AIAA JOURNAL
Vol. 37, No. 3, March 1999

Flutter Computation of Turbomachinery Cascades Using a Parallel Unsteady Navier–Stokes Code

Shanhong Ji* and Feng Liu†
University of California, Irvine, Irvine, California 92697-3975

A quasi-three-dimensional multigrid Navier–Stokes solver on single- and multiple-passage domains is presented for solving unsteady flows around oscillating turbine and compressor blades. The conventional direct store method is used for applying the phase-shifted periodic boundary condition over a single blade passage. A parallel version of the solver using the message passing interface standard is developed for multiple-passage computations. In parallel multiple-passage computations, the phase-shifted periodic boundary condition is converted to a simple in-phase periodic condition. Euler and Navier–Stokes solutions are obtained for unsteady flows through an oscillating turbine cascade blade row with both the sequential and the parallel codes. It is found that the parallel code offers almost linear speedup with a slope close to 1 on multiple CPUs. In addition, significant improvement is achieved in convergence of the computation to a periodic unsteady state in the parallel multiple-passage computations due to the use of the in-phase periodic boundary conditions compared with that in the single-passage computations with phase-shifted periodic boundary conditions via the direct store method. The parallel Navier–Stokes code is also used to calculate the flow through an oscillating compressor cascade. Results are compared with experimental data and computations by other authors.

Nomenclature

Cp_1	= first harmonic unsteady pressure coefficient
c	= blade chord length
E	= total energy per unit mass
e	= internal energy per unit mass
H	= total enthalpy per unit mass
h	= enthalpy per unit mass $(e + p/\rho)$
k	= turbulence mixing energy
Pr_L	= laminar Prandtl number
Pr_T	= turbulence Prandtl number
p	= pressure
t	= time
u, v	= velocity components in x and y directions
$\bar{u}, \bar{v}$	= relative velocity components $(u - u_b, v - v_b)$
u_b, v_b	= grid velocity components in x and y directions
x, y	= spatial coordinate components
γ	= ratio of specific heats
$\theta(x)$	= stream tube thickness ratio in x direction
κ	= reduced frequency $(\Omega c/2U_\infty)$
μ	= molecular viscosity
μ_T	= turbulence eddy viscosity
ρ	= density
Ω	= angular frequency
ω	= turbulence specific dissipation rate

Subscripts

i	= when 1, vector component in x direction; when 2, vector component in y direction
l	= lower side of the passage
u	= upper side of the passage
∞	= inlet

Superscript

n	= discrete time index

Received June 22, 1998; revision received Oct. 12, 1998; accepted for publication Oct. 12, 1998. Copyright © 1998 by Shanhong Ji and Feng Liu. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Graduate Student Researcher, Department of Mechanical and Aerospace Engineering; currently Engineer, CFD Group, Adina R & D, Inc., 71 Elton Avenue, Watertown, MA 02172. Member AIAA.

†Associate Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.

I. Introduction

THE energy method[1] is widely used in predicting flutter boundaries. In this method one calculates the unsteady aerodynamics for a given blade vibration mode and then determines system stability based on the net energy transfer. As such, the prediction of the unsteady flowfield for a given blade motion is of essential importance.

Substantial efforts have been expended in developing reliable and accurate computational procedures for predicting unsteady flows within turbomachinery passages. Early methods applied time-linearized models in a global[2] or local[3] sense. With the rapid advances in computational fluid dynamics (CFD) and computer technology, unsteady solutions of the full Euler and the Navier–Stokes (N–S) equations by time-marching methods become more and more popular. Such computations remove the assumption of time linearization. Solutions of the Euler equations by the time-marching method for unsteady flows in oscillating blades were developed by Joubert,[4] Koya and Kotake,[5] Gerolymons,[6] and He.[7] Computations of the unsteady N–S equations for turbomachinery flows can be found in the work of Kikuchi et al.,[8] Scott and Hankey,[9] and Rai and Madavan.[10] Hwang and Liu[11] computed two-dimensional steady and unsteady turbine cascade flows by both the Euler and the N–S equations with the Baldwin–Lomax model. Abhari and Giles[12] computed aerodynamic damping using a hybrid structured and unstructured grid. He and Denton[13] obtained three-dimensional unsteady inviscid and viscous flow solutions over vibrating blades. A good review of numerical simulation of unsteady flows in turbomachines can be found in the work of He.[14]

A dilemma in unsteady flow calculations is the choice of implicit or explicit time-marching schemes. Implicit schemes allow the use of large time steps but usually require large computational time for each time step. Explicit schemes cost little per time step but may use only small time steps due to the numerical stability limit. The explicit time-marching method of the Runge–Kutta type initially proposed by Jameson et al.[15] has been widely used in the aerodynamics community and proved very efficient for steady-state solutions when coupled with local time steps, residual averaging, and multigrid. In an effort to take advantage of the existing fast algorithms for steady flows, Jameson[16] extended the use of multigrid for explicit finite volume schemes in an implicit time-stepping scheme of solving the unsteady Euler equations by reformulating the implicit equations for each unsteady time step into a pseudo-time-marching problem so that the converged result of the pseudo-time marching yields a time-accurate solution for one time step in the original implicit real-time marching. Using this concept of dual time stepping,

the authors of this paper developed a multigrid time-accurate N–S code with the k–ω two-equation turbulence model and calculated the unsteady transonic flow over an airfoil in pitching motion,[17] self-excited unsteady transonic flow around an airfoil in a channel,[18] and aerodynamic damping for vibrating cascades.[19]

In a typical flutter calculation, the adjacent blades in a blade row are assumed to vibrate with a constant phase difference, the interblade phase angle (IBPA). Single blade passages are usually used to minimize the computational effort. Consequently, a phase-shifted periodic boundary condition has to be applied when the IBPA is not zero. A conventional way of implementing this condition is the direct store method by Erdos and Alzner,[20] in which the flow variables on the periodic boundaries are stored over a whole period of time. These stored time histories of the flow are then used to update solutions at corresponding periodic boundaries at a later time. When the computation reaches a periodic solution, the phase-shifted boundary condition is then satisfied with the specified IBPA.

