\alpha F_{-1,j}^- + (1-\alpha)F_{-1,j+1}^-]}{2\Delta x} \\ &+ \frac{F_{0,k}^+ - [\alpha F_{0,j}^- + (1-\alpha)F_{0,j+1}^-]}{2\Delta x}\end{aligned} \tag{19}$$

The first term of the previous expression is a standard second-order derivative, where the contribution from the left-hand side is a weighted average of the fluxes at the rows j and $j+1$. The second, ϵ_x, is an errorlike term that, if F varies linearly between j and $j+1$, reduces to

$$\epsilon_x = \left(1 - \frac{\Delta y^+}{\Delta y^-}\right)\left(\alpha - \frac{1}{2}\right)\frac{(F_{0,j+1}^- - F_{0,j}^-)}{2\Delta x} \tag{20}$$

This error term vanishes if either the variation of the flux along the sliding plane is null or the grid spacing at both sides of the sliding plane is the same (i.e., $\Delta y^+ = \Delta y^-$), independently of the relative position of the sliding plane α. A second observation is that the flux across the sliding plane, F_{sp}, is not actually needed. This is a direct consequence of the fact that the sliding plane is an internal boundary of the control volume.

The spatial discretization is, therefore, second-order accurate in a uniform grid, since it is obtained as either a linear combination of second-order derivatives at both sides of the interface plane or as second-order derivatives for which the elements are obtained by a linear combination of variables at the same side of the interface plane.

Geometric Area Computation

The present method requires the projection of all the faces of the dual mesh abutting on one side of the sliding plane on the opposite side. In our approach, all the polygons of the dual mesh at the sliding plane are subdivided first in triangles in a preprocessing step (see

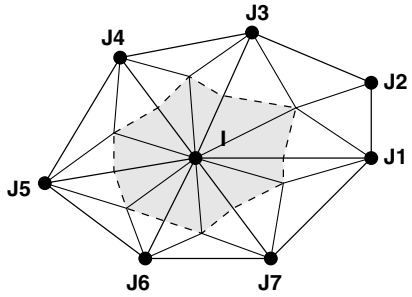

Fig. 7 Subdivision of the faces of the dual mesh in triangles to perform intersection checking.

Fig. 7). Then, every vertex of this set of triangles is projected along, normal to the surface, on the opposite side of the sliding plane, to form the footprint of the projected triangle. With this information, the intersection of the edges of all these triangles against the triangles of the dual-surface mesh of the opposite side is computed to determine the areas that need to be associated with each control volume (see Fig. 8).

These geometric operations are performed in a preprocessing step for a static nonconformal plane. However, in a rotor/stator simulation, these area-weighting coefficients need to be updated at every time step, since the relative position between the grids at both sides of the interface is changing with time. In this case, special attention must be paid to the efficiency of the method to avoid its computational cost becoming a significant part of the code. If the number of points of each domain is N, the number of nodes of the interface scales as $N^{2/3}$; therefore, a brute force algorithm to compute all the projections would require $N^{4/3}$ operations, which is larger than the number of operations per iteration.

Since the transfer of properties at the slide plane is based on the projection of the areas, the method does not require overlapped or coincident grids at the sliding plane. The method is more topological, and the meshes could even be physically separated, provided that they are facing each other, which makes the method very robust.

The algorithm reads as follows:

1) Create a list of points containing the vertices, edge midpoints, and face centers of all the triangles of the departing grid.

2) Create a list of triangles of the dual mesh containing the node of the original grid to which the triangle is associated, the three nodes of the triangle in a prescribed order (increasing index), and the three neighbors by the face of the triangle in a given order (the first triangle is the neighbor of the edge formed by nodes 1 and 2, etc.). If one edge has no neighbor, it is flagged with a zero.

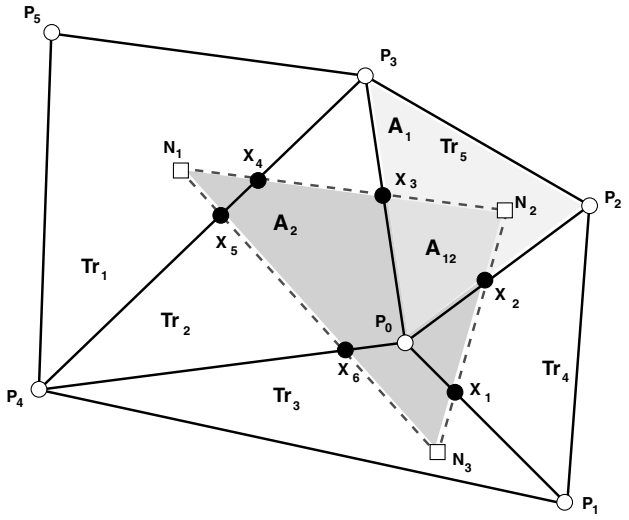

Fig. 8 Intersection between triangles of the dual areas of two control volumes abutting on the opposite sides of the sliding plane.

3) Project the nodes of one side of the slide plane on the triangles of the other side (the so-called base triangles) and obtain the triangles that contain the projection of every node. We use an ADT to discard most of the checks, which is an essential part of the algorithm. The ADT needs to be constructed only for the triangles conforming the stationary part of the sliding plane. The projection is performed in the normal direction associated to every node of the donor side, which is actually a weighted area average of the normals of the cell faces that surround that node. Both the projection and the search are performed in Cartesian coordinates, and they do not make use of the fact that the sliding plane is a surface of revolution. The main advantage of this approach is that the method may be reused to transfer fluxes between arbitrary pairs of nonconforming grids.

4) Project the edges of every triangle of the dual mesh. The projection plane is defined by the projections of the two nodes of the edge and the average of the normals at both nodes. Since the edge is the basic entity for the projections and only information associated to the edges is used, gaps or slits between adjacent triangles are not created. The procedure also ensures that the node projections are unique and lie on the projected edges. Because of the rotor motion, the triangles of the rotating part of the sliding plane need to be rotated a discrete number of rotor pitches to face the stationary part of the sliding plane. This process has to be repeated several times for each time step: the actual number depending on the relative size of stator and rotor pitches.

5) Determine the intersection of the projected edges with the edges of the triangles that conform the dual mesh of the opposite side of the sliding plane (black symbols in Fig. 8) using the parametrization of the edge. Only the valid intersections (i.e., those for which the intersecting point is between the nodes that define the edge) have to be considered.

