Parallel Unsteady Simulation of the Flow Through a Helicopter Rotor in Hover Including Aeroelastic Effects

by

J.J. Alonso, S.G. Sheffer, L. Martinelli and A. Jameson

Department of Mechanical and Aerospace Engineering
Princeton University, Princeton, New Jersey, USA

Proceedings
First AFOSR Conference on Dynamic Motion CFD

Rutgers University, New Brunswick
June 1996

Parallel Unsteady Simulation of the Flow Through a Helicopter Rotor in Hover Including Aeroelastic Effects

Juan J. Alonso[+], *Scott G. Sheffer*[*], *Luigi Martinelli*[†], *and Antony Jameson*[‡]
CFD Laboratory for Engineering Analysis and Design
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey 08544 U.S.A.

First AFOSR Conference on Dynamic Motion CFD
Rutgers University, New Brunswick, NJ, June 3-5, 1996

1 Abstract

The following work presents preliminary developments in the simulation of aeroelastic helicopter rotor flows in the hover regime. Simulations are carried out by solving the full time-accurate compressible Euler equations on a moving multiblock mesh around a complete rotor configuration. Aeroelastic effects are introduced as an integral part of the simulation. The resulting program, ROTOR87, is implemented in parallel, using a domain decomposition approach and the MPI (Message Passing Interface) Standard for communication purposes. Comparisons of the solutions with experimental data are presented for two-bladed and five-bladed helicopter rotors in hover. The results demonstrate that combining highly efficient algorithms with high performance parallel computing yields an accurate and efficient method for the computation of complex vortical flows of interest to the rotorcraft community.

2 Introduction

During the course of the last three years, much effort has been placed at Princeton on the development of accurate and efficient methods for the calculation of unsteady viscous and inviscid flows including aeroelastic effects. Efficiency has been achieved through the usage of fast implicit algorithms and the utilization of high performance parallel computing platforms. The pursuit of high accuracy has focused on the implementation of refined artificial dissipation algorithms which provide the necessary upwind bias without unnecessarily corrupting the flow solution and the use of properly resolved meshes for the physical phenomena at hand. It is our opinion that a successful tool for the computation of flows of relevance to the rotorcraft community must be composed of parts of both of these ingredients.

The accurate computation of helicopter rotor flows in both hover and forward flight is a particularly challenging problem due to the inherent difficulties that it entails. Two aspects of these computations stand out as being especially complex. On one hand, reliable prediction of helicopter hover and forward flight performance is heavily dependent on the proper resolution of the blade/vortex interaction that occurs near the tip region. This interaction has a strong influence on the inflow angles and pressure distributions of the outboard sections of a blade. On the other hand, the establishment of a full rotor wake is a problem of inherent stiffness due to the varying scales present in the problem: while it is necessary to accurately resolve the turning motion of the blade, a large number of revolutions is required for the establishment of a steady wake pattern.

Furthermore, the simulation of the forward flight problem has not been thoroughly attempted yet because of the phenomenal computational requirements necessary to complete the work. Typical CFD computations of two-bladed helicopter rotors in forward flight use a pseudo-steady formulation and ad-hoc modeling of the wake on the half of the rotor in which the calculations are not carried out [9]. This paper will show that a full multigrid-implicit approach to the solution of the unsteady Euler equations for helicopter flows is now computationally feasible using a parallel implementation of the method, and an algorithm where the time-step of the computation is solely dictated by accuracy requirements, and not by numerical stability restrictions. Using this method, the flowfield of a helicopter rotor in hover is easily calculated with a moving mesh strategy without the need to resort to a quasi-steady formulation. With this in mind, the solution of the forward flight problem only requires an additional factor in computing time equal to the number of blades in the rotor, since periodic

+ Graduate Student
* Graduate Student
† Assistant Professor
‡ James S. McDonnell Distinguished University Professor of Aerospace Engineering

boundary conditions can no longer be used. In addition, we also present a steady-state approach to the calculation of the hover problem, that yields faster turnaround for this type of calculations. Both the fully unsteady and the steady-state formulations yield the exact same results upon convergence, but only the fully unsteady approach is applicable to the computation of the forward flight regime.

The results presented in this paper are computed using a fully-implicit discretization of the Euler equations. At every time-step, the inversion of the implicit equations is achieved with the aid of a pseudo-time inner iteration which takes advantage of convergence acceleration techniques such as multigrid and residual averaging [1, 2, 3, 7]. This implicit discretization allows the time-step to be based on accuracy requirements, and not on numerical stability issues: for isolated inviscid wing calculations, CFL numbers on the order of $3,000 - 5,000$ are typical. For viscous calculations on meshes with wide ranges in the sizes of the cells in the domain (for instance, for appropriate resolution of the boundary layer and blade tip vortex regions), the CFL number can easily reach 50,000 and would make the use of an explicit time integration completely impracticable.

The rotor blades are allowed to deform aeroelastically forced by the instantaneous aerodynamic load around the blade. The aeroelastic solution is performed using a truncated modal decomposition approach of the finite element equations of motion of the structure, and the mode shapes and frequencies are provided by a finite element solver based on 16 degrees-of-freedom plate elements which are appropriate for this kind of calculation. The aeroelastic equations are implicitly coupled to the flow solver solution leading to a high degree of fidelity in the simulation, even when large time-steps are taken. Details of the aeroelastic coupling to the flow solver can be found in [1].

The flow solver uses a multiblock mesh configuration to allow for future high resolution of more complex blade tip shapes and the inclusion of full helicopter geometries (rotor hub and fuselage). Initially, these multiblock meshes are generated by the decomposition of an O-H mesh, but in the future, they will be constructed using traditional elliptic multiblock mesh generation techniques.