A main disadvantage of this method is the requirement of a large amount of computer storage, especially for an explicit time-marching scheme. To overcome this problem, Giles[21] proposed a novel space–time transformation method in which the computation time plane was inclined along the blade pitch direction according to the given IBPA. After this transformation, the transformed flow variables will exhibit zero IBPA so that the ordinary in-phase periodic condition may be applied. However, the time-inclination angle and, therefore, the IBPAs are restricted by the characteristics of the governing equations.

He[7] proposed a Fourier-transform-based method, called the shape correction method, in which the Fourier components of the flow variables are stored instead of the time history of the flow variables themselves. Although the need for computer storage is greatly reduced in this method, extra computational load is added.

The direct store method is used in the previous work of the authors.[19] Because we used an implicit method, fewer physical time steps are needed within each period of the blade oscillation. Hence, the memory needed to store one period of the flowfield is much less than that by an explicit method. However, it was found that the computation converges slowly in reaching a final periodic solution by using the direct store method because the stored history of the flowfield that is used to set the boundary condition in the computation of the next period is from a less converged previous solution. The phase-shifted boundary condition is satisfied only when the computation reaches a final converged periodic solution. In this sense, the phased-shifted periodic boundary condition is applied in an iterative manner.

The use of the phase-shifted boundary condition implies that the solution is spatially periodic. In fact, the direct store method also assumes time periodicity. For many phenomena, there are components of the flow that may be nonperiodic in either space or time or periodic but with multiple nonharmonic frequencies. Computation for such flows must be done over multiple passages. As parallel computers and networked clusters of workstations with distributed memory become more and more commonplace and as standard software such as parallel virtual machine (PVM) and message passing interface (MPI) libraries become available for writing parallel computer codes, the authors feel that it may be much easier and more effective to compute multiple passages by utilizing multiprocessors in parallel and to eliminate the use of the phase-shifted periodic boundary condition. In such an approach, the elapsed computational time is kept the same while the problem to be solved can be scaled up linearly with the number of CPUs available.

In the present work, the multigrid time-accurate N–S code with a two-equation k–ω turbulence model developed in the authors' previous work is extended to calculate quasi-three-dimensional unsteady flows around oscillating turbine blades. The code is made parallel by using MPI so that multiple passages can be calculated without the use of phase-shifted periodic boundary conditions for blade flutter problems. In the following sections, we will describe the parallel method and present results that show that the parallel speedup scales almost linearly with the number of CPUs used. Moreover, the computation converges much faster to the final periodic solution for the flutter problem considered with the multiple-passage code than with the single-passage code because we do not need to rely on the iterative direct store method to enforce the phase-shifted periodic boundary condition.

With the availability of parallel computers consisting of hundreds or even thousands of CPUs, it becomes feasible to simulate a broad range of periodic or nonperiodic flows in complete blade rows or multiple stages with the proposed method.

II. Governing Equations

Consider a moving and possibly deforming control volume V in two dimensions whose boundary ∂V moves at velocity $\boldsymbol{V}_b = (u_b, v_b)$. The quasi-three-dimensional Favre-averaged N–S equations for an unsteady compressible turbulent flow with a k–ω turbulence model by Wilcox[22] can be written in integral form for such a control volume as follows:

$$\frac{\partial}{\partial t}\iint_V \theta(x)\boldsymbol{w}\,\mathrm{d}V + \oint_{\partial V} \theta(x)\boldsymbol{f}\,\mathrm{d}S_x + \theta(x)\boldsymbol{g}\,\mathrm{d}S_y = \oint_{\partial V} \theta(x)\boldsymbol{f}_\mu\,\mathrm{d}S_x + \theta(x)\boldsymbol{g}_\mu\,\mathrm{d}S_y + \iint_V \boldsymbol{S}\,\mathrm{d}V \tag{1}$$

where $\theta(x)$ is introduced to account for the streamtube thickness variation along the axial direction of the blade row and where

$$\boldsymbol{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho k \\ \rho \omega \end{bmatrix} \tag{2}$$

$$\boldsymbol{f} = \begin{bmatrix} \rho\bar{u} \\ \rho u\bar{u} + p \\ \rho v\bar{u} \\ \rho E\bar{u} + pu \\ \rho k\bar{u} \\ \rho\omega\bar{u} \end{bmatrix}, \qquad \boldsymbol{g} = \begin{bmatrix} \rho\bar{v} \\ \rho u\bar{v} \\ \rho v\bar{v} + p \\ \rho E\bar{v} + pv \\ \rho k\bar{v} \\ \rho\omega\bar{v} \end{bmatrix} \tag{3}$$

$$\boldsymbol{f}_\mu = \begin{bmatrix} 0 \\ \hat{\tau}_{xx} \\ \hat{\tau}_{yx} \\ u\hat{\tau}_{xx} + v\hat{\tau}_{yx} + (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial x} - q_x \\ (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial x} \\ (\mu + \sigma\mu_T)\dfrac{\partial \omega}{\partial x} \end{bmatrix} \tag{4}$$

$$\boldsymbol{g}_\mu = \begin{bmatrix} 0 \\ \hat{\tau}_{xy} \\ \hat{\tau}_{yy} \\ u\hat{\tau}_{xy} + v\hat{\tau}_{yy} + (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial y} - q_y \\ (\mu + \sigma^*\mu_T)\dfrac{\partial k}{\partial y} \\ (\mu + \sigma\mu_T)\dfrac{\partial \omega}{\partial y} \end{bmatrix} \tag{5}$$

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ \left[p - 2(\mu + \mu_t)\dfrac{u}{\theta(x)}\dfrac{\partial\theta(x)}{\partial x} + \dfrac{2}{3}(\mu + \mu_t)\dfrac{\partial u_i}{\partial x_i}\right]\dfrac{\partial\theta(x)}{\partial x} \\ 0 \\ 0 \\ \theta(x)\left(\tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta^*\rho\omega k\right) \\ \theta(x)\left(\dfrac{\alpha\omega}{k}\tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta\rho\omega^2\right) \end{bmatrix} \tag{6}$$

$$\mu_T = \alpha^*(\rho k/\omega) \tag{7}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{8}$$

$$\tau_{ij} = 2\mu_T\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - 2/3\rho k\delta_{ij} \tag{9}$$

$$\hat{\tau}_{ij} = 2\mu\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) + \tau_{ij} \tag{10}$$

$$q_j = -\left(\frac{\mu}{Pr_L} + \frac{\mu_T}{Pr_T}\right)\frac{\partial h}{\partial x_j} \tag{11}$$

where $\bar{u}$ and $\bar{v}$ are the relative velocity components to the surface of the control volume, τ_{ij} is the turbulence shear stress tensor, and $\hat{\tau}_{ij}$ is the total shear stress tensor. The total energy and enthalpy are $E = e + k + u_i u_i/2$ and $H = h + k + u_i u_i/2$, respectively, with $h = e + p/\rho$ and $e = [p/(\gamma - 1)\rho]$. The molecular viscosity μ is calculated by Sutherland's law. The closure coefficients in the k–ω two-equation turbulence model are

$$\beta = \tfrac{3}{40}, \qquad \beta^* = \tfrac{9}{100}, \qquad \alpha = \tfrac{5}{9}$$
$$\alpha^* = 1, \qquad \sigma = \tfrac{1}{2}, \qquad \sigma^* = \tfrac{1}{2} \tag{12}$$