6) Use the list of neighbors to find additional intersections in other triangles of the base grid until reaching the end of the edge. This process is prone to roundoff errors when similar types of grids and high resolutions are used at both sides of the sliding plane. Partial or total coincidence of elements, as well as other pathologies (see Fig. 9, bottom), are likely to occur, and roundoff errors may cause misleading paths and prevent the closure of the projected triangle, even if the geometry is locally renormalized using the element local size. This may happen when the projected edge is close to one of the vertices of the base grid or parallel to the other edge. A merit function

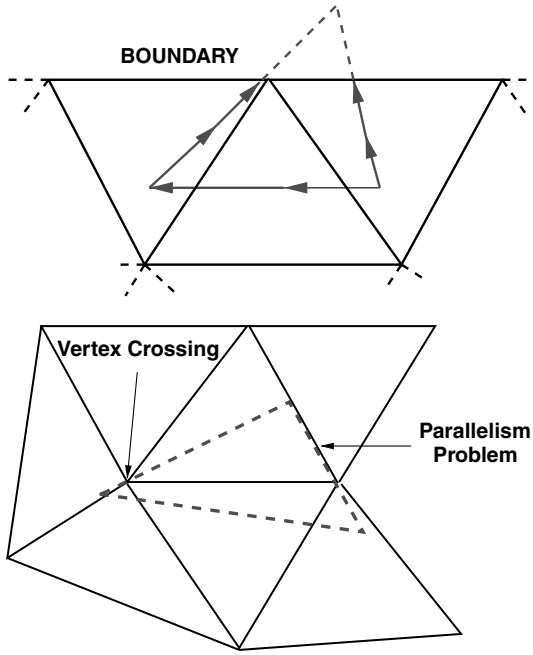

Fig. 9 Pathological cases looping the path: the scheme of a path reaching boundary and sense inversion to close the path (top) and the potential causes of roundoff problems (bottom).

is given to the three cuts of the segment with the triangle that depends on the distance of the cut along the segment that actually conforms the triangle, the distance between the two lines (if they are parallel), and the distance between the segment boundaries and the cut performed in its line. The best cut is selected using this merit function, and the algorithm is forced to march forward by preventing the revisiting of a previous cell.

7) Repeat the procedure for every projected edge of the triangle.

8) Define the segmented polygonal shape of the projected triangle on the elements of the base grid in an anticlockwise sense.

9) Compute the common area of the triangles (e.g., the area A_{12} in Fig. 8 is the polygon defined by P_0, X_2, N_2, and X_3), making use of the sense of the polygon to distinguish what is in and out of the polygon. A special logic is needed to define these areas in the subtriangles that define the intersection.

10) Seek small triangles that are fully embedded in the projected triangle and have no cuts by flagging all the intersected triangles and looking for triangles that are surrounded by them, making use of the sense of the loop.

11) Fix the logic for paths for which the projection is partially outside the boundaries of the base grid. Upon reaching the boundary, the algorithm reverts the sense of looping to close the loop using the same logic (see Fig. 9, top).

12) Clean the path to eliminate round-error contamination. When a side of the projected triangle lies very close to a vertex of the receiving side, the process described in step 6 may lead to a series of spurious cuts around the vertices, usually under the form of microloops that need to be eliminated to avoid a misinterpretation of which are the inner and the outer parts of the triangle. Figure 10 describes how an error in the determination of the intersections together with the algorithm that prevents the search of intersections between the node N_1 and N_2 to backtrack may create a microloop of the size of the roundoff error that can make the whole logic fail.

The result of the aforementioned algorithm is an area that links two control volumes, one at each side of the sliding plane, which is straightforward to implement following an edge-based structure. Since our code uses an edge-based data structure, the integration of this logic is simple, and it is based on a expanded list of edges that connects every node of one of the sides of the sliding plane with a node at the opposite side.

When pseudo-Laplacian operators are used to include artificial viscosity terms, these edges can be used by the nodes at the slide plane to take into account the influence coming from the other row. The rest of the flux contributions do not need additional corrections. The multigrid algorithm is handled in a transparent way, since the described method is entirely edge based.

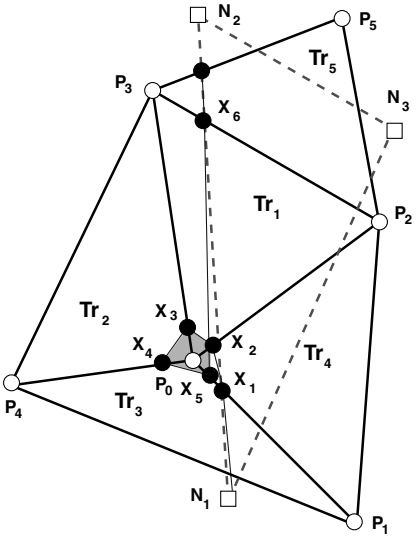

Fig. 10 Cleaning process of a loop upon contamination due to roundoff errors. Shadowed polygon is eliminated.

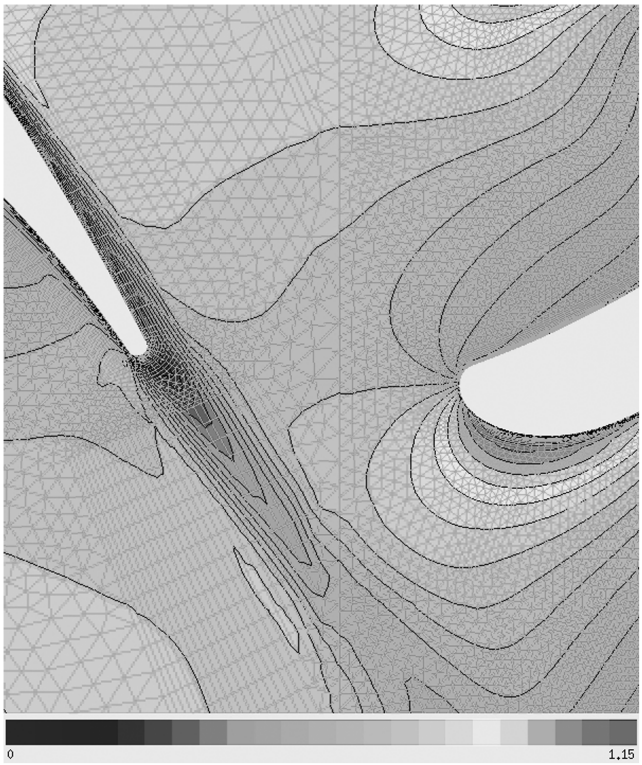

Fig. 11 Mach number isocontours in the sliding plane region of an HP turbine.

Examples

Figure 11 shows a close up of the sliding plane region of the HP turbine. The Mach number isocontours exhibit a great degree of regularity in the interface, although the grid is relatively coarse. Still, a careful traverse of the solution at the interface shows small discontinuities at the interface due to the mismatching between the grids at both sides of the interface plane. The magnitude of the discontinuity is smaller than the variation of the variables within one cell.

Phase-Lagged Boundary Conditions

Background

To retain the actual blade count in a rotor/stator simulation, several approaches are possible. The standard procedure is to select a flow unit that exactly represents the original blade count. This approach necessarily leads to the use of several airfoils per row, and it is expensive in terms of CPU and memory. Alternatively, an approximated blade count may be selected to accommodate the actual blade count in a smaller number of airfoils in the simulations. The main drawback of this approach is that essential characteristics of the unsteady flowfield may be missed in the approximation.

