In Proceedings of the 9th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines (ISUAAAT) and Legendre Lecture Series, Lyon, France, Sept. 4-8, 2000. Edited by Pascal Ferrand and Stephane Aubert, pp. 709-720. Presses Universitaires de Grenoble.

A THREE DIMENSIONAL NAVIER-STOKES CODE FOR AEROELASTICITY IN TURBOMACHINERY

I.W. MCBEAN
Department of Mechanical Engineering
Monash University
P.O. Box 31, Clayton, 3800, Australia

F. LIU
Department of Mechanical and Aerospace Engineering
University of California, Irvine
Irvine, CA 92697-3975

M.C. THOMPSON
Department of Mechanical Engineering
Monash University
P.O. Box 31, Clayton, 3800, Australia

Abstract. A multiblock method is presented for the solution of a three dimensional model of aeroelasticity in a turbomachine blade row. The method employs a fully coupled approach and the structural model involves modal reduction. Transfinite interpolation is used to adapt the fluid grid to the moving structure.

1. Introduction

As designers in the turbomachinery industry strive to design machines that are lighter, more powerful and more efficient, blade flutter has become one of the most important limiting factors on the design process. The aeroelastic response is a complex phenomenon that is not well modeled or predicted by current design techniques. Codes that implement 2-dimensional models can simulate this behaviour in a meridional plane, however the flow structures found in blade passages are generally three dimensional and such models provide a qualitative rather than quantitative analysis. Furthermore, im-

portant flow phenomena are not modeled including hub and casing vortices and tip effects.

A structured 3-dimensional Navier-Stokes code is developed to solve the unsteady governing equations. These are solved using an explicit Runge-Kutta scheme, implementing residual averaging and multigrid. The problem is then solved in a time accurate manner through a fully implicit scheme as proposed by Jameson [8]. This scheme has already been used in a 2-dimensional model of aeroelasticity in turbomachinery [17, 9]. The development of the present code is an extension of the previous 2-dimensional method to 3 dimensions. Similar algorithms have been successfully implemented in a 3-dimensional Navier-Stokes external solver that models flow over a flexible wing [18, 10].

2. Fluid Model

The present 3 dimensional multiblock and parallel code has been developed from a proven steady solver designed to model turbomachinery cascade flow [12, 11, 14, 19, 15, 16]. The governing equations for the unsteady fluid problem in a Eulerian reference frame with a moving mesh.

$$\frac{\partial}{\partial t}\iint_{\Omega} \mathbf{w}\, d\Omega + \oint \mathbf{f}\, dS_x + \mathbf{g}\, dS_y + \mathbf{h}\, dS_z = 0 \tag{1}$$

where

$$\mathbf{w} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \tag{2}$$

$$\mathbf{f} = \begin{pmatrix} \rho\bar{u} \\ \rho u\bar{u} + p \\ \rho v\bar{u} \\ \rho w\bar{u} \\ \rho E\bar{u} + pu \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho\bar{v} \\ \rho u\bar{v} \\ \rho v\bar{v} + p \\ \rho w\bar{v} \\ \rho E\bar{v} + pv \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} \rho\bar{w} \\ \rho u\bar{w} \\ \rho v\bar{w} \\ \rho w\bar{w} + p \\ \rho E\bar{w} + pw \end{pmatrix} \tag{3}$$

The time dependent and semi-discrete form of the governing equations may be written as

$$\frac{d\mathbf{w}}{dt} + R(\mathbf{w}) = 0 \tag{4}$$

A dual time stepping scheme [8] is used to calculate the unsteady flow problem. A second order accurate, fully implicit scheme is used to integrate Equation (4) to evolve the unsteady problem in a time accurate manner.

The discrete form of (4) is

$$\frac{3w^{n+1} - 4w^n + w^{n-1}}{2\Delta t} + R(w^{n+1}) = 0 \tag{5}$$

This equation may be recast into

$$\frac{d\mathbf{w}}{dt^*} + R^*(\mathbf{w}) = 0 \tag{6}$$

where

$$R^*(\mathbf{w}) = \frac{3w}{2\Delta t} + R(w) - \frac{2}{\Delta t}w^n + \frac{1}{2\Delta t}w^{n-1} \tag{7}$$

The steady state solution w in equation (6) is then equivalent to the time accurate solution w^{n+1} to equation (5). Any efficient algorithm may be used to obtain the steady-state solution to (6). In this paper, the above mentioned Runge-Kutta type scheme with multigrid is used. Minimum modification of the steady solver to make it time accurate in the above manner.