III. Dual Time Stepping with Multigrid

Equation (1) can be discretized in space by a staggered finite volume scheme described by Liu and Zheng.[23,24] The convective terms of the N–S equations are discretized by either the standard Jameson–Schmidt–Turkel[15] (JST) scheme or an upwind scheme using the splitting by Roe.[25] Computations in the present paper are done using the JST scheme. The convective terms of the k–ω turbulence equations are discretized by a second-order upwind scheme using a splitting based on the sign of the flow velocity. The diffusion terms of both the N–S and the turbulence model equations are discretized by using central differencing. Details of the discretization can be found in Refs. 23 and 24.

After the spatial discretization, Eq. (1) can be written in the following semidiscrete form:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} + \boldsymbol{R}(\boldsymbol{w}) = 0 \tag{13}$$

where $\boldsymbol{w}$ is the vector of flow variables at each mesh point and $\boldsymbol{R}$ is the vector of the residuals, consisting of the spatially discretized flux balance of Eqs. (1–6). This system of equations can be integrated in time by a suitable time-marching method.

As shown by Liu and Zheng,[24] fast convergence could be achieved by a strongly coupled multigrid method for the N–S and the k–ω equations. However, the multigrid method is applicable only to steady flow calculations because time accuracy is lost through the use of different time steps on the different levels of grids and also because of the use of local time steps that are usually employed in combination with the multigrid method for solving steady-state problems.

To take advantage of the fast convergence property of the multigrid method without sacrificing time accuracy, Eq. (13) is reformulated by following the idea from Jameson[16]:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t^*} + \boldsymbol{R}^*(\boldsymbol{w}) = 0 \tag{14}$$

where

$$\boldsymbol{R}^*(\boldsymbol{w}) = \frac{3\boldsymbol{w}V^{n+1} - 4\boldsymbol{w}^n V^n + \boldsymbol{w}^{n-1}V^{n-1}}{2\Delta t V^{n+1}} + \boldsymbol{R}(\boldsymbol{w}) \tag{15}$$

where t^* is a pseudotime. The solution of the implicit equation (13) is now made equivalent to the steady-state solution of Eq. (14) with pseudotime t^*.

With the preceding reformulation, existing techniques for steady-state calculations such as local time steps, residual smoothing, and the strongly coupled multigrid method described in Ref. 24 may then be used to efficiently solve Eq. (14) for each physical time step Δt.

IV. Motion of the Computational Grid

The computational domain and, hence, the grid move with time as the blades vibrate. The finite volume formulation described in Sec. II includes the effect of grid motion. However, the spatial location and velocity of each grid point must be computed at every time step. In our computation, an initial H grid at time zero is generated with an elliptic grid generator. At every physical time step, the new blade position is recalculated according to the specified blade motion. The grid points on the blade surface move with the blades. The grid points at inflow and outflow boundaries and at the centerline within each blade passage are kept stationary. The location and grid velocity of the rest of the grid points are obtained by linear interpolation based on their distances to the boundaries and the centerline.

V. Single-Passage Computation

For a single blade passage with upper and lower periodic boundaries, the periodic condition has to be applied as

$$\boldsymbol{w}_l(x,t) = \boldsymbol{w}_u[x, t - (\sigma/\Omega)] \tag{16}$$

where σ is the IBPA, Ω is the blade vibration angular frequency, and subscripts l and u denote lower and upper boundaries. For the case with nonzero σ, the direct store method by Erdos and Alzner[20] is used to apply the phase-shifted periodic conditions. In the present implementation, the six conservative variables defined in Eq. (2) at the periodic boundaries are stored for a period of oscillation. At every physical time step, these flow variables at the boundaries are updated as a weighted average of the data obtained from the current time-marching solution and those stored according to a given IBPA. The condition given in Eq. (16) can be completely satisfied when the time-marching process converges to a periodic solution.

Standard configuration 4 of a turbine cascade from Ref. 26 is used as a first test case. In the experiment, the turbine blades are under harmonic bending motion with variable IBPA. A detailed description of the geometry and working conditions is given by Bolcs and Fransson.[26] Experimental steady and unsteady pressure distributions and damping coefficients at different working conditions are also provided in Ref. 26. Both Euler and N–S solutions are obtained by the present method. The Euler calculations were done on a mesh of 97×25 grid points. The N–S calculations were done on a mesh of 113×65 grid points. He[7] did Euler calculations for this test case. Abhari and Giles[12] calculated this case using an unsteady N–S code. To match the experimental conditions, the stream tube area divergence ratio $\theta(x)$ is chosen to vary linearly from 1.0 at the blade leading edge to 1.1 at the blade trailing edge, as in Ref. 7.

The steady-state solution is first obtained. Figure 1 shows the computed steady-state isentropic Mach number distribution compared with the experimental data. It can be seen that both our Euler and N–S solutions in general agree well with the experimental data. Although the N–S solution does seem to give better agreement with

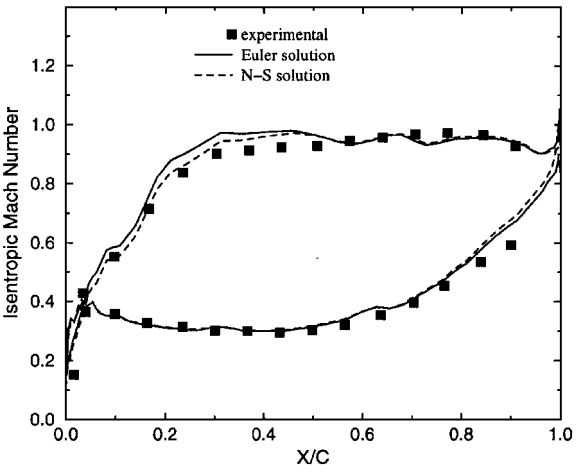

Fig. 1 Steady-state isentropic Mach number distribution over the blade.

the experimental data on the suction surface, the difference between the Euler and the N–S solutions is small, indicating that the viscous effect is not an important factor for this case.