Several methods exist to take into account this lack of spatial periodicity without recurring to the computation of very large domains in the circumferential direction. The pioneering ideas of the concept of phase-lagged boundary conditions are due to Erdos et al. [10], who employed a direct store technique in the periodic boundaries to account for the fact that the time periodic signals in the upper periodic boundary U_U lead or lag a certain amount of time, that of the lower periodic boundary U_L. The method was extended to the computation of 3-D turbine blade–row interaction [25] and further pursued by He [11], who developed the so-called shape correction (SC) method, where the Fourier coefficients of the raw signal, instead of the signal itself, are stored to minimize code memory requirements. Lately, the SC approach was implemented for blade–row interactions [12,13]. The basic hypothesis of both methods is that the signal is periodic in time with a well-known single fundamental frequency. The convergence history of both methods is very similar,

although the Erdos method always tends to converge slightly better [26]. In this work, the direct store method has been used.

The problems associated with the implementation of phase-lagged boundary conditions in an unstructured code are discussed in the following lines.

Axial Rotor/Stator Interaction

When simulating blade–row interactions using single-passage techniques, not only the circumferentially periodic boundary conditions but the axial communication between the rotor and stator blocks need to be modified. In full annulus simulations, the boundary condition consists of exchanging information between the blocks in the most transparent way. Single-passage methods need to reconstruct part of the interface information, because the stator, P_s, and rotor, P_r, pitches do not match. This process takes place in a buffer region, sketched in Fig. 12 (top). Let us denote m and q as the stator and rotor passages we are interested in, being the zeroth passages that are actually computed. The first step is to relate the variables in the passages m and q with their corresponding values in the reference channel in a past time. In general, at the interface, the following relations hold:

$$U_q(x,R,\theta,t) = U_{q=0}\left(x,R,\theta,t-\frac{qP_r}{\omega}\right)$$
$$U_m(x,R,\theta,t) = U_{m=0}\left(x,R,\theta,t+\frac{mP_s}{\omega}\right) \tag{21}$$

where θ is relative to the individual-blade passage. Figure 12 (bottom) represents the time signal of a generic point of the zeroth rotor passage in a frame of reference that moves with the rotor. A point for which the relative position to the second-blade passage is (x, R, θ_0) lags the signal of the corresponding point in the zeroth rotor passage by $2P_r/\omega$; this means that the current value of the point of the second passage corresponds to the point marked as $\theta_0^{(q=2)}$ in Fig. 12 (bottom). The second step is to reconstruct the required values in the reference passage using either the stored variables (direct store method) or an assumed shape for the time evolution of that point (a precomputed Fourier series if the generalized SC method is used). Both make the hypothesis that the stator- and rotor-blade signals have a period equal to the blade passing frequency of the adjacent blades. In the Erdos method, the stored data represent a complete period of the unsteady flowfield. This approach is sketched in Fig. 12 (bottom), where the information of a certain point of the second rotor passage is first associated with a past time of the reference passage [Eq. (21) point $\theta_0^{(q=2)}$ in Fig. 12] and then related with the stored data that only cover one rotor-blade period (time shift from point $\theta_0^{(q=2)}$ to point A). Formally, this may be expressed as

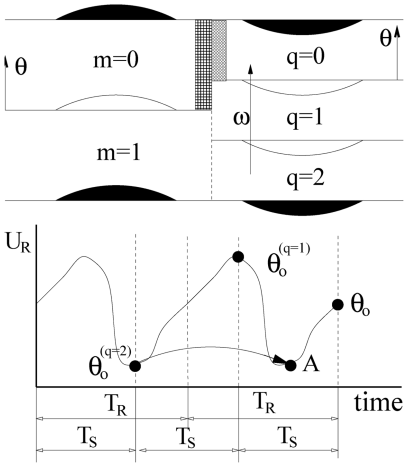

Fig. 12 Axial block communication in a stator/rotor configuration. The shaded areas in the interface region (top) display the rotor and stator buffers. The time evolution of a rotor-blade point is sketched (bottom).

$$U_{q=0}(x,R,\theta,t-qT_s) = U_{q=0}\left[x,R,\theta,t-qT_s+\mathrm{int}\left(\frac{qT_s}{T_r}\right)T_r\right] \tag{22}$$

where $T_s = P_r/\omega$ and $T_r = P_s/\omega$.

This is a pointwise operation and does not pose any additional difficulty to an unstructured method, apart from those related with the exchange of information in the interface plane and that have been described in the previous section.

Results

National Aerospace Laboratory (The Netherlands) Fan Description

The method has been applied for the computation of a low-speed fan tested at NLR [27]. The experiment was designed to validate the prediction of the unsteady pressure in the airfoils by means of a simplified method: the so-called lifting surface theory (LST). The ultimate goal was to characterize the acoustic signature of the fan; therefore, little attention was paid to its characterization from a steady point of view. The fan has straight inner and outer annuli (see Fig. 13, top). The stator consists of 18 unleaned and unswept vanes aligned with the uniform flow for which the streamwise sections are uncambered NACA 0010-35 profiles with a relative thickness of 10%. The chord length, which is constant along the span, is 50 mm.

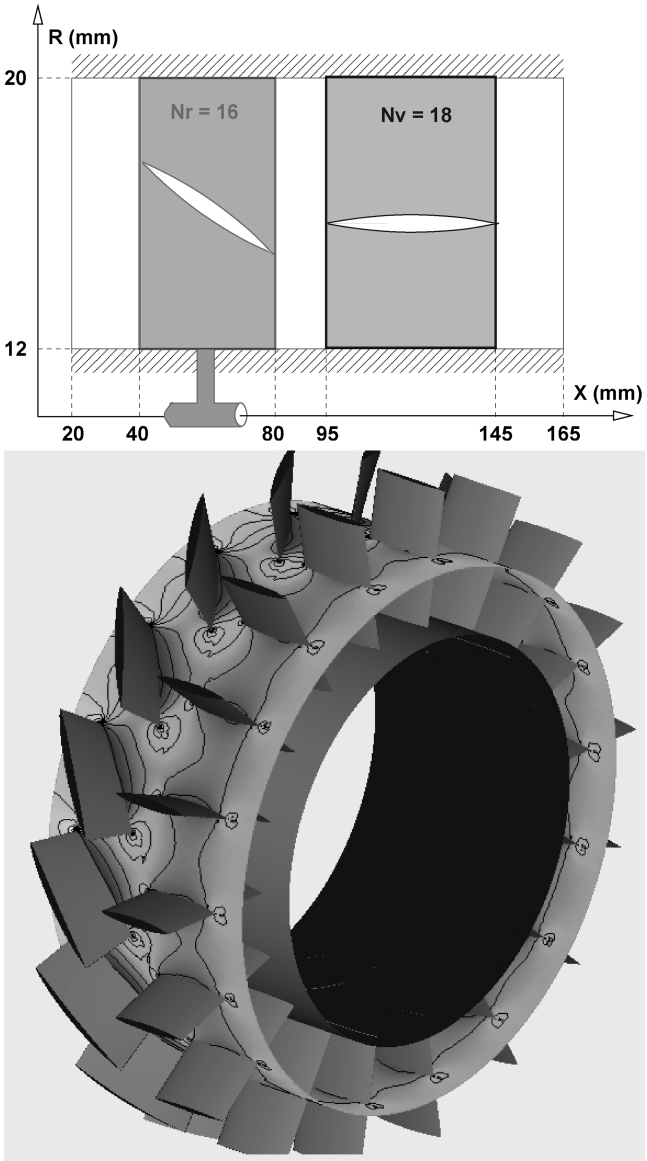

Fig. 13 Layout (top) and 3-D view and static pressure at midspan (bottom) of the low-speed NLR fan.