3. Structural Model

Modal decomposition, otherwise known as the Rayleigh Ritz approach reduces the structural problem to a series of uncoupled, second order differential equations. These are reduced to first order differential equations and solved by the same dual time stepping method as that for the flow equations. The problem is first solved in pseudo time using a Runge-Kutta scheme, then advanced it time through an implicit time accurate formulation [1, 2].

The generalized form of the structural equations are reduced through modal reduction through

$$\ddot{q}_i + 2\zeta_i\omega_i\dot{q}_i + \omega_i^2 q_i = Q_i \tag{8}$$

where q_i is the generalized normal mode displacement, ζ_i is the modal damping, ω_i is the modal frequency, and Q_i is the generalized aerodynamic force. The structural displacement vector is written as a summation of N modal shapes extracted from a finite element analysis of the structure.

$$u_s = \sum_{i=1}^{N} q_i h_i \tag{9}$$

where h_i are the modal shapes.

Equation (8) is further reduced to a first order system of equations for each i and integrated in time by a second-order fully implicit scheme.

Following Alonso and Jameson, we assume

$$\begin{aligned} x_{1i} &= q_i \\ \dot{x}_{1i} &= x_{2i} \\ \dot{x}_{2i} &= Q_i - 2\zeta_i\omega_i x_{2i} - \omega_i^2 x_{1i} \end{aligned} \tag{10}$$

for each of the modal equations. Thus in matrix form:

$$\{\dot{X}_i\} = [A_i]X_i + \{F_i\}, \quad i = 1, 2 \tag{11}$$

where $\{\dot{X}_i\} = \begin{Bmatrix} x_{1i} \\ x_{2i} \end{Bmatrix}$, $[A] = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\omega_i\zeta_i \end{bmatrix}$ and $\{F_i\} = \begin{Bmatrix} 0 \\ Q_i \end{Bmatrix}$. After proper diagonalization, the above equation can be decoupled.

$$\frac{dz_{1i}}{d\tau} = \omega_i(-\zeta_i + \sqrt{\zeta_i^2 - 1})z_{1i} + \frac{(-\zeta_i + \sqrt{\zeta_i^2 - 1})}{2\sqrt{\zeta_i^2 - 1}}Q_i \tag{12}$$

$$\frac{dz_{2i}}{d\tau} = \omega_i(-\zeta_i - \sqrt{\zeta_i^2 - 1})z_{2i} + \frac{(\zeta_i + \sqrt{\zeta_i^2 - 1})}{2\sqrt{\zeta_i^2 - 1}}Q_i \tag{13}$$

In these equations, the time derivative operator is discretized by a second order accurate scheme for each mode

$$\frac{3z_{1i}^{n+1} - 4z_{1i}^n + z_{1i}^{n-1}}{2\Delta\tau} = \omega_i(-\zeta_i + \sqrt{\zeta_i^2 - 1})z_{1i}^{n+1} + \frac{(-\zeta_i + \sqrt{\zeta_i^2 - 1})}{2\sqrt{\zeta_i^2 - 1}}Q_i^{n+1} \tag{14}$$

$$\frac{3z_{2i}^{n+1} - 4z_{2i}^n + z_{2i}^{n-1}}{2\Delta\tau} = \omega_i(-\zeta_i - \sqrt{\zeta_i^2 - 1})z_{2i}^{n+1} + \frac{(\zeta_i + \sqrt{\zeta_i^2 - 1})}{2\sqrt{\zeta_i^2 - 1}}Q_i^{n+1} \tag{15}$$

The above equations are coupled with the time-marching of the Navier-Stokes equations and solved by a dual time stepping algorithm similar to that used for the fluid solver.

4. Multiblock and Parallel Implementation

A method using multiple blocks of structured grids is used to maximize the use of computational resources and to allow the generation of grids for complex geometries. While each block consists of a structured grid, the blocks can be connected to each other in an unstructured manner provided the mesh geometry is matched at the block interfaces.