After getting the steady-state solution, the cascade blades are then set to motion, and the computation is shifted to a time-accurate unsteady mode. In the experiment, the blades are oscillated in a bending mode. The computation was performed with different IBPAs under the condition of a constant inlet-to-exit-pressure ratio corresponding to an isentropic exit Mach number of 0.9. The blades are under pure harmonic bending motion with a reduced frequency $\kappa = 0.115$. Convergence of the unsteady computations is measured by

$$\max\left|\frac{\boldsymbol{w}^n - \boldsymbol{w}^{n-N_p}}{\boldsymbol{w}^n}\right| \le \delta \tag{17}$$

where N_p is the number of time steps for one oscillation period. In the computation, δ is set to be 0.001. The solution can reach such a level of convergence in about 10 periods of real-time marching when using the direct store method.

The calculated amplitude and phase distributions of the first harmonic unsteady-pressure coefficient are plotted in Figs. 2–5 for IBPA = 90 and 180 deg. It can be seen that the trend of the computational results matches the experiment. Although there are discrepancies for the amplitude prediction, the calculated phase angle fits the experimental data better. Hence, it can be expected that the magnitude of the aerodynamic damping coefficient might differ from the experiment, but the stability boundaries should approach the experimental result. It also can be seen that the N–S solution slightly

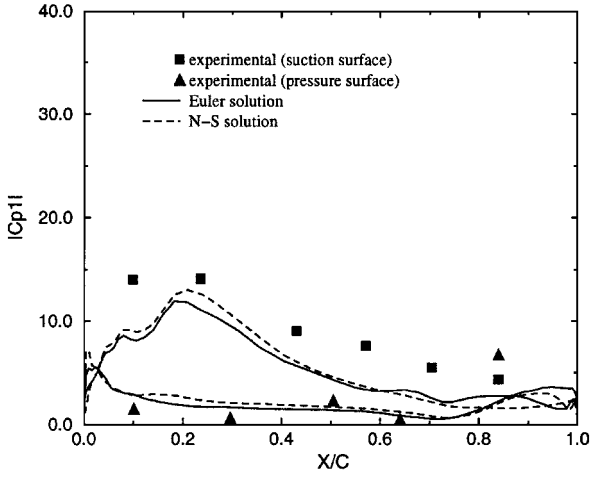

Fig. 2 Amplitude of the first harmonic unsteady-pressure coefficient over the blade with 90-deg IBPA.

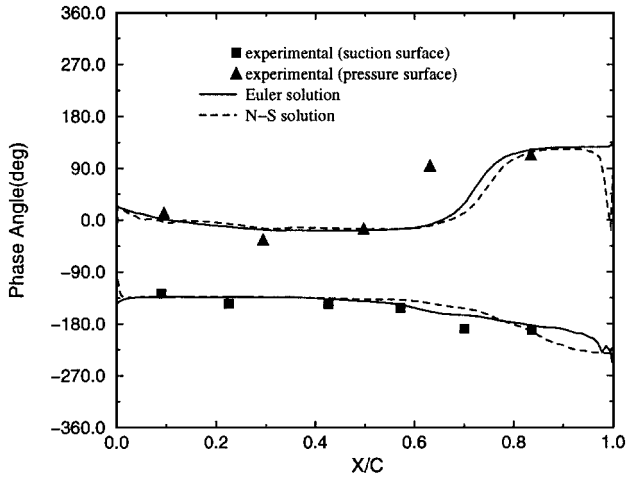

Fig. 3 Phase of the first harmonic unsteady-pressure coefficient over the blade with 90-deg IBPA.

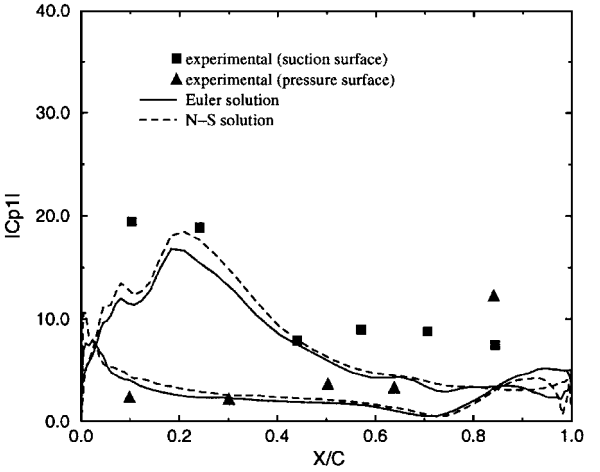

Fig. 4 Amplitude of the first harmonic unsteady-pressure coefficient over the blade with 180-deg IBPA.

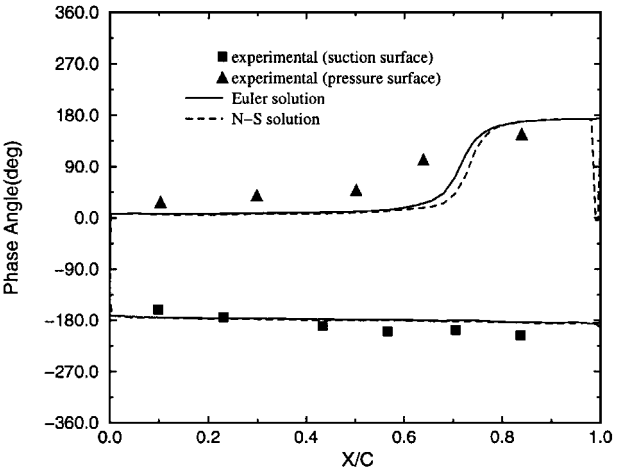

Fig. 5 Phase of the first harmonic unsteady-pressure coefficient over the blade with 180-deg IBPA.