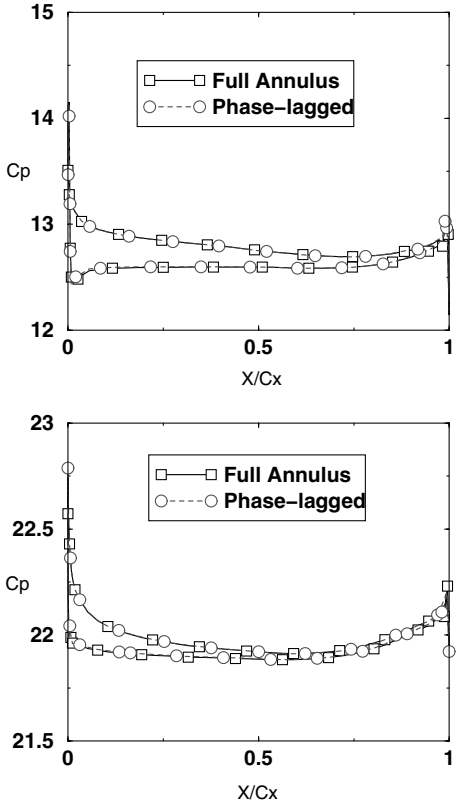

Fig. 14 Computed mean pressure coefficient distribution at the rotor (top) and stator (bottom) midsections of the NLR fan for $\Omega = 5550$ rpm (□: full wheel; ○: phase-lagged boundary conditions).

Along a chordwise section, 20 unsteady pressure transducers were mounted symmetrically at the upper and lower sides. This vane could be moved radially, and measurements were carried out at three spanwise positions: 30% annular span (from the hub), 53% span, and 76% span.

A rotor of 16 unleaned and unswept blades was mounted 15 mm upstream of the stator. The stagger angles were adjusted in such a way that the inlet relative flow angle was zero when the rotor rotated at 6650 rpm and the axial velocity was 85 m/s. The rotor blades had a constant axial chord of 40 mm; hence, the true actual chord varies in the radial direction. The blade section at the hub was a NACA 0010-35 section. At other radii, the relative thickness of this section is reduced, keeping constant the absolute maximum thickness along the span. Also, the midsection of one of the rotor blades was equipped with unsteady pressure transducers. At rotor shaft speeds different from 6650 rpm, the axial Mach number is modified to keep a zero incidence angle in the rotor. The information of the rotor tip clearance in the experiment is not available, and the existence of an abradable material in the outer casing has been assumed; therefore, all the simulations in this work have been set up with zero tip clearance.

Operating Point

All the simulations presented in this work correspond to a fan rotational speed of 5550 rpm, which is the operating point better described in [27]. For this condition, the axial Mach number is 0.187 and the tip Mach number of the rotor 0.343.

Steady Loading

The nondimensional pressure coefficient distribution on the midsection of the rotor and stator at $\Omega = 5550$ rpm may be seen in Fig. 14. It may be appreciated that, although the inlet Mach number is adjusted to have local zero incidence at the midsection of the rotor, the stagger angle of the rotor blade makes the configuration nonsymmetric and induces some loading. This loading on the rotor changes its exit angle and the incidence in the stator, which is not zero anymore. As a consequence, both the rotor and the stator have some degrees, even small, of loading.

The steady and unsteady pressure distributions are nondimensionalized, with the relative dynamic pressure at the inlet of every individual section, $q = \frac{1}{2}(\rho w^2)_{\text{inlet}}$, where w is the relative velocity. It is worth mentioning that the ratio between the dynamic pressure in the rotor and the stator is about three; that is, $q_{\text{rotor}}/q_{\text{stator}} \simeq 3$. The dynamic pressures used to nondimensionalize all the figures have been obtained from the experimental measurements.

National Aerospace Laboratory (The Netherlands) Fan Unsteady Results

Phase-Lagged Boundary Condition Assessment

The accuracy of the phase-lagged boundary conditions has been checked computing a number of different 3-D configurations [28], but here, for the sake of brevity, we will use the results of the NLR fan to both assess the phase-shifted boundary conditions and validate the method.

Since the configuration has 16 rotor blades and 18 stators, we have constructed a computational domain consisting of eight rotor blades and nine stators at a relatively low resolution to reduce the computational cost. The coarse rotor blade and stator grids contain 0.84×10^5 and 1.4×10^5 nodes per passage, respectively. The grid at the midsection of the single-passage phase-lagged simulation is displayed in Fig. 2. This 2-D mesh is extruded in the radial direction to obtain a 3-D semi-unstructured grid [29] with 61 radial planes. The simulation of the whole wheel requires 17 passages, and the grid contains about 2 million nodes. The number of physical time steps per rotor blade passing period is 72. Typically, three multigrid cycles with 20 time steps per level are used.

Figure 15 compares the results obtained for the first, second, and third harmonics of the nondimensional unsteady static pressure

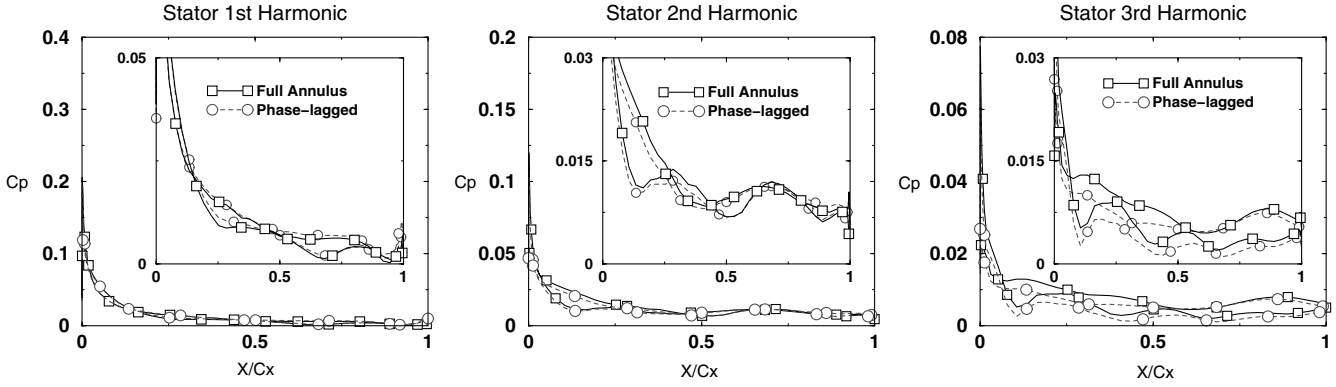

Fig. 15 Comparison of the first (left), second (middle), and third (right) harmonics of the nondimensional unsteady static pressure modulus on the stator midsection, computed using the full annulus (□) and the phase-lagged boundary conditions (○) obtained using the coarse grid.