Finally, the complete rotor solution (aeroelastic effects included) is implemented for distributed memory architectures using a static domain decomposition approach and the MPI (Message Passing Interface) Standard for communication purposes. The heart of the computational algorithm is of an explicit nature, and therefore, high parallel efficiencies can be achieved for this type of implementation [8]. This paper represents our first step toward complete unsteady simulation of helicopter aeroelastic phenomena in forward flight.

3 Governing Equations and Discretization

Consider a control volume $\mathcal{V}$ with boundary $\partial\mathcal{V}$ in a Cartesian coordinate system. The control volume moves with velocity $\mathbf{b} = (x_t, y_t, z_t)$ while the fluid velocity is $\mathbf{u} = (u, v, w)$. In this coordinate system, the three-dimensional unsteady compressible Euler equations can be written as:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0 \tag{1}$$

where $\mathbf{w}$ is the vector of conserved flow variables

$$\mathbf{w} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix}$$

and $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ are the convective flux vectors

$$\mathbf{f} = \begin{Bmatrix} \rho(u - x_t) \\ \rho u(u - x_t) + p \\ \rho v(u - x_t) \\ \rho w(u - x_t) \\ \rho E(u - x_t) + pu \end{Bmatrix}, \mathbf{g} = \begin{Bmatrix} \rho(v - y_t) \\ \rho u(v - y_t) \\ \rho v(v - y_t) + p \\ \rho w(v - y_t) \\ \rho E(v - y_t) + pv \end{Bmatrix}$$

$$\mathbf{h} = \begin{Bmatrix} \rho(w - z_t) \\ \rho u(w - z_t) \\ \rho v(w - z_t) \\ \rho w(w - z_t) + p \\ \rho E(w - z_t) + pw \end{Bmatrix}$$

in each of the coordinate directions. The equations of motion of the fluid can then be written in integral form as

$$\frac{d}{dt} \iiint_{\mathcal{V}} \mathbf{w}\, d\mathcal{V} + \iint_{\partial\mathcal{V}} \mathbf{w}(\mathbf{u} - \mathbf{b}) \cdot \mathbf{n}\, dA = \iint_{\partial\mathcal{V}} \mathbf{S}\, dA, \tag{2}$$

where the right hand side vector $\mathbf{S}$ is given by

$$\mathbf{S} = \begin{Bmatrix} 0 \\ -pn_x \\ -pn_y \\ -pn_z \\ -p(\mathbf{u} \cdot \mathbf{n}) \end{Bmatrix}.$$

Also, for an ideal gas, the equation of state may be written as

$$p = (\gamma - 1)\, \rho \left[E - \frac{1}{2}(u^2 + v^2 + w^2) \right].$$

When the integral governing equations (2) are independently applied to each cell in the domain, we obtain a set of coupled ordinary differential equations of the form

$$\frac{d}{dt}(\mathbf{w}_{ijk}\, V_{ijk}) + \mathbf{E}(\mathbf{w}_{ijk}) + \mathbf{D}(\mathbf{w}_{ijk}) = \mathbf{0}, \tag{3}$$

where $\mathbf{E}(\mathbf{w}_{ijk})$ are the convective Euler fluxes and $\mathbf{D}(\mathbf{w}_{ijk})$ are the artificial dissipation fluxes added for numerical stability reasons. This equation (3) can be discretized implicitly as follows (drop the i, j, k subscripts

for clarity):

$$\frac{d}{dt}[\mathbf{w}^{n+1}\, V^{n+1}] + \mathbf{R}(\mathbf{w}^{n+1}) = \mathbf{0}, \tag{4}$$

where $\mathbf{R}$ is the sum of the two flux contributions, and the superscripts denote the time step of the calculation. If we discretize the time derivative term with, say, a backwards difference second order accurate operator, we obtain

$$\frac{3}{2\Delta t}[\mathbf{w}^{n+1}V^{n+1}] - \frac{2}{\Delta t}[\mathbf{w}^{n}V^{n}] + \frac{1}{2\Delta t}[\mathbf{w}^{n-1}V^{n-1}] + \mathbf{R}(\mathbf{w}^{n+1}) = \mathbf{0}.$$

The time integration of the Euler equations at each time step can then be seen as a modified pseudo-time steady-state problem with a slightly altered residual

$$\mathbf{R}^*(\mathbf{w}) = \frac{3}{2\Delta t}[\mathbf{w}\, V^{n+1}] - \frac{2}{\Delta t}[\mathbf{w}^{n}V^{n}] + \frac{1}{2\Delta t}[\mathbf{w}^{n-1}V^{n-1}] + \mathbf{R}(\mathbf{w}),$$

In this case, the vector of flow variables $\mathbf{w}$ which satisfies the equation $\mathbf{R}^*(\mathbf{w}) = \mathbf{0}$ is the $\mathbf{w}^{(n+1)}$ vector we are looking for. In order to obtain this solution vector, we can reformulate the problem at *each* time step as the following modified steady-state problem in a fictitious time, t^*:

$$\frac{d\mathbf{w}}{dt^*} + \mathbf{R}^*(\mathbf{w}) = \mathbf{0}, \tag{5}$$

to which one can apply the fast convergence techniques used for steady-state calculations. Applying this process repeatedly one can advance the flow field solution forward in time in a very efficient fashion.