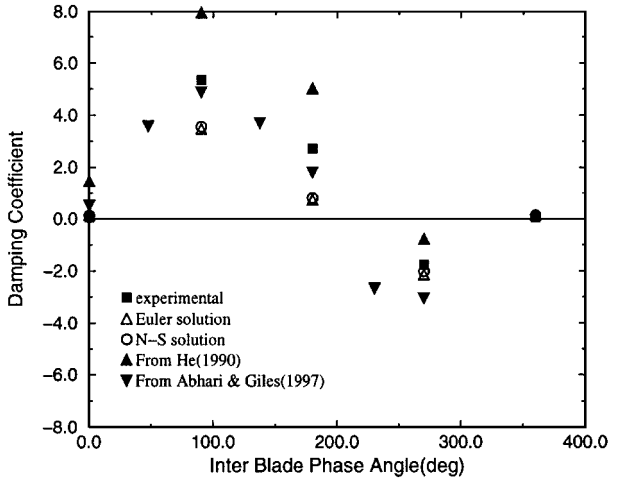

Fig. 6 Damping coefficient computed on a single blade passage.

improves the computational solution. Figure 6 plots the calculated damping coefficients vs IBPAs. The Euler solution by He[7] and the viscous solution by Abhari and Giles[12] are also plotted in Fig. 6. A positive damping coefficient indicates a stable motion. As anticipated from the result of the pressure distributions, the computed damping coefficient deviates from the experimental data in magnitude but agrees with the experimental data in sign. This ensures that the stability region is correctly predicted.

VI. Parallel Multiple-Passage Computation

Most parallel algorithms use the approach of divide and conquer, in which a big problem is divided into many small problems to be solved on each CPU of a parallel computer system. Domain decomposition is commonly utilized to do the dividing part, and message passing is used to achieve communication between CPUs with distributed memory. To do domain decomposition, an initially single-block CFD program is often converted to a multiblock program. In the simplest case, the computation of each block is assigned to one CPU and standard libraries such as PVM[27] or MPI[28] can be used to do the message passing between neighboring blocks. If N CPUs are used to solve one problem, a parallel speedup measure can be defined as

$$\text{speedup} = \frac{\text{time needed to compute the problem on one CPU}}{\text{time needed to compute the same problem on } N \text{ CPUs}}$$

where the time should be the wall clock time. Ideally, this should be a linear curve vs N with a slope of 1 if the communication time is negligible.

In turbomachinery calculations the most convenient domain decomposition seems to be the division of the blade passages within each blade row. A program originally written for a single blade row passage can be easily converted to run in a parallel manner on multiple CPUs by assigning each blade passage to one CPU and by using PVM or MPI to communicate between the blade passages, as shown in Fig. 7. Two layers of dummy cells for each blade passage are used to assist in the information exchange and the setting up of the boundary conditions, as shown in Fig. 8. Because we use an explicit pseudo-time-marching scheme for the implicit real-time marching, all of the PVM or MPI calls need to be done only in the initial setup and the subroutine for the original periodic boundary conditions, which need not be periodic or even phased-shifted periodic any more. Communication is done at every multiple grid level and every stage of the Runge–Kutta time stepping. This requires minimum modification of the code. In this approach, however, the original problem is not divided into smaller problems. Instead, the problem is scaled up while the computational effort on each CPU is kept the same. The parallel speedup may then be defined as

$$\text{speedup} = \frac{N \times (\text{time to do one passage sequentially})}{\text{time needed to compute } N \text{ passages on } N \text{ CPUs}} \quad (18)$$

This approach may be called a coarse-grain parallelization. With the ever-increasing power of a single CPU, this coarse-grain approach may offer a better load balancing than the usual divide-and-conquer approach. It may also leave room for automatic parallelization by compilers on systems using hybrid shared and distributed memory schemes and clusters of CPUs.

With this parallelization scheme, our computation is not limited to periodic solutions, provided that enough blade passages are used. For flutter computations, the number of passages is chosen so that the unsteady flow and the blade vibration on the top and bottom of the domain are in phase. For example, Fig. 7 may represent a scheme for solving the flow with an IBPA of ±90 deg. For each passage, the phase difference between two neighboring blades is 90 deg. This makes the upper blade of passage 4 become in phase with the lower blade of passage 1. Consequently, the simple in-phase periodic boundary conditions may be applied between the lower boundary of blade passage 1 and the upper boundary of blade passage 4. In general, the minimum IBPA is related to the number of passages used by the following equation:

$$\sigma(\text{deg}) = \pm\frac{360}{\text{number of passages}}$$

To test this parallel algorithm, the same turbine test case discussed in the preceding section is computed by the parallel code. The code is tested on a 16-processor Hewlett Packard/Convex (HP/Convex) Examplar SPP2000 and the Aeneas parallel cluster of 16 300-MHz Pentium II personal computers running Linux and connected by a 100-Mb network. Figure 9 shows the calculated damping coefficient by the N–S solver. It can be seen that the solution on the multiple passages by the parallel code is practically identical to that by the single-passage sequential code with the direct store method.

Although message passing is done within every pseudo-time step in the parallel computation, the amount of message transferred is still small compared with the number of operations needed to compute the flow over the interior cells. The wall clock time needed to complete such a parallel run on multiple passages is almost the same as that needed to compute the flow over a single passage on a single processor with the sequential code without any message passing. Figure 10 shows the parallel speedup as defined by Eq. (18) for runs on the HP/Convex SPP2000 parallel computer and the Aeneas personal computer cluster. As can be seen, an almost linear speedup with slope close to 1 is achieved with both platforms. The linearity of these curves indicates that the message passing has not saturated the communication bandwidth of the system yet as the number of CPUs used by the computation increases for the given maximum

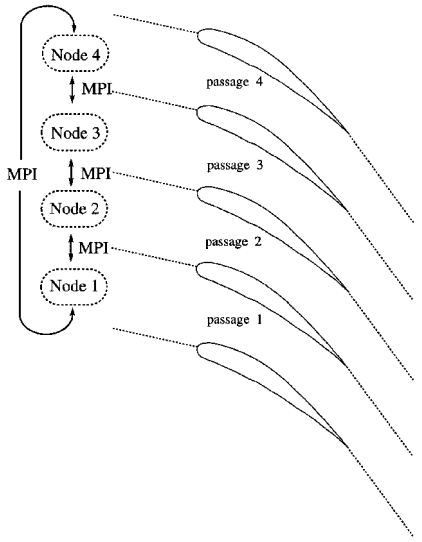

Fig. 7 Multiple-passage computation using MPI.