The method of code parallelization is not a trivial one as it adds another level to code complexity. There are many different standards and implementations that may be used, however the Message Passing Interface (MPI) has become one of the most popular. The majority of large, parallel computers support this standard, as do the more economical PC based Beowulf clusters. Open MP is another standard, that makes use of compiler directives to parallize loops.

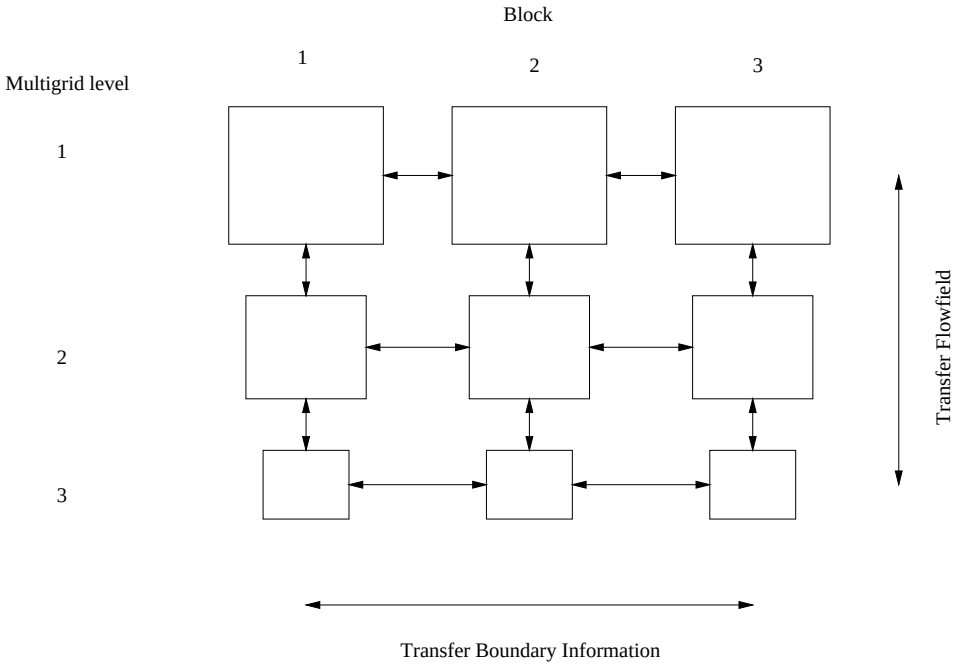

Figure 1. Schematic of multigrid and multiblock communication

Although Open MP may provide an easy method to implement a parallel code, its use is restricted to large shared memory symmetric multiple processor (SMP) machines. There are super computers that use shared memory that is only accessed by a limited number of nodes. Thus it is not guaranteed even on the large super computer that Open MP alone is adequate for efficient parallelization.

It was decided early in the development of the code to make use of MPI and some of the high level programming constructs available in Fortran90. A number of different objects were created in the code data structure to facilitate the parallel calculation of the fluid problem.

Each fluid block is treated as a single object or entity. A schematic of the multiblock data structure is shown in Figure 1. A processor may be allocated more than one fine grid block and each fine grid block will have associated a number of coarser, multiblock grids. The machine calculates for each multigrid level simultaneously, then copies the solution or interpolates the residual to the next multigrid level.

A subface is defined as another object. This is used for the interface between the present block and another block, or a region to which a single boundary condition is to be applied. These objects are cycled through each time the boundary of the blocks are to be updated, upon which com-

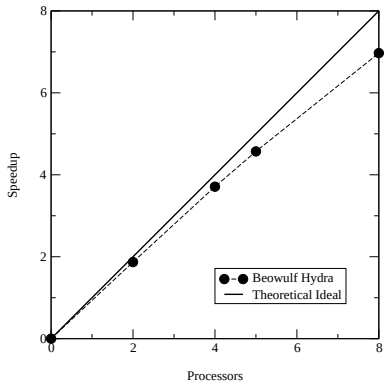

Figure 2. Speed Performance of multiblock code

munication is effected or a boundary condition is applied. So that separate boundary conditions are not required for each coordinate direction, subface objects are transformed into a single coordinate system. This increases code complexity marginally. However it maintains that the code that is relatively compact.