If we wished to solve for the absolute velocities in hover without physically rotating the computational grid, we may follow the formulation of Holmes and Tong [6]. This involves adding source terms to equation 2 so that our governing equation becomes:

$$\iiint_{\mathcal{V}} \frac{\partial \mathbf{W}}{\partial t}\, d\mathcal{V} + \iint_{\partial\mathcal{V}} \mathbf{w}(\mathbf{u}-\mathbf{b})\cdot\mathbf{n}\, dA = \iint_{\partial\mathcal{V}} \mathbf{S}\, dA + \iiint_{\mathcal{V}} \mathbf{T}\, d\mathcal{V}. \tag{6}$$

The grid velocity vector is $\mathbf{b} = (-\Omega z, 0, \Omega x)$, while the additional term on the right hand side of equation 6 is introduced to account for the actual motion of the grid and has the form

$$\mathbf{T} = \begin{Bmatrix} 0 \\ \rho\Omega w \\ 0 \\ -\rho\Omega u \\ 0 \end{Bmatrix}$$

for a rotor which lies in the x, z plane. With this formulation, the problem at hand can now be solved as a steady-state hover problem.

4 Parallel Multiblock Flow Solver: ROTOR87

4.1 Solver Algorithm

The essential algorithm used in the flow solution for the multiblock flow solver is similar to the one used in the original version of the code which used a single block mesh [7]. A cell-centered discretization of the governing flow equations is used, and the time derivative operator in equation 4 is discretized with a second-order accurate backwards difference formula. The Euler and artificial dissipation fluxes are lumped with the discretization of the time derivative operator, and a new *modified* residual is thus formed. A modified 5-stage Runge-Kutta time-stepping scheme is used to drive this modified residual to an acceptable level of convergence. When this is achieved, the solution corresponds to the flow field variables at the new time-step (for the new mesh location).

The only difference with the solution strategy adopted for the original solver reported in [7] resides in the fact that an additional outer loop over all the blocks in the domain is added. The internal structure of the flow solver is, however, completely different, since it is optimized for large loops over all the cells in the domain, regardless of the number of blocks present. The solver is now split into a pre-processing step that sets up the required data structures and array sizes, and the actual flow solver itself.

4.2 Parallelization Strategy

The original single block solver was parallelized using a static domain decomposition along the three coordinate directions of the block. The parallelization strategy for the multiblock solver, however, is quite different. Similarly to the single block solver (from now on referred to as UFLO87), ROTOR87 is parallelized using a domain decomposition model, a SPMD (Single Program Multiple Data) strategy, and the MPI Library for message passing. The partitioning of the mesh could be performed in the same fashion as in UFLO87 for each and every one of the blocks in the multiblock mesh. Since the sizes of the blocks can be quite small sometimes, further partitioning would severely limit the number of multigrid levels that could be used in the flow solution, and thus would hurt convergence. For this reason, it was decided to use a domain decomposition strategy that allocated complete blocks to a given processor.

The underlying assumption is the fact that there are always more blocks than processors available. If this is the case, every processor in the domain would be responsible for the computations inside one or more complete blocks. In the case in which there are more processors than blocks available, the blocks can be adequately partitioned during a pre-processing step in order to have at least as many blocks as processors. This approach has the advantage that the number of multigrid levels that can be used in the parallel implementation of the

code is always the same as in the serial version. Moreover, the number of processors in the calculation can be any integer number, since no restrictions are imposed by the partitioning in all coordinate directions used by the single block program. In sum, each processor runs a copy of a multiblock flow solver (although typically only with a small number of blocks), and the various processors communicate at different stages of the calculation in order to send/receive the necessary information for all fluxes to be computed adequately.

The only drawback of this approach is the loss of the exact load balancing that one had in UFLO87. The blocks in the calculation can have different sizes, and consequently, it is very likely that different processors will be assigned a different total number of cells. This, in turn, will imply that some of the processors will be waiting until the processor with the largest number of cells has completed its work and parallel performance will suffer. The approach that we have followed to solve the load balancing problem is to assign to each processor, in a pre-processing step, a certain number of blocks such that its total number of cells is as close as possible to the exact share for perfect load balancing. An algorithm is used that distributes the blocks trying to minimize the maximum number of cells in all processors. Although this algorithm is not assured to obtain a truly optimum distribution of the blocks, it has been found in practice to yield quite satisfactory results [8].

Within each processor there will be several blocks that need to communicate with their neighboring blocks. The data for these neighboring blocks can reside in a different processor, and therefore, communication is necessary. In order to minimize communication cost, it was decided to pack all data that needed to be sent from one processor to another in one single message, regardless of the number of blocks that resided in each of the processors. Within each processor, the data for the flow variables and grid locations is stored in a large one-dimensional array. In order to accomplish this type of communication, during the pre-processing step, each processor compiles a pointer list with all the entries in these large arrays that need to be sent to all other processors. Similarly, another pointer list for the locations of the data to be received is also set up. At the time of information exchanges, each processor communicates all the necessary data for the blocks that it contains to those processors that need to receive it. The communication is implemented using the asynchronous (non-blocking) send and receive MPI constructs in order to be able to perform some useful work while the information is being transferred.

4.3 Boundary Conditions

Four different types of boundary conditions are imposed on the faces of the blocks in the mesh. All block faces which lie directly on the surface of the blade use a flow tangency boundary condition. For moving meshes, the velocity of the mesh cells must be taken into account in order to properly implement these kind of boundary conditions.

At the inflow boundary, and on the far-field side walls of the mesh, non-reflecting boundary conditions based on one-dimensional Riemann invariants normal to the boundary are imposed. At the outflow boundary, a direct extrapolation of the flow quantities is performed, as reported elsewhere in the literature [13, 10, 12, 14].