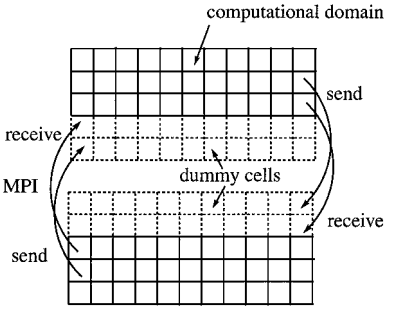

Fig. 8 Data transfer between different passages to set flow variables in dummy cells.

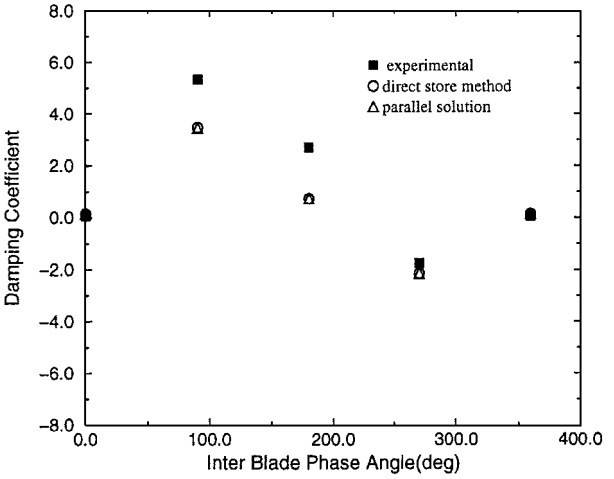

Fig. 9 Comparison of damping coefficient computed on a single-blade passage and that computed on multiple passages by the parallel code.

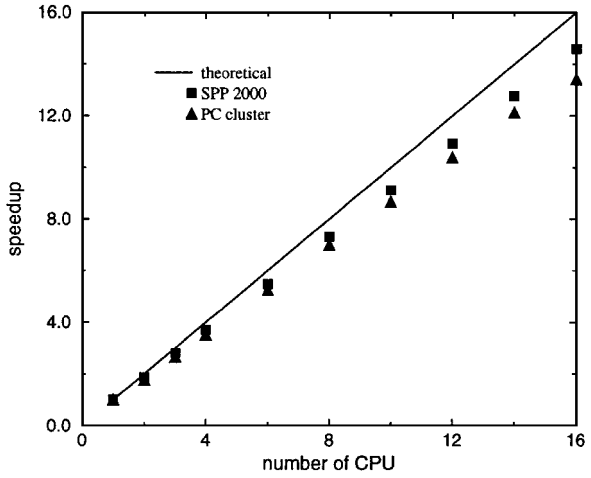

Fig. 10 Parallel speedup for the multiple-passage calculations.

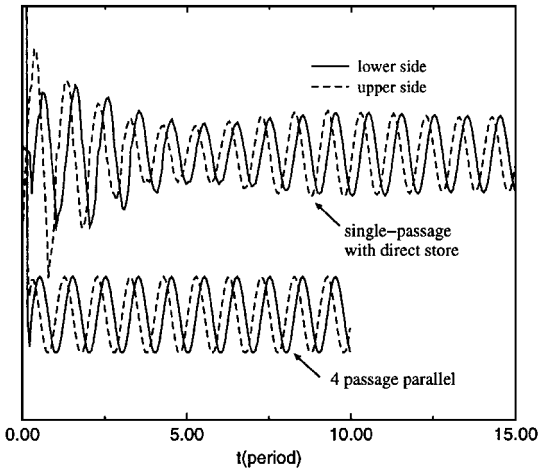

Fig. 11 Time history of the pressure at a grid point ahead of the blade leading edge.

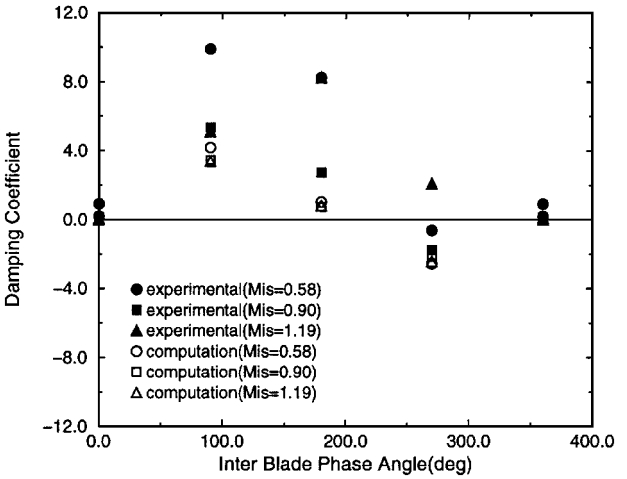

Fig. 12 Aerodynamic damping for different isentropic exit Mach number with different IBPAs.

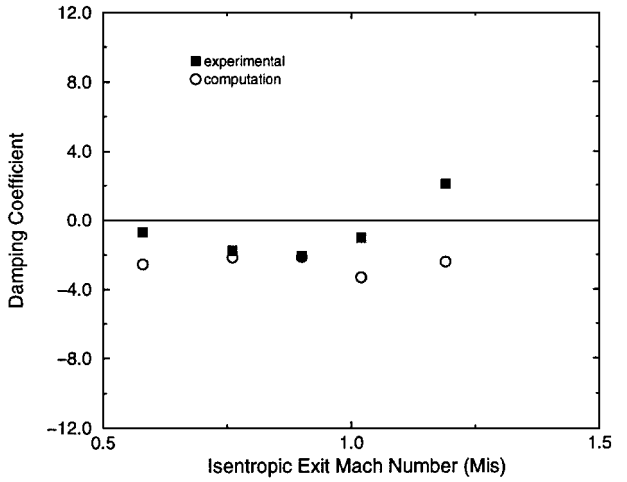

Fig. 13 Aerodynamic damping for different isentropic exit Mach number with 270-deg IBPA.

number of CPUs available to us at the time. The HP/Convex machine has much faster message-passing network connection than the personal computer clusters. Consequently, the computation on the HP/Convex performs better than that on the personal computer cluster with slower network, as demonstrated by the higher slope of the curve for the HP/Convex machine shown in Fig. 10. However, due to the small amount of message passing in this type of multiple-passage parallel computation, the performance degradation on the low-cost personal computer cluster is almost negligible. This suggests potential wide application of such parallel computations on networked personal computers or workstations available in even a relatively small company.