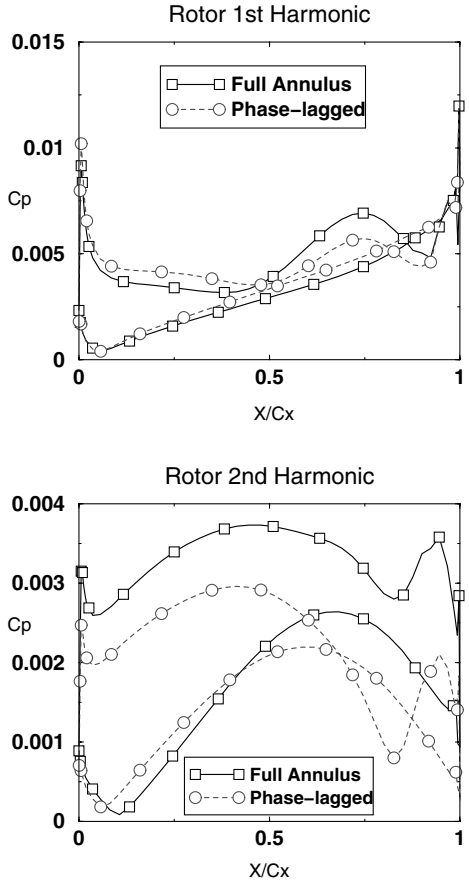

Fig. 16 Comparison of the first (top) and second (bottom) harmonics of the nondimensional unsteady static pressure modulus on the rotor midsection, computed using the full annulus (□) and the phase-lagged boundary conditions (○).

modulus C_p on the stator midsection, using phase-lagged boundary conditions (dashed line marked with circles) and a full wheel simulation (solid line marked with squares). It may be appreciated that only in the inset of the subfigures that enlarge the flatter region of the curve, some differences may be appreciated. It should be emphasized that, in this case, the largest differences between both cases are of the order of 5 Pa and are associated with small discrepancies in the convergence process. These differences are better seen in the third harmonic, which is an order of magnitude smaller than the first.

Figure 16 compares the results obtained for the first and second harmonics of the nondimensional unsteady static pressure modulus on the rotor midsection using phase-shifted boundary conditions and a full wheel simulation. The amplitude of the pressure perturbations for the rotor are about 10 times smaller than for the stator, and some discrepancies may be appreciated. The first harmonic is correctly reproduced but, for the second, the distributions are only approximately correct.

Finally, Fig. 14 contains the mean values of the nondimensional static pressure, both for the stator and the rotor, that obviously present a perfect matching between the full annulus and the single-passage simulations.

Figure 17 shows the convergence history of the unsteady norm of the static pressure on the stator surface as a function of the number of single-blade passing periods. The dashed lines indicate the physical amplitude of the unsteady error referred to the amplitude of the first harmonic of the static pressure. The unsteady convergence is based on the L^2 norm:

$$\epsilon = \frac{1}{TS}\sqrt{\int_{\partial S} \mathrm{d}S \int_{t_0}^{t_0+T} \left\{\frac{p(t)-p(t-T)}{p(t-T)}\right\}^2 \mathrm{d}t} \qquad (23)$$

where T is either the single-blade or the half-annulus passing period, S is the blade surface, and t_0 is a reference time. If the relative error ϵ is 10^{-3}, this means that the error in the unsteady pressure between two consecutive periods is 0.1% of the mean static pressure that, translated in terms of the amplitude of the first harmonic in the stator, is a 100% error. An error in the unsteady pressure of at least 10% of the first harmonic should be required to obtain meaningful results

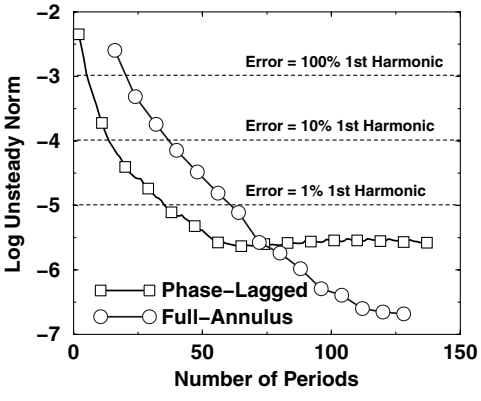

Fig. 17 Unsteady norm convergence history.

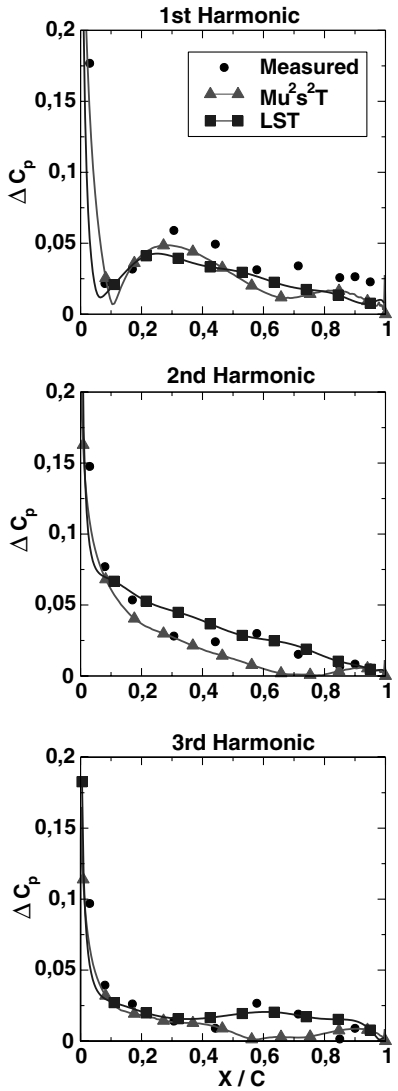

Fig. 18 First three harmonics of the stator unsteady pressure modulus of the pressure jump across the airfoil for $\Omega = 5550$ rpm at a 53% span obtained using the fine grid (○: measurements; △: present method; □: LST theory).

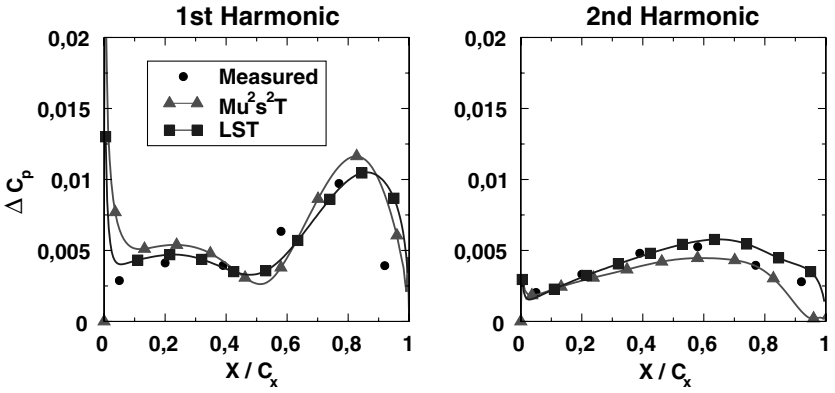

Fig. 19 First and second harmonics of the rotor unsteady modulus of pressure jump for $\Omega = 5550$ rpm at a 50% span.