Finally, for hover calculations which only use one blade sector, periodicity boundary conditions are used to transmit the data from the vertical inflow plane to the outflow and vice versa.

Notice that no attempt is made to artificially correct the boundary conditions in order to match the experimental thrust coefficients. Every effort is made to achieve boundary conditions that resemble the conditions of a rotor in free hover.

4.4 Mesh Movement

For rigid rotor blade calculations, the mesh is only required to rotate about the vertical axis in a solid body fashion. Therefore, a simple rotational transformation is applied to every point in each of the blocks in the mesh.

For calculations in which the rotor blades are allowed to deform aeroelastically, a procedure for moving the mesh points within each block must be determined. In general, the original mesh for the undeformed rotor blade is generated using an elliptic or hyperbolic mesh generator. During the solution process, the blades deform due to the unsteady airloads, and the mesh must be moved to conform at all times with the instantaneous position of the surface of the blade.

Clearly, the process of mesh generation is a highly interactive and time consuming process, and thus cannot be embedded in the calculation process. Since the mesh deflections are typically small, an automatic procedure to achieve mesh deformations was pursued.

Reuther et al. [11] have used a procedure called WARP3D for the deformation of multiblock meshes used in automatic aerodynamic design calculations. In this case, the blocks on the surface of the wing must be deformed due to the effect of the changing values of the design variables in the optimization problem. The mesh motion requirements for the aeroelastic rotor simulation are perfectly addressed by this mesh motion strategy, and thus, WARP3D was used here as well. In a sense, the mesh deflections in an unsteady aeroelastic simulation can be viewed as deformations caused by design variables which correspond to the modal coordinates of the different modes of vibration of the structure.

WARP3D uses an algorithm which is quite similar to transfinite interpolation (TFI). Unlike TFI, where there is no prior knowledge of the interior mesh, WARP3D makes use of the relative interior point distributions in the initial mesh. The algorithm allows the perturbation of all the points in a given block by specifying the final location of the faces that move during the simulation process. The reader is referred to [11] for more details.

5 Structural Equations and Coupling

The structural equations are obtained from a finite element model and generally take the form

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{F\}, \qquad (7)$$

where $[M]$, $[C]$, and $[K]$ are $n \times n$ mass, damping, and stiffness matrices for an n-dof structure. The solution is obtained using a modal decomposition approach in which only the first N normal vibration modes are considered so that the truncated model becomes

$$\ddot{\eta}_i + 2\zeta_i\omega_i\dot{\eta}_i + \omega_i^2\eta_i = f_i, \quad i = 1, \ldots, N \qquad (8)$$

where η_i is the i-th normal coordinate, ω_i is the natural frequency of the i-th mode, ζ_i is the modal damping constant and f_i is the corresponding forcing term.

For true unsteady calculations such as the ones required for forward flight simulations, these equations are decomposed into a first order system, discretized using second or third-order accurate backward differencing and then marched to a steady state in pseudo-time as described in [1].

The structural equations are coupled to the flow solution through the forcing terms f_i which reflect the instantaneous pressure distribution on the surface of the blade. Information is exchanged between the fluid and structural solvers at several points within the pseudo-time iteration so that the blade position and velocity are consistent with the pressure distribution when full convergence is achieved.

In the hover case, the same approach can be followed if a time-accurate formulation with a moving mesh is used. If instead we solve for the steady-state solution, carrying the additional information from the time histories of the normal coordinates and the mass matrix effects leads to slower aeroelastic convergence. This time to convergence depends heavily on the atmospheric conditions and the true stability characteristics of the blade. In order to obtain faster aeroelastic convergence, the time dependent terms in equation 8 can be dropped in this formulation. After the forcing terms for all normal modes of vibration are calculated, the deflected position of the blade can simply be obtained from:

$$\eta_i = \frac{f_i}{\omega_i^2}, \quad i = 1, \ldots, N \qquad (9)$$

Further acceleration to aeroelastic convergence can usually be obtained by employing an overrelaxation approach to the evolution of each of the normal coordinates, in a similar fashion to the work of Borland [4]. This method has not been yet tried in our work, but will be used in the near future.

The structural model for the present work employs 16-dof plate finite elements but the structural information could be provided by other more elaborate finite element models since the solver only relies on a description of the normal modes.

Mention should be made of the fact that additional data structures needed to be set up in the case of a multiblock flow solver, since different portions of the blade now reside on arbitrary blocks in the mesh, which, in turn, might even reside in different processors. This issue is taken care of in a preprocessing step by distributing the complete mode shapes to all the processors, and maintaining the proper masks in all blocks that allow their cells to know which global structural cell number they correspond to.

6 Results

This section presents some preliminary results of Euler calculations for helicopter rotors in hover. The rotors in this work were simulated with both the full unsteady formulation utilizing a moving grid and the quasi-steady hover formulation. Both results were essentially identical. Because the quasi-steady formulation consumed less time, most of the results in this paper were computed using the quasi-steady code.

6.1 Rigid Rotor Configurations

Three sets of calculations were performed on a $128 \times 32 \times 48$ cell mesh modelling an untwisted, untapered two-bladed NACA 0012 rotor with an aspect ratio of 6. Experimental results for this rotor at varying collective pitch angles and rotational speeds have been obtained by Caradonna and Tung [5]. The first case considered is a collective pitch of 0 degrees and a tip Mach number of 0.520. This case is a good test of the flow solver in the absence of downwash effects (thus removing possible reflections from the boundaries). Figure 1 shows computational and experimental pressure coefficient distributions at three spanwise locations in the outer portion of the blade. The computational results are in excellent agreement with the experimental results.