The structured cell numbering within each block is unimportant as transformations are used to reorient the face so that the numbering matches with the neighboring face. For example for a C-grid where a single block is wrapped around the blade, at the interface downstream of the trailing edge the cell numbering will be different on the upper and lower blocks, thus requiring reorientation of the 2-dimensional arrays.

In keeping with the use of high levels of Fortran90 code, the communication module also exploits some of the more sophisticated MPI routines. The use of MPI derived types allows the direct access of memory for the transfer of data, reducing the number of copies required during the communication of ghost cells. Due to the repetitive nature of the CFD computation, the "pipe-lining" of message passing calls is also implemented. The performance of the implementation was tested on a nine node Linux Beowulf cluster and the speed performance of the code is shown in Figure 2. A steady Euler calculation was used for the test using a mesh of 10240 cells.

5. The Moving Grid

The movement of the fluid boundary requires the fluid grid to be regenerated over the entire flow domain. Thus in the multiple block code, given that the grid for each structured block is regenerated independently, the position of the corner points of each block must be somehow defined. This is effected by using a spring network analogy as proposed by Batina [3] to

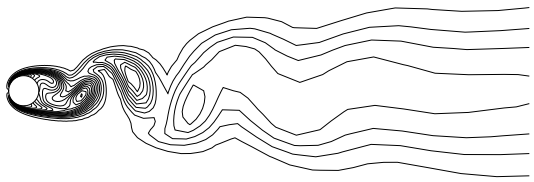

Figure 3. Plot of entropy contours for unsteady cylinder

maintain grid regularity. Grid regularity is particularly important where Navier-Stoke calculations are performed.

The network is formulated by connecting each block corner with hypothetical springs and corner positions are determined by a solution of the static equations. This simple and efficient calculation is performed on a single processor. Initially an unstructured grid network is constructed on the root processor. This contains nodal locations for each block corner and the connecting node information. New nodal positions are determined for free nodes through a predictor corrector scheme. These are distributed to the respective processors, where transfinite interpolation is performed to interpolate the local grid at the previous time step to the new position. The details of this method may be found in Wong [18].

6. Model Validation and Results

6.1. UNSTEADY CYLINDER

The low Reynolds number, unsteady cylinder is a well documented case in both experimental and numerical fields. In this case it was used to check the time accuracy of the unsteady implementation. A single block O-grid was generated with the farfield boundary approximately 50 chords from the cylinder surface. The code may only calculate for 3-dimensional mesh geometries, so 2 cells were used in the spanwise direction. The Mach number of the compressible solver was set at 0.2 as at this Mach number, the effects of compressibility are assumed to be negligible. The Strouhal number for a grid of 196 x 96 x 3 was calculated as 0.181, which is close to the experimental value of 0.182. A grid resolution study was performed, the results of which are beyond the focus of this paper. The shedding behind the cylinder may be observed in Figure 3.

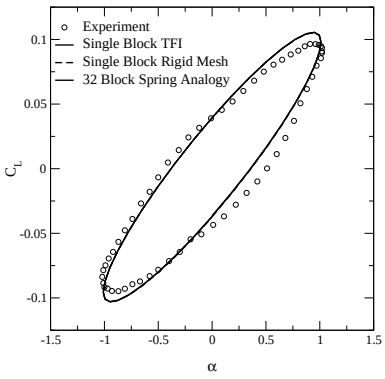

Figure 4. Comparison of unsteady NACA64A010 results with experiment

6.2. FORCED AIRFOIL OSCILLATION

To demonstrate the validity of the moving mesh, multiblock and unsteady implementation, the NACA64A010 case is presented. Computational results are compared in Figure 4 for different configurations and with experiment. An unsteady Euler calculation is performed in the flow solver. In the first case, a single block O-grid is used in combination with TFI to deform the grid to the oscillating airfoil. The far field boundary remains rigid. The second case involves a mesh that is not deformed, but rotates rigidly with the displacement of the airfoil surface. For the third calculation, the same grid as used for the single block cases is divided into 32 equal blocks, with 4 blocks in the radial direction and 8 in the circumferential direction. In this case the block corners were located using the spring analogy. The results for inviscid flow compare similarly with results presented elsewhere [13, 1] and there is little difference between the results for the different configurations.