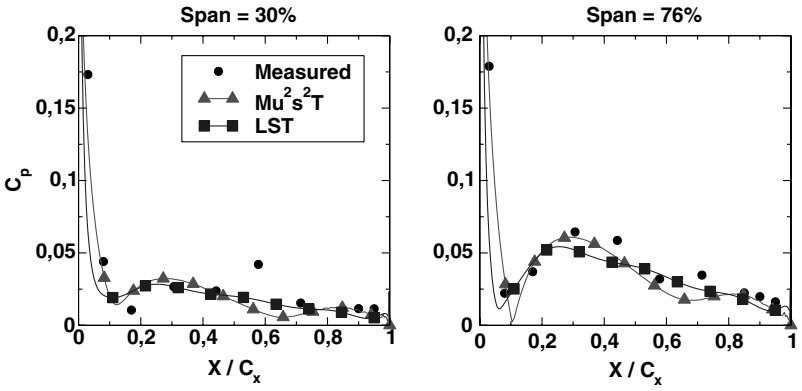

Fig. 20 First harmonic of the stator unsteady modulus of pressure jump at 30% (left) and 76% (right) span.

from the unsteady point of view. It may be appreciated that the convergence rate of the full annulus simulation is similar to that of the single-passage technique, or slightly better. The convergence of the single-passage technique stalls at a higher value than the full annulus simulation, probably due to minor inconsistency errors in the periodicity treatment, which are difficult to avoid in the phase-lagged boundary conditions.

Stator Midsection Validation

Figure 18 compares the first three harmonics of the stator unsteady pressure modulus jump across the airfoil, at a 53% span and 5550 rpm, obtained in the NLR wind tunnel with the LST [27] and the present method, using phase-lagged boundary conditions. The unsteady static pressure differences across the lower and upper parts of the airfoil are the only data available from [27] and the relevant parameter from a noise point of view.

The rotor-blade and stator grids contain 0.84×10^6 and 1.3×10^6 nodes per passage, respectively, which are much finer than those used for the verification of the phase-lagged boundary conditions. The semi-unstructured grid is constructed using 95 radial planes, and the distance to the wall of the first point of the grid in both airfoils is $y^+ \leq 3$.

The level and distribution of the first three harmonics in the stator are correct overall. However, the second and third harmonics are underpredicted in the rear part of the airfoil, probably due to an excessive dissipation of the wake during the propagation.

Rotor Midsection Validation

The unsteady pressure on the rotor blades is induced by incident acoustic modes, radiating from the stator, with frequencies of $\omega_m = m \times 16\ \Omega$ and spatial wave numbers of $k = m \times 16 - n \times 18$. The induced unsteady rotor-blade loading consists of frequencies of $\omega_n = n \times 18\ \Omega$, multiples of the stator-vane passing frequency. It is important to highlight that every wave number of the stator contributes to each harmonic of the rotor.

Another source of unsteadiness to the rotor is the steady flow distortion caused by the thickness of the stator vanes. This gives rise to wave numbers $k = -n \times 18$, and the corresponding frequencies in the rotor-blade frame of reference are again $\omega_n = n \times 18\ \Omega$.

Figure 19 displays the first and second harmonics of the unsteady pressure on the midsection of the rotor blade. The unsteady

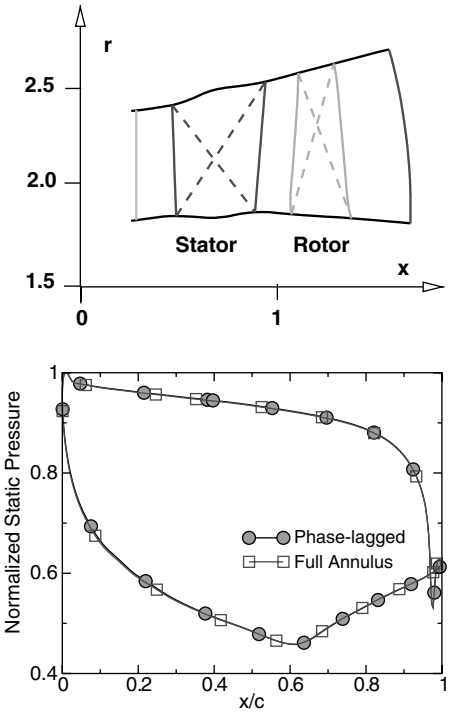

Fig. 21 Meridional view of the turbine (top) and time-averaged nondimensional static pressure in the midsection of the rotor (bottom).

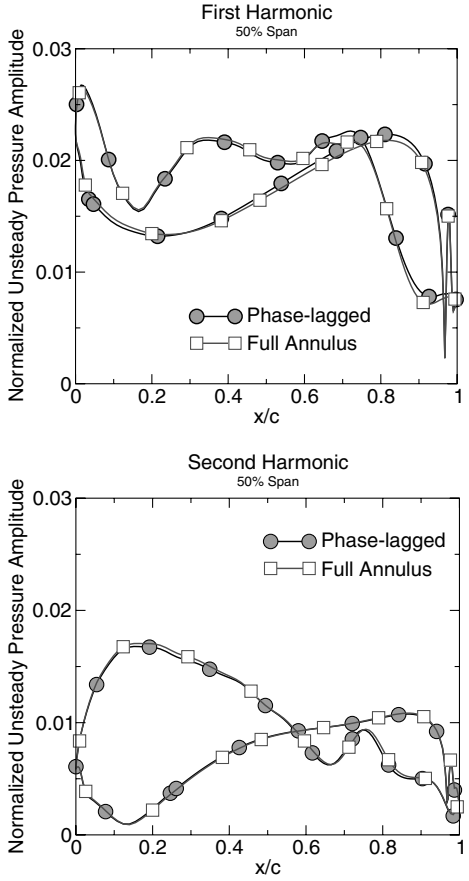

Fig. 22 First (top) and second (bottom) harmonics of the unsteady pressure amplitude in the rotor midsection of a high-speed turbine obtained with the full annulus (□) and the single-passage method (○).

component of the static pressure in the rotor blade is one order of magnitude lower than in the stator vane; however, this is well reproduced for both the LST and the Mu^2s^2T code. This is probably due to the fact that the wavelength of this acoustic perturbation is much larger than that of the vortical perturbations, and it is less affected by numerical diffusion.

Tip and Hub Results

The variation of the unsteady solution along the span is presented in Fig. 20. The first harmonic of the pressure jump close to the hub (30% span) and tip (76% span) shows that the unsteady variation along the span is significant, but it is well captured by the code. The

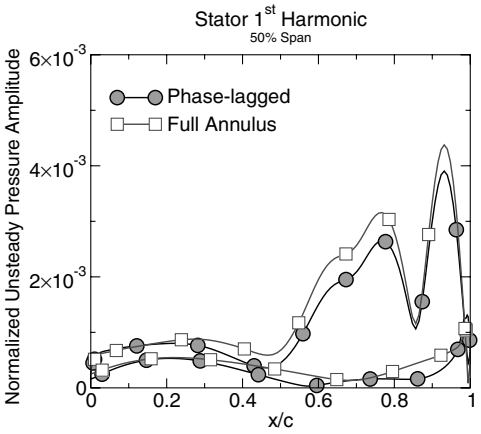

Fig. 23 First harmonic of the unsteady pressure amplitude in the stator midsection of a high-speed turbine obtained with the full annulus (□) and the single-passage method (○).

confidence in the overall 3-D performance of the code is backed by these results.