The second case has a collective pitch of 8 degrees and a tip Mach number of 0.439. Figure 2 shows the pressure coefficient distribution at the same spanwise locations, again indicating excellent agreement with the experimental data. The final case is also at a collective pitch of 8 degrees but has a higher tip Mach number of 0.877, which produces a region of supersonic flow over the blade. Figure 3 shows the pressure coefficient distribution at the same three near-tip locations. Note that the bottom surface and post-shock regions are captured quite well, while the absence of boundary layer effects leads to an inaccurate prediction of the shock location. In these last two cases, the farfield boundaries on the top and bottom of the domain were located five rotor radii away from the rotor plane. As mentioned before, no special boundary corrections were applied.

Figure 4 shows the topology of an O-H mesh for a single sector of a linearly twisted five-bladed rotor with a NACA 0012 blade section. The blade belonging to this sector has a finer mesh definition on its surface and appears to be solid black. The grid in the figure has been intentionally coarsened for presentation purposes.

The results presented for this rotor have been calculated on a mesh containing $96 \times 32 \times 56$ cells.

Figure 5 shows the contours of downwash velocity on a cutting plane that passes through the rotor hub. The calculation corresponds to the same rotor in Figure 4 with a collective pitch of 10° at the $\frac{3}{4}$ radius location. The tip Mach number for this calculation is 0.576. As can be seen in the figure, the wake contracts below the rotor plane, and a strong downwash is created at this azimuthal location. The calculation has already reached a steady-state hover condition, and this cutting plane is simply representative of the complete solution. Except for the near wake area, the contours of downwash velocity look almost the same for arbitrary azimuthal locations.

Figure 6 presents a prediction of thrust versus collective pitch with experimental results for the same five-bladed rotor. The results are reasonably close to the experimental curve, but have a slightly greater slope. Since the flow solver was demonstrated to be accurate using the experimental results above, it is believed that boundary conditions may have a greater influence on this case than the previous comparisons with experiment. In addition, the higher collective pitch results may be lacking in accuracy due to the omission of viscous effects and inaccurate capture of the shed vortex.

6.2 Aeroelastic Rotor Results

In order to document the aeroelastic capabilities of the ROTOR87 code, the aeroelastic model described in section 5 was applied to the same five-bladed rotor from Figure 4. In this case, the intention was to demonstrate the feasibility of obtaining aeroelastic responses for this type of flow, without large additional computational costs. For this purpose, the structure of this rotor was modelled as a flat plate of aluminum of thickness equal to 4% of the blade chord. The blade was considered to be cantilevered at its root, and the finite element model consisted of 72 16-degree-of-freedom flat plate elements covering the span of the blade. For this calculation, the first seven modes of vibration of the structure (with natural frequencies shown in Table 1) were kept. Of these seven modes, the first four are bending modes of varying order, the fifth and sixth modes are torsional modes, and the seventh mode of vibration is a sectional bending mode. All these modes are present in the aeroelastic response of this rotor, but they are clearly dominated in magnitude by the first bending mode.

The coupled aeroelastic calculation was carried out by computing an update to the position of the structure every 30 multigrid iterations of the flow solver. At each aeroelastic solution, forcing terms for all of the modes in the calculation were computed from the current pressure distribution around the blades. From these forcing terms, new modal coordinates were found and a new deflected blade surface was obtained. Using WARP3D, new deformed blocks for the multiblock mesh are calculated, and all grid metrics are recomputed. Sea-level atmospheric conditions were used to set up the freestream pressures and densities.

Mode number	Modal frequency (Hz)
1	2.103
2	82.487
3	647.200
4	2397.528
5	2495.157
6	6878.083
7	15572.949

Table 1: Modal Frequencies for the First Seven Modes of Vibration of a Helicopter Rotor Blade

Figure 7 shows the evolution of the first three modes of vibration of the structure during the aeroelastic calculation process (the second and third modes have been rescaled for presentation purposes). The other modes exhibit similar responses at much smaller magnitudes. For finer meshes, these responses require a longer time to converge to a steady solution due to the slower response of the wake system to the iteration process. It is in this case when an overrelaxation correction of the form described above can be useful. Figure 8 shows the position of the airfoil section (NACA 0012) at the tip of the rotor blade as the aeroelastic iterations proceed. The differences become small after the initial jump is overcome. This figure shows the extent to which the response is dominated by the first mode of vibration: a bending mode. Thus, it appears as if the tip of the blade is merely displacing upwards (mimicking a coning effect). Upon closer look, a small amount of twist is also present. The final lift coefficient for this aeroelastic rotor is slightly higher than the one for the rigid blade. This difference in lift could be expected due to the higher twist in the outboard sections of the aeroelastic rotor. However, one can see that the accuracy of this lift prediction depends heavily on the exactitude of the structural model, which was chosen somewhat arbitrarily. In particular, the torsional rigidity of this model most likely does not properly represent that of the real rotor, and therefore, the sectional pitch of the blade is likely to be incorrect. Further information about the real structural models of these blades is required in order to present comparisons between computed and experimental data.