6.3. COUPLED AIRFOIL OSCILLATION

The modeling of aeroelasticity requires the simulation of the interaction of elastic member with an unsteady flow. One of the simplest examples is Isogai's wing model [6, 7], a 2-dimensional NACA64A010 airfoil that has been studied numerically by a number of authors [1, 10]. Experimental unsteady flow measurements are available for the airfoil for forced oscillation and these results may be used to validate the unsteady flow model.

A qualitative study was performed for the neutrally stable, unstable and damped configurations and results are shown in Figures 5 to 7. These results are similar to those found by other authors [10], providing confidence in the implementation of the structural model and the numerical algorithm

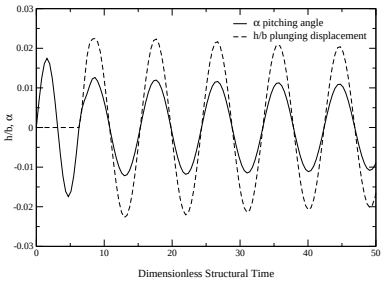

Figure 5. Stable case; $M_\infty = 0.825$, $V_f = 0.630$

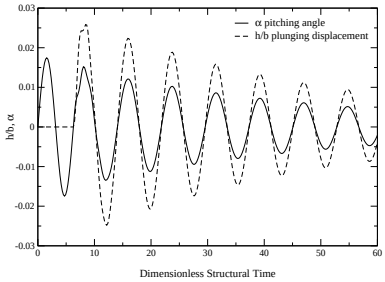

Figure 6. Damped case; $M_\infty = 0.825$, $V_f = 0.530$

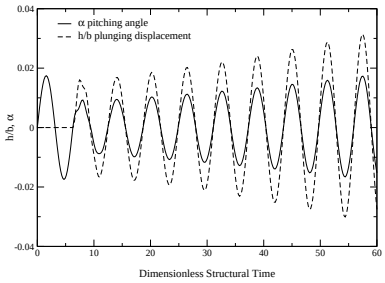

Figure 7. Diverging case; $M_\infty = 0.825$, $V_f = 0.725$

for coupling the fluid flow and structural dynamics solutions.

6.4. UNSTEADY CASCADE RESULTS

Simulations of forced oscillation in a turbomachinery cascade can provide useful insights into the influence of the flow on the aeroelastic stability of the turbine blade. A study was undertaken by Bell and He [5, 4] into the linearity of the unsteady flow features for a turbine blade. This involved a blade mounted in a single passage with the upper and lower tunnel surfaces contoured to the blade shape and inlet and outlet flow angles. The blade was pivoted about its root and forced to oscillate rigidly in the bending mode.

The surface pressure coefficient was obtained for bending amplitudes at the blade tip of 5.5 % and 2.75 % of chord, at a reduced frequency of 0.5. Due to the fact that the fluid solver implements the compressible form of the Navier-Stokes equations, the simulations were at a higher Mach number of 0.2 compared with 0.1 in the experiment. Although Mach number similarity was not maintained, the velocity or time scale was preserved by ensuring a constant reduced frequency.

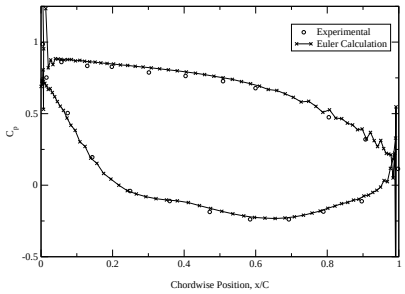

Figure 8. Cascade steady solution compared with experiment

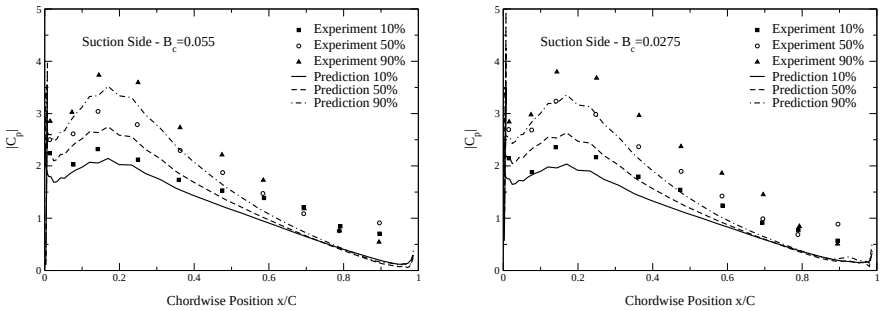

Figure 9. Comparison of magnitude of first harmonic of unsteady pressure coefficient with experiment for Suction Surface, $K_c = 0.5$