Helicopter Engine Turbine

The accuracy of the phase-lagged boundary conditions has been double checked by computing the 3-D stator/rotor interaction of a helicopter turbine and comparing the results using phase-shifted boundary conditions with those obtained computing the full annulus. On the contrary, other than the NLR fan, this case is highly loaded and may be used to assess the behavior of the phase lagged at high Mach number regimes. The original blade count of the turbine stage was 24/50 and was approximated to 25/50 in order to compare the unsteady static pressure computed in a sector of one stator and two rotor blades with the results obtained using phase-lagged boundary conditions. The total-to-static pressure ratio of the stage is 2.01, the mean exit Mach number of the stator and rotor are, respectively, 0.74 and 0.83. Figure 21 displays the meridional view of the turbine (top) and the time-averaged unsteady static pressure on the rotor for both the 1:2 and the single-passage configurations (bottom). The static pressure has been nondimensionalized using the relative inlet total pressure of the rotor. Figure 11 displays a close up of the Mach number isocontours in the sliding plane region.

Figure 22 displays the amplitude of the first and second harmonics of the unsteady pressure at the midsection of the rotor normalized with the rotor inlet relative total pressure. It may be seen that the differences between the results obtained simulating the whole wheel (1:2) and using phase-shifted boundary conditions are very small.

Figure 23 shows the first harmonic of the nondimensionalized unsteady static pressure in the midsection of the stator. The main differences with the rotor blade are the amplitude that, in this case, is about 10 times smaller and the distribution that, contrary to the rotor, has the highest values in the trailing-edge region since the unsteadiness is caused by the potential interaction of the stator with the rotor blade. The relative differences between the single-passage and the full annulus simulations are larger in the stator than in the rotor; this behavior was already observed in the NLR case. The absolute values of the differences are essentially the same in the rotor and the stator but are more clearly seen in the latter due to the lower level of the unsteadiness.

Computational Efficiency

The computational cost per physical time step for the fine case of the NLR fan case, which contains about 2×10^6 nodes, may be found in Table 1. The computational framework is a Linux cluster of a dual-core Xeon 5160 at 3 GHz[§] interconnected using a standard fast ethernet channel. The interpolation algorithm of the slide plane does not make use of the ADT in this case. The interpolation procedure may be very expensive, accounting for between 25 and 50% of the total cost. This cost is completely eliminated when the ADT is used in the seeking process of the interpolation procedure. This is a clear indication that the main cost is associated to the searches in the interface plane, since the actual cost of the interpolation is negligible compared with the evaluation of the fluxes. The result is essentially the same when a lookup table of the interpolation coefficients is stored for the whole period. The disadvantage of the latter approach is that it is very intensive in terms of memory. The second observation is that interpolation cost is constant, independent of the number of CPUs. This behavior is associated with the domain decomposition procedure, which currently requires that the whole interface is located within the same partition to avoid a dynamic update of the interprocessor communication matrix. This fact makes the use of an efficient interpolation procedure even more critical.

The computational cost in the coarse grid used for the verification of the phase lagged, which contains about 2.2×10^5 nodes, is 2.75 min. per physical time step in two processors, of which 0.85 min. correspond to the interpolation algorithm. The number of

[§]BlackCache memory: 4 Mb. RAM bus: 1.3 MHz, message-passing interface (MPI) implementation: MPICH.

Table 1 Computational cost per physical time step of the NLR fine case in a Xeon 5160 cluster with fast ethernet communication[a]

	Number of cores		
	4	9	19
Interpolation time, min.	5.5	5.5	5.5
Inner scheme time, min.	15.0	8.0	6.0
Total wall-clock time, min.	20.5	13.5	11.5

[a]The ADT is not used in the sliding plane.

Table 2 Computational cost per physical time step of the NLR fine case in a Xeon x5472 cluster interconnected using a InfiniBand network

Number of cores	Total wall-clock time, min.
4	9.9
9	4.2
19	2.5

nodes in the interface plane in the coarse grid is about 5500 while, in the fine grid, it is 8000. This means that a proportional scaling of the interface interpolation cost from the coarse to the fine grids would lead to 1.2 min., while the actual cost is 5.5 min. This is due to the fact that the brute force search algorithm does not scale linearly with the number of faces of the sliding plane, highlighting the need of an efficient method for the computation of the interface plane in fine meshes.

Finally, Table 2 contains the computation cost per physical time step of the fine grid in a Linux cluster of a quadcore Xeon-x5472@3GHz[¶] interconnected with a high-speed InfiniBand network. Only two out of the four cores of each CPU are used. The search algorithm makes use of the ADT, and the cost of the interpolation procedure in the sliding plane is negligible. It may be appreciated that not only are the processors faster (about 50% more), probably due to a larger cache memory, but the speedup with the number of cores is significantly better than in the previous case. This is due to both the use of the ADT and the high-speed network. Depending on the required accuracy, the results presented in the present work could be obtained in an elapsed time of 2–3 days.

The constraint that the whole sliding plane be contained in the same domain limits the scalability for a large number of CPUs when the size of the simulation is kept constant. The sliding plane creates an unbalance that finally stalls the speedup when the computational cost of the rest of the domains decreases with the partition size. This problem may be solved with a global array containing all the information of the opposite side of the sliding plane that would be updated for just the affected domains.

Conclusions

An efficient method for the simulation of the rotor/stator interaction problem in unstructured hybrid grids has been presented. The method uses an edge-based data structure and a multigrid method, which adds an extra complexity to the treatment of a rotor/stator interface, which is discussed in great detail. The numerical treatment of the sliding plane is conservative and second-order accurate in uniform grids. It is concluded that time spent in the communication between the stator and the rotor may become a significant part of the whole method unless efficient search algorithms, such as the ADT, are used. The method has been shown to be robust in a number of applications, and two of them are described in detail.

[¶]Cache memory: 12 Mb. RAM bus: 1.6 MHz. MPI implementation: MVAPICH

The equivalence between the single-passage and multipassage methods has been demonstrated using unstructured grids. A low-speed fan and a high-speed turbine have been used to assess this point. There are slight differences between the numerical treatment of the phase-lagged boundary conditions and the full annulus that induce a small error, which is more clearly seen in the low-speed case. Finally, the unsteady pressure data obtained at the NLR low-speed fan are compared with the present method. The results are considered quite satisfactory with the exception of the behavior of the second and third harmonics at the trailing edge of the stator.

Acknowledgments

This work has been partially funded under grant TRA2006-15015 by the Spanish Minister of Education and Science. The authors wish to thank Industria de Turbopropulsores, S. A., for allowing the publication of this paper and for support during the project. The authors also want to thank P. Sijtsma for completing the experimental data of the National Aerospace Laboratory (The Netherlands) low-speed fan.