6.3 Parallel Performance

Preliminary parallel efficiency results for the automatic design version of the multiblock code were presented earlier this year [8]. These performance figures, however, depend heavily on the size of the mesh used, the number of blocks in the mesh, and the load balancing of the calculation. Since one of the factors that enables the helicopter rotor calculations presented in this paper is the parallel implementation of the computational method, we found it appropriate to present the parallel performance results for the meshes used in this work. The

total number of internal cells in one sector of the five-bladed rotor is 172032, decomposed into 18 blocks of varying sizes. Figure 9 presents the parallel speedup for this calculation for a number of nodes ranging from 1 to 12. As we can see, with up to 8 processors, the curves show the high performance that can be accomplished with this implementation. For 12 processors, the performance drops heavily due to the following two effects: for a mesh of this size, the granularity of the solution becomes quite high. Moreover, the load balancing that can be accomplished with 18 blocks in 12 processors is rather poor. For future Navier-Stokes calculations, the granularity of the solution will become much lower, and more blocks will be introduced to allow a load balanced calculation with a larger number of processors. These two effects combined will sustain the high parallel performance of ROTOR87 for a much larger number of nodes.

7 Conclusions

Preliminary results for helicopter rotors in hover with aeroelastic deflections have been presented. Reasonable agreement with experimental pressure distributions and thrust curves has been achieved, but better resolution of the tip vortex and improved farfield boundary conditions are necessary for more accurate solutions.

The current formulation provides the accuracy and efficiency required to tackle fully resolved, unsteady, viscous, forward flight computations in a reasonable amount of time, including aeroelastic effects. For these calculations to be meaningful, proper consideration must be given to the issues of low numerical dissipation in the intrinsic algorithm, resolution of the tip vortex, and meaningful blade-vortex interaction.

These issues are currently under investigation, and will be addressed in the coming months.

Acknowledgements

The authors wish to thank Professor H. C. Curtiss and Jeff Keller of Princeton University for the five-bladed rotor data as well as their insights and comments. The second author was supported in part by a Fannie and John Hertz Foundation/Princeton Research Center Fellowship.

References

[1] J. J. Alonso and A. Jameson. Fully-implicit time-marching aeroelastic solutions. *AIAA paper 94-0056*, AIAA 32nd Aerospace Sciences Meeting, Reno, Nevada, January 1994.

[2] J. J. Alonso, L. Martinelli, and A. Jameson. Multigrid unsteady Navier-Stokes calculations with aeroelastic applications. *AIAA paper 95-0048*, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.

[3] A. Belov, L. Martinelli, and A. Jameson. Parallel computations of unsteady incompressible viscous flows with a fully-implicit multigrid driven algorithm. In *Proceedings of the 6th International Symposium on Computational Fluid Dynamics*, Lake Tahoe, NV, 1995.

[4] C. J. Borland. A multidisciplinary approach to aeroelastic analysis. *Computing Systems in Engineering*, 1(2–4):197–209, 1990.

[5] F. X. Caradonna and C. Tung. Experimental and analytical studies of a model helicopter rotor in hover. NASA TM 81232, AVRADCOM Research and Technology Laboratories, 1981.

[6] D. G. Holmes and S. S. Tong. A three-dimensional euler solver for turbomachinery blade rows. *J. of Engineering for Gas Turbines and Power*, 107:258–264, 1985.

[7] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA paper 91-1596*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 1991.

[8] A. Jameson and J. J. Alonso. Automatic aerodynamic optimization on distributed memory architectures. *AIAA paper 96*, AIAA 34th Aerospace Sciences Meeting, Reno, Nevada, January 1996.

[9] M. Kuntz. Rotor noise predictions in hover and forward flight using different aeroacoustic methods. *AIAA paper 96-1695*, 2nd AIAA/CEAS Aeroacoustics Conference, State College, PA, May 1996.

[10] M. A. Moulton, K. Ramachandran, M. M. Hafez, and F. X. Caradonna. The development of a hybrid CFD method for the prediction of hover performance. In *Proceedings of the 2nd International Aeromechanics Specialists' Conference*, Bridgeport, CT, October 1995.

[11] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA paper 96-0094*, AIAA 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1996.

[12] G. R. Srinivasan, J. D. Baeder, S. Obayashi, and W. J. McCroskey. Flowfield of a lifting rotor in hover: a Navier-Stokes simulation. *AIAA Journal*, 30(10):2371–2378, October 1992.

[13] G. R. Srinivasan and L. N. Sankar. Status of Euler and Navier-Stokes CFD methods for helicopter applications. In *Proceedings of the 2nd International Aeromechanics Specialists' Conference*, Bridgeport, CT, October 1995.

[14] B. E. Wake and J. D. Baeder. Evaluation of a Navier-Stokes analysis method for hover performance prediction. *Journal of the American Helicopter Society*, pages 7–17, January 1996.