Figure 8 shows the pressure coefficient distribution on the blade surface when the blade is stationary. Except for some oscillations near the leading and trailing edges, the numerical result compares well with the experimental data. Figures 9 and 10 shows the magnitude of the first harmonic pressure distribution on the blade suction and pressure surfaces and 10%, 50% and 90% spanwise locations. Although the computational results show the same trend as the experimental data, the computed magnitude is lower.

Figure 11 shows the phase angle distribution of the first harmonic unsteady pressure coefficient. The computational result agrees well with the experimental data except near the trailing edge. The results for the first harmonic pressure coefficient are extremely close between the two different amplitudes of oscillation. This confirms the findings by Bell and He that the flow is essentially linear for these conditions.

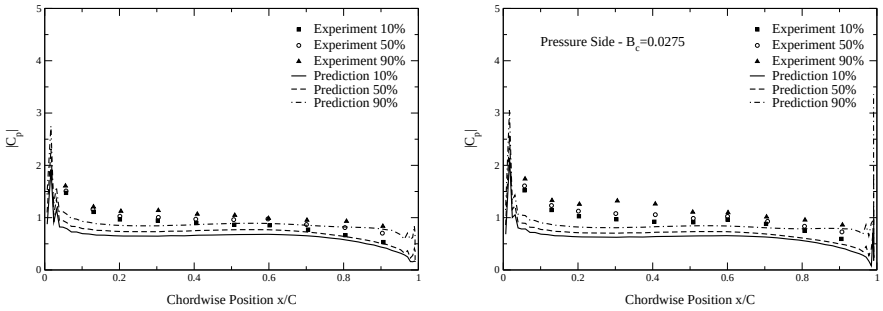

Figure 10. Comparison of magnitude of first harmonic of unsteady pressure coefficient with experiment for Pressure Surface, $K_c = 0.5$

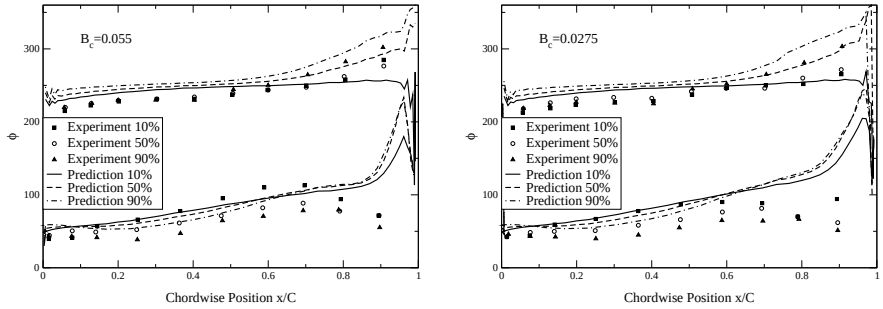

Figure 11. Comparison of phase of first harmonic of unsteady pressure coefficient with experiment, $K_c = 0.5$

7. Concluding Remarks

A novel multiblock and parallel, integrated structural and fluid solver has been presented. The implementation is general and is not limited to particular geometries and thus is flexible in that it may be applied to a broad range of problems. The moving mesh and structural model allow for the coupled solution of aeroelastic problems.

A number of different cases have been presented that compare computed results with experiment or other numerical results. Navier-Stokes solution of the flow past a circular cylinder compares well with experimental data. The code is also validated for the unsteady flow around a pitching airfoil with either a rigid grid or a deforming grid generated by a multiblock TFI method. Coupled flow and structure solution for an airfoil with two-degrees of freedom demonstrate the ability of the code to simulate damped, neutral and diverging motions of the system. Finally, the 3-dimensional flow through a turbine model with a blade performing bending motion is calculated. The computed first harmonic pressure coefficient on the blade

agrees well with experimental data in phase but is smaller in magnitude.