In the calculation of the speedup shown in Fig. 10, the number of time steps done on the multiple-passage run is kept the same as that on the single-passage run. Examination of the computational unsteady solutions shows, however, that the multiple-passage computation reaches the final periodic solution significantly faster than the single-passage computation. Figure 11 plots time histories of the static pressure at two points on the periodic boundaries ahead of the blade leading edge. Computation on a single passage with the direct store method needs about 10 periods of computation before the computed periodic solution reaches a convergence level of $\delta = 0.001$, as defined by Eq. (17). The multiple-passage computation with the in-phase periodic boundary conditions at top and bottom boundaries needs less than three periods to reach the same convergence level, as shown by Fig. 11. On the HP/Convex SPP2000 machine, it takes 6.4 min for an Euler solution with 97×25 mesh to converge within 3 periods (each period has 64 real-time steps) with the parallel code using four passages, whereas the sequential code needed 18.6 min to get the same converged solution in 10 periods. This is an added benefit of the multiple-passage computation. The reason for this is that the flow quantities at the periodic boundaries used to update the solution in the direct store method are stored during the previous period of physical time, and thus, there is always a time lag before it converges to the final periodic solution. From this point of view, the parallel code not only scales up the problem linearly with the number of CPUs but also saves a significant amount of computational time due to the accelerated convergence.

After validating the parallel code against the single-passage computation, more computations are performed using the parallel multiple-passage N–S solver. For the standard configuration 4 test case, experimental results are available for different exit isentropic Mach number conditions. Figure 12 shows the computed results for isentropic exit Mach numbers $Mis = 0.58$, 0.90, and 1.19. Again, the computation shows general agreement with the trend of the experimental data but deviates in magnitude. It also can be seen that for $Mis = 1.19$ at IBPA = 270 deg, the experiment shows a positive damping. The computation, however, shows a negative damping. Figure 13 shows the variation of the computed damping coefficient with different exit Mach numbers $Mis = 0.58$, 0.76, 0.90, 1.02, and 1.19 at IBPA = 270 deg. It can be clearly seen that the computational result tends to follow the same trend as the experiment, but at $Mis = 1.19$ the experiment shows a stable motion, whereas the computation predicts an instability. Bolcs and Fransson[26] attribute this sudden change of stability at $Mis = 1.19$ to the appearance of a shock wave on the suction surface of the blade, which, however, is not observed in our computation.

A compressor test case is listed in Ref. 26 as standard configuration 7. In the experiment, the compressor blades were oscillated in

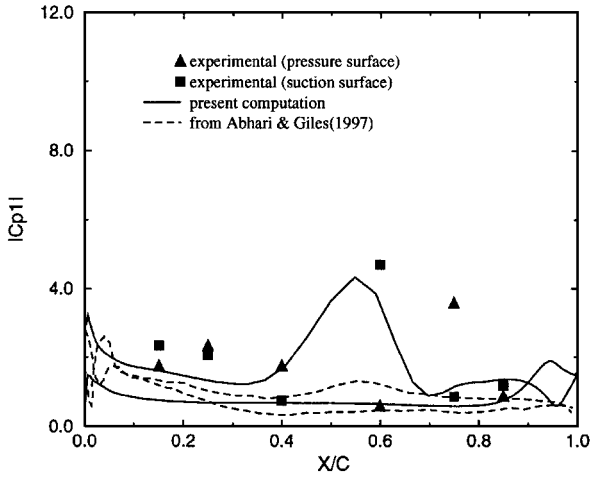

Fig. 14 Amplitude of the first harmonic unsteady-pressure coefficient over the blade with 180-deg IBPA.

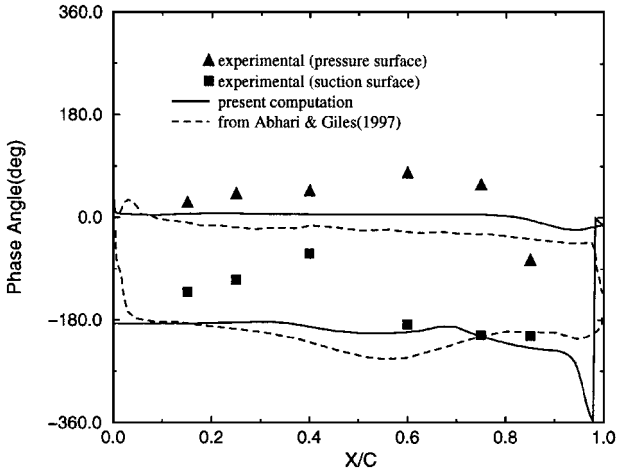

Fig. 15 Phase of the first harmonic unsteady-pressure coefficient over the blade with 180-deg IBPA.

a torsion-only mode under transonic/supersonic flow conditions. In the test case studied, the compressor was operated with an inlet Mach number of 1.315 and exit Mach number of 1.25. The computation is performed for a reduced frequency $\kappa = 0.44$, IBPA = 180 deg, and the amplitude of the torsion is 0.00122 rad. (Bending amplitude is zero.) Computation was also done by Abhari and Giles.[12] Figures 14 and 15 show the calculated amplitude and phase distributions of the first harmonic unsteady-pressure coefficient. The result from Abhari and Giles[12] by calculating a single passage with the direct store method is also plotted in Figs. 14 and 15. Figure 14 shows that the result by Abhari and Giles underpredicts the large suction peak on the suction surface of the blade. Abhari and Giles showed results that recovered the peak by simulating the complete five blades of the actual experimental cascade with each blade vibrating at the measured phase and magnitude, which varied significantly among the five blades. Although our computations were done on two passages, each blade was vibrated at the identical nominal frequency and magnitude. Contrary to the result by Abhari and Giles,[12] our computation suggests that the peak is resolved without simulating the differences of the vibration of the blades.

The preceding computational results show that our computations provide the same level of agreement with experiments as results by other authors. They also suggest much room for the future improvement for a CFD code to predict flutter accurately.