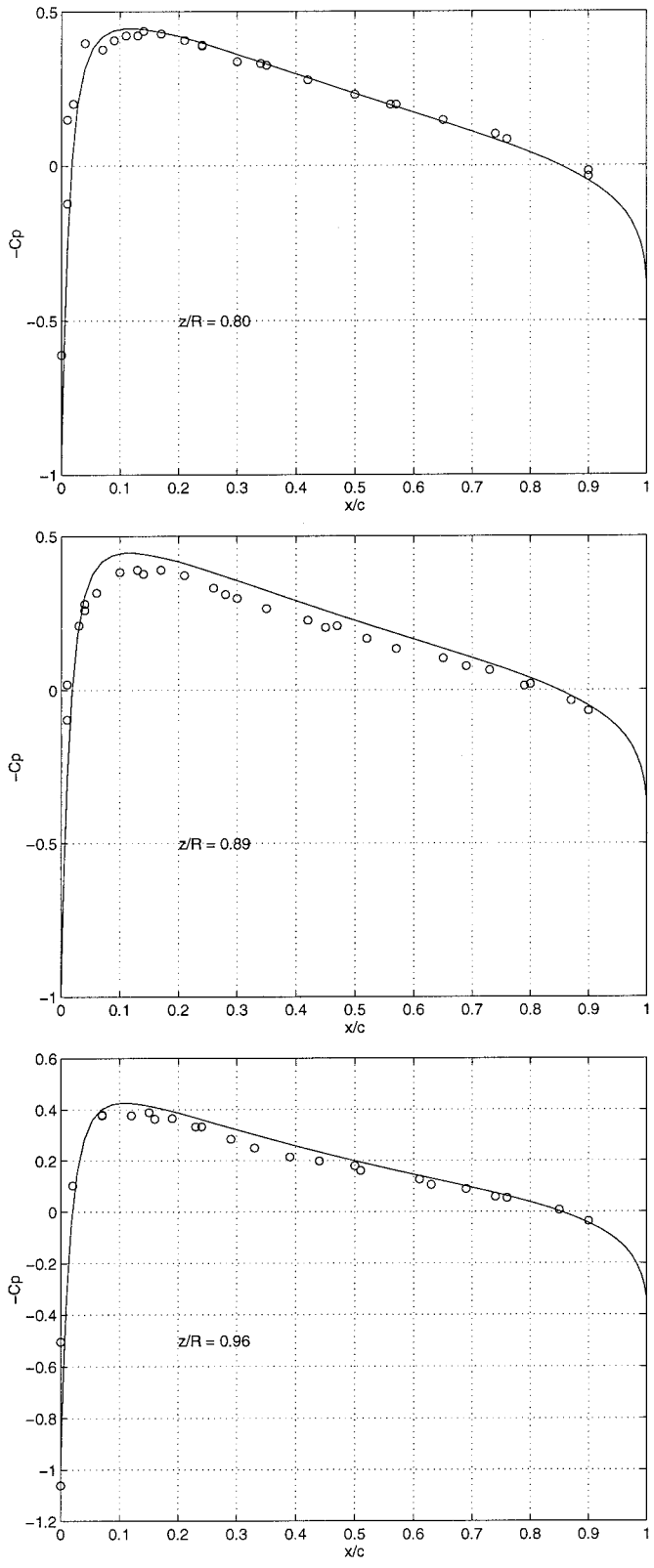
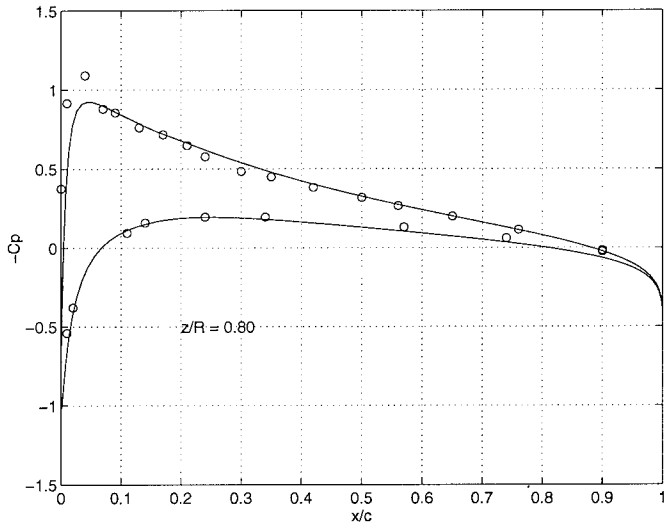
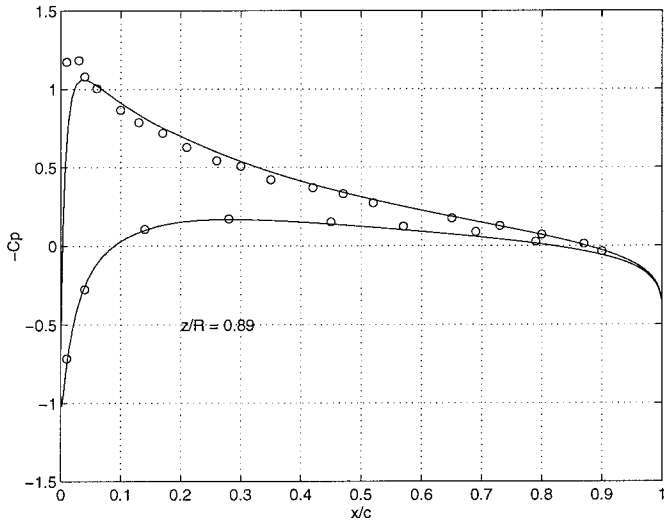


Figure 1: Pressure distribution on a nonlifting rotor in hover, $\theta_c = 0°$, $M_t = 0.520$.