References

[1] Tyler, J. M., and Sofrin, T. G., "Axial Flow Compressor Noise Studies," *SAE Transactions*, Vol. 70, 1962, pp. 309–332.

[2] Srinivasan, A. V., "Vibrations of Bladed-Disk Assemblies: A Selected Survey," *Journal of Vibration, Acoustics, Stress, and Reliability on Design*, Vol. 106, No. 2, 1984, pp. 165–168. doi:10.1115/1.3269162

[3] Hodson, H., and Howell, R., "Blade Row Interactions Transition and High-Lift Airfoils in Low-Pressure Turbines," *Annual Review of Fluid Mechanics*, Vol. 37, No. 1, Jan. 2005, pp. 71–98. doi:10.1146/annurev.fluid.37.061903.175511

[4] Novack, R., "Streamline Curvature Computing Procedures for Fluid-Flow Problems," *Journal of Engineering for Power*, Vol. 89, Oct. 1967, pp. 478–490.

[5] Jennions, I., and Stow, P., "A Quasi-Three Dimensional Blade Design System. Part I: Throughflow Analysis," *Journal of Engineering for Gas Turbines and Power*, Vol. 107, No. 2, April 1985, pp. 301–307. doi:10.1115/1.3239715

[6] Adamczyk, J., "Aerodynamic Analysis of Multistage Turbomachinery Flows in Support of Aerodynamic Design," *Journal of Turbomachinery*, Vol. 122, No. 2, 2000, pp. 189–217. doi:10.1115/1.555439

[7] Dawes, W., "Toward Improved Throughflow Capability: The Use of Three-Dimensional Viscous Flow Solver in a Multistage Environment," *Journal of Turbomachinery*, Vol. 114, No. 1, 1992, pp. 8–17. doi:10.1115/1.2928002

[8] Dawes, W., "Simulating Unsteady Turbomachinery Flows on Unstructured Meshes that Adapt both in Space and Time," American Society of Mechanical Engineers Paper 93-GT-104, Fairfield, NJ, May 1993.

[9] Sayma, A., Vahdati, M., Sabardella, L., and Imregun, M., "Modelling of 3D Viscous Compressible Turbomachinery Flows Using Unstructured Hybrid Grids," *AIAA Journal*, Vol. 38, No. 6, 2000, pp. 945–954. doi:10.2514/2.1062

[10] Erdos, J., Alzner, E., and McNally, W., "Numerical Solution of Periodic Transonic Flow through a Fan Stage," *AIAA Journal*, Vol. 15, No. 11, 1977, pp. 1559–1568. doi:10.2514/3.60823

[11] He, L., "An Euler Solution for Unsteady Flows Around Oscillating Blades of Turbomachinery," *Journal of Turbomachinery*, Vol. 112, No. 4, 1990, pp. 714–722. doi:10.1115/1.2927714

[12] Dewhurst, S., and He, L., "Unsteady Flow Calculations through Turbomachinery Stages Using Single Passage Domain with Shape Correction Method," *9th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, edited by P. Ferrand and S. Aubert, Presses Univ. de Grenoble, Lyon, France, Sept. 2000, pp. 338–354.

[13] Burgos, M., and Corral, R., "Comparison and Validation of Phase-Shifted Boundary Conditions," *9th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, edited by P. Ferrand and S. Aubert, Presses Univ. de Grenoble, Lyon, France, Sept. 2000, pp. 296–309.

[14] Corral, R., Crespo, J., and Gisbert, F., "Parallel Multigrid Unstructured Method for the Solution of the Navier–Stokes Equations," 42nd AIAA

Aerospace Sciences Meeting and Exhibit, AIAA Paper 2004-0761, Jan. 2004.

[15] Roe, P., "Approximate Riemann Solvers, Parameter Vectors, And Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5

[16] Moinier, P., "Algorithm Developments for an Unstructured Viscous Flow Solver," Ph.D. Thesis, Univ. of Oxford, U. K., 1999.

[17] Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, Philadelphia, 1998.

[18] Jameson, A., "Time Dependent Calculations Using Multigrid with Applications to Unsteady Flows past Airfoils and Wings," AIAA Paper 1991-1596, 1991.

[19] Martinelli, L., "Calculations of Viscous Flow with a Multigrid Method," Ph.D. Thesis, Princeton Univ., Princeton, NJ, 1987.

[20] Nyukhtikov, M., Smelova, N., Mitchell, B., and Holmes, D., "Optimized Dual-Time Stepping Technique for Time-Accurate Navier–Stokes Calculations," *10th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, Durham, NC, edited by K. C. Hall, R. E. Kielb, and P. Thomas, Springer, Dordrecht, The Netherlands, Sept. 2003, pp. 449–462.

[21] Hills, N., "Achieving High Parallel Performance for an Unstructured Unsteady Turbomachinery CFD Code," *The Aeronautical Journal*, Vol. 111, No. 1117, March 2007, pp. 185–193.

[22] Moinier, P., and Giles, M., "Preconditioned Euler and Navier–Stokes Equations on Unstructured Grids," *6th ICFD on Numerical Methods for Fluid Dynamics*, Oxford, U. K., 1998.

[23] Cali, P., Couallier, V., and Jameson, A., "Conservative Interfacing for turbomachinery Applications," Proceedings of the 46th ASME Gas Turbine and Aero Engine Congress, Exposition and Users Symposium, American Society of Mechanical Engineers Paper 2001-GT-0357, Fairfield, NJ, June 2001.

[24] Yang, H., Nuernberger, D., and Kersken, H., "Towards Excellence in Turbomachinery CFD: Hybrid Structured-Unstructured RANS Solver," *Journal of Turbomachinery*, Vol. 128, No. 2, 2006, pp. 390–402. doi:10.1115/1.2162182

[25] Koya, M., and Kotake, S., "Numerical Analysis of Fully Three-Dimensional Periodic Flows through a Turbine Stage," *Journal of Engineering for Gas Turbines and Power*, Vol. 107, No. 4, Oct. 1985, pp. 945–952. doi:10.1115/1.3239840

[26] Burgos, M., and Corral, R., "Application of Phase-Lagged Boundary Conditions to Rotor/Stator interaction," 46th ASME Gas Turbine and Aeroengine Congress, American Society of Mechanical Engineers Paper 2001-GT-0586, Fairfield, NJ, June 2001.

[27] Sijtsma, P., Rademaker, E., and Schulten, J., "Experimental Validation of Lifting Surface Theory for Rotor-Stator Interaction," *AIAA Journal*, Vol. 36, No. 6, June 1998, pp. 900–906. doi:10.2514/2.493

[28] Burgos, M., Contreras, J., and Corral, R., "Numerical Treatment of the Rotor/Stator Interaction Problem in Unstructured Grids," 18th AIAA Fluid Dynamics Conference, AIAA Paper 2007-4474, June 2007.

[29] Contreras, J., Corral, R., Fernández-Castañeda, J., Pastor, G., and Vasco, C., "Semiunstructured Grid methods for Turbomachinery Applications," 47th ASME Gas Turbine and Aeroengine Congress, American Society of Mechanical Engineers Paper 2002-GT-30572, Fairfield, NJ, June 2002.

S. Fu
Associate Editor