VII. Conclusions

A parallel fully implicit time-marching method for solving the N–S equations using dual time stepping and multigrid is used to calculate the aerodynamic damping of oscillating blades. The code is validated against experimental data and results by other authors. In its sequential mode, the code calculates the unsteady flow through one single blade passage by using the direct store method to enforce the phase-shifted periodic boundary condition. It is shown that a sequential code can be easily modified to solve multiple-passage problems parallel by assigning the computation of each blade passage to one CPU of a parallel computer and using MPI or PVM to do the message passing between the CPUs. Almost linear speedup is achieved with this parallelism based on the proposed time-stepping method. In addition, because the multiple-passage approach eliminates the need for the phase-shifted periodic boundary condition, significantly faster convergence to a steady periodic solution is obtained. Although results for flutter test cases with phase-shifted boundary conditions are presented, the strength of the method is the ability to study nonperiodic problems, for example, flutter with mistuned blades.

Acknowledgments

Support for this research was provided by the National Science Foundation under Grant CTS-9410800. Computer time was provided on the Hewlett Packard SPP2000 by the Office of Academic Computing of the University of California (UC) Irvine and the Aeneas parallel computer, also at UC Irvine. The authors would also like to thank Donald Dabdub and Herbert W. Hamber at UC Irvine for helpful discussions on parallel computing using the message passing interface.

References

[1]Carta, F. O., "Coupled Blade–Disc–Shroud Flutter Instabilities in Turbojet Engine Rotors," *Journal of Engineering for Power*, Vol. 89, No. 3, 1967, pp. 419–426.

[2]Smith, S. N., "Discrete Frequency Sound Generation in Axial Flow Turbomachines," Ph.D. Dissertation, Engineering Dept., Cambridge Univ., Cambridge, England, UK, 1971.

[3]Verdon, J. M., and Caspar, J. R., "A Linearized Unsteady Aerodynamic Analysis for Transonic Cascades," *Journal of Fluid Mechanics*, Vol. 149, Dec. 1984, pp. 403–429.

[4]Joubert, H., "Supersonic Flutter in Axial Flow Compressor," *Proceedings of Third Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines and Propellers*, Engineering Dept., Cambridge Univ., Cambridge, England, UK, 1984, pp. 231–254.

[5]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, 1985, pp. 945–952.

[6]Gerolymos, G. A., "Numerical Intergration of the 3D Unsteady Euler Equations for Flutter Analysis of Axial Flow Compressors," American Society of Mechanical Engineers, ASME Paper 88-GT-255, Amsterdam, June 1988.

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

[8]Kikuchi, O., Nakahashi, K., and Tamura, A., "Navier–Stokes Computations of Two- and Three-Dimensional Cascade Flowfields," *Journal of Propulsion and Power*, Vol. 5, No. 3, 1989, pp. 320–326.

[9]Scott, J. N., and Hankey, W. L., Jr., "Navier–Stokes Solutions of Unsteady Flow in a Compressor Rotor," *Journal of Turbomachinery*, Vol. 108, No. 2, 1986, pp. 206–215.

[10]Rai, M. M., and Madavan, N. K., "Multi-Airfoil Navier–Stokes Simulations of Turbine Rotor–Stator Interaction," *Journal of Turbomachinery*, Vol. 112, No. 3, 1990, pp. 377–384.

[11]Hwang, C. J., and Liu, J. L., "Analysis of Steady and Unsteady Turbine Cascade Flows by a Locally Implicit Hybrid Algorithm," *Journal of Turbomachinery*, Vol. 115, No. 4, 1993, pp. 699–706.

[12]Abhari, R. S., and Giles, M., "A Navier–Stokes Analysis of Airfoil in Oscillating Transonic Cascades for the Prediction of Aerodynamic Damping," *Journal of Turbomachinery*, Vol. 119, No. 1, 1997, pp. 77–84.

[13]He, L., and Denton, J. D., "Three-Dimensional Time-Marching Inviscid and Viscous Solutions for Unsteady Flows Around Vibrating Blades," *Journal of Turbomachinery*, Vol. 116, No. 3, 1994, pp. 469–476.

[14]He, L., "Time-Marching Calculations of Unsteady Flows, Blade Row Interaction and Flutter," *Unsteady Flows in Turbomachines*, Lecture Series 1996-05, von Kármán Inst. for Fluid Dynamics, Rhode Saint Genese, Belgium, 1996, pp. 28–75.

[15]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[16]Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, June 1991.

[17] Liu, F., and Ji, S., "Unsteady Flow Calculations with a Multigrid Navier–Stokes Method," *AIAA Journal*, Vol. 34, No. 10, 1996, pp. 2047–2053.

[18] Ji, S., and Liu, F., "Computation of Self-Excited Oscillation of Transonic Flow Around an Airfoil in a Channel," AIAA Paper 96-0063, Jan. 1996.

[19] Ji, S., and Liu, F., "Computation of Unsteady Flows Around Oscillating Blades and Aeroelasticity Behavior," AIAA Paper 97-0161, Jan. 1997.

[20] Erdos, J. I., and Alzner, E., "Numerical Solution of Periodic Transonic Flow Through a Fan Stage," *AIAA Journal*, Vol. 15, No. 11, 1977, pp. 1559–1568.

[21] Giles, M., "Stator/Rotor Interaction in a Transonic Turbine," *Journal of Propulsion and Power*, Vol. 6, No. 5, 1990, pp. 621–627.

[22] Wilcox, D. C., "Reassessment of the Scale-Determining Equation for Advanced Turbulence Models," *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

[23] Liu, F., and Zheng, X., "Staggered Finite Volume Scheme for Solving Cascade Flow with a k–ω Turbulence Model," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 991–998.

[24] Liu, F., and Zheng, X., "A Strongly-Coupled Time-Marching Method for Solving the Navier–Stokes and k–ω Turbulence Model Equations with Multigrid," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 289–300.

[25] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 7, 1981, pp. 357–372.

[26] Bölcs, A., and Fransson, T. H., "Aeroelasticity in Turbomachines—Comparison of Theoretical and Experimental Cascade Results," *Communication du Laboratoire de Thermique Appliquée et de Turbomachines*, No. 13, L'École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 1986.

[27] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Mancheck, R., and Sunderam, V., *PVM, Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA, 1994.

[28] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI, Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, MA, 1996.

S. K. Aggarwal
Associate Editor