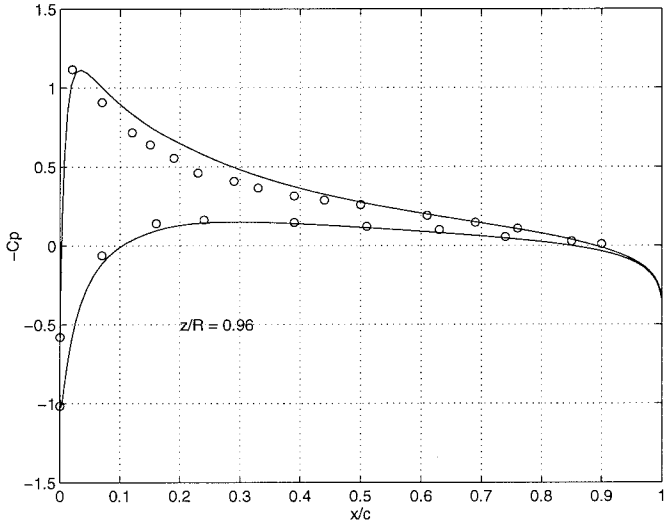
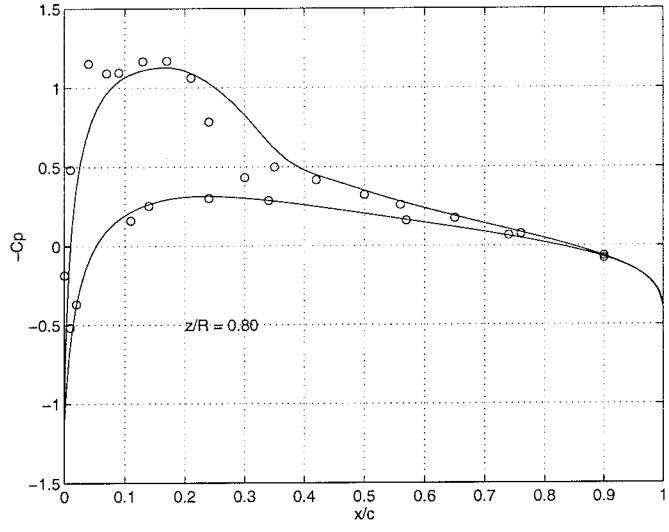
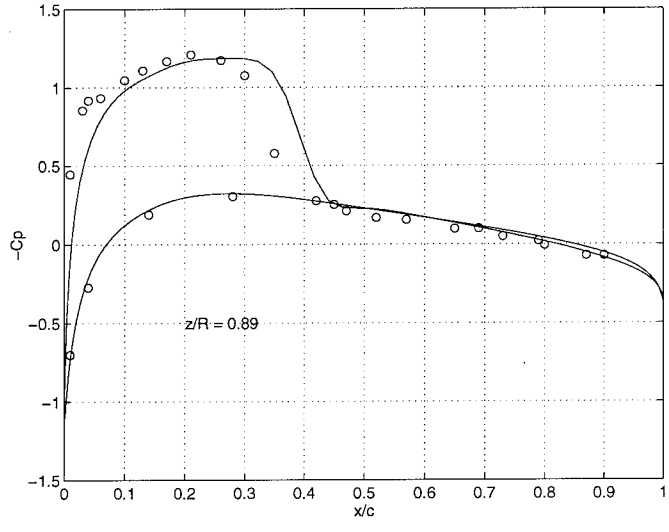


Figure 2: Pressure distribution on a rotor in hover, $\theta_c = 8°$, $M_t = 0.439$.





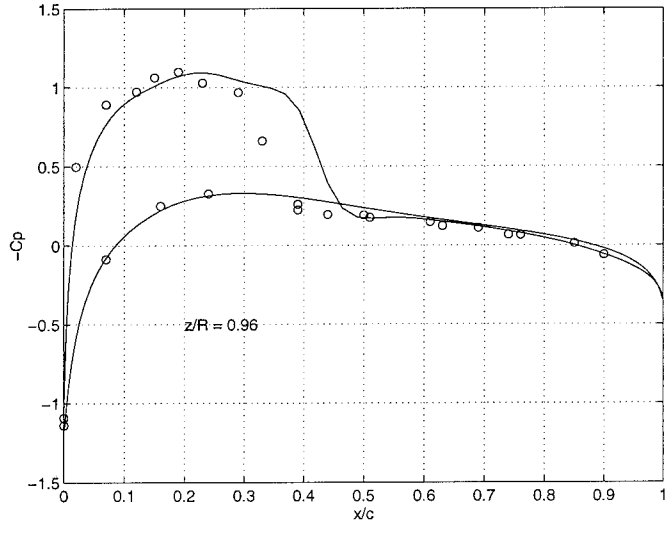


Figure 3: Pressure distribution on a rotor in hover, $\theta_c = 8°$, $M_t = 0.877$.

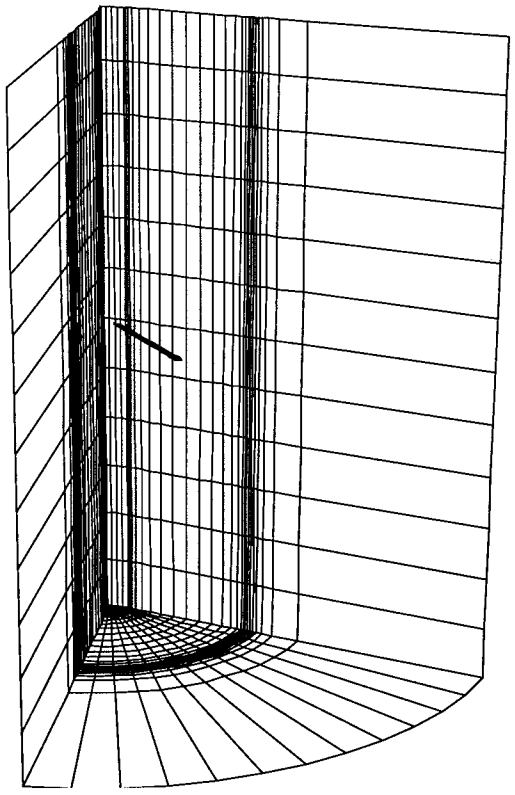


Figure 4: Perspective View of the O-H Mesh Used in One Sector of a Five-Bladed Rotor.