Work is needed to further validate the code for the coupled simulation of three-dimensional fluid-structure systems.

References

1. Alonso, Juan, & Jameson, Antony. 1994 (Jan.). Fully-implicit time-marching aeroelastic solutions. *In: AIAA 32nd Aerospace Sciences Meeting*. AIAA. 94-0056.
2. Alonso, Juan Jose. 1997 (June). *Parallel computation of unsteady and aeroelastic flows using an implicit multigrid-driven algorithm*. Ph.D. thesis, Pinceton University.
3. Batina, J. T. 1990. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, **28**(8), 1381–1388.
4. Bell, D. L., & He., L. 1997. Three dimensional unsteady pressure measurements for an oscillating turbine blade. *In: International gas turbine & aeroengine congress & exhibition*. ASME. 97-GT-105.
5. Bell, D. L., & He, L. 1998. Three dimensional unsteady flow around a turbine oscillating in bending mode = an experimental and computational study. *Pages 53–66 of:* Fransson, T. (ed), *Unsteady Aerodynamics and Aeroelasticity of Turbomachines*. Kluwer Academic Publishers.
6. Isogai, K. 1979. On the transonic-dip mechanism of flutter of a sweptback wing. *AIAA Journal*, **17**(7), 793–795.
7. Isogai, K. 1981. On the transonic-dip mechanism of flutter of a sweptback wing: Part ii. *AIAA Journal*, **19**(7), 1240–1242.
8. Jameson, Antony. 1991 (June). Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *In: AIAA 10th computational fluid dynamics conference*. AIAA.
9. Ji, Shanhong, & Liu, Feng. 1999. Flutter computation of turbomachinery cascades using a parallel unsteady Navier-Stokes code. *AIAA Journal*, **37**(3), 320–327.
10. Liu, F., Cai, J., Zhu, Y., & A. S. F. Wong, H. M. Tsai. 2000 (Jan.). Calculation of wing flutter by a coupled CFD-CSD method. *In: AIAA 38th Aerospace Sciences Meeting & Exhibit*. AIAA. AIAA 2000-0907.
11. Liu, Feng, & Jameson, Antony. 1993a. Cascade flow calculations by a multigrid Euler method. *Journal of Propulsion and Power*, **9**(1), 90–97.
12. Liu, Feng, & Jameson, Antony. 1993b. Multigrid Navier-Stokes calculations for three-dimensional cascades. *AIAA Journal*, **31**(10), 1785–1791.
13. Liu, Feng, & Ji, Shanhong. 1996. Unsteady flow calculations with a multigrid navier-stokes method. *AIAA Journal*, **34**(10), 2047–2053.
14. Liu, Feng, & Zheng, Xiaoqing. 1994. Staggered finite volume scheme for solving cascade flow with a k-ω turbulence model. *AIAA Journal*, **32**(8), 1589–1597.
15. Liu, Feng, & Zheng, Xiaoquing. 1996. A strongly coupled time-marching method for solving the Navier- Stokes and k-ω turbulence model equations with multigrid. *Journal of Computational Physics*, **128**, 289–300.
16. Liu, Feng, Jennions, Ian K., & Jameson, Antony. 1998 (Jan.). Computation of turbomachinery flow by a convective-upwind-split- pressure (CUSP) scheme. *In: 36th Aerospace Sciences Meeting and Exhibit*. AIAA.
17. Sadeghi, M., & Liu, F. 2000 (Jan.). Computation of mistuning effects on cascade flutter. *In: AIAA 38th Aerospace Sciences Meeting & Exhibit*. AIAA. AIAA 2000-0230.
18. Wong, A. S. F., Tsai, H. M., Cai, J., Zhu, Y., & Liu, F. 2000 (Jan.). Unsteady flow calculations with a multi-block moving mesh algorithm. *In: AIAA 38th Aerospace Sciences Meeting & Exhibit*. AIAA. To be Presented, AIAA 2000-1002.
19. Zheng, Xiaoqing, & Liu, Feng. 1995. Staggered finite volume scheme for solving Navier-Sokes and k-ω turbulence model equations. *AIAA Journal*, **33**(6), 991–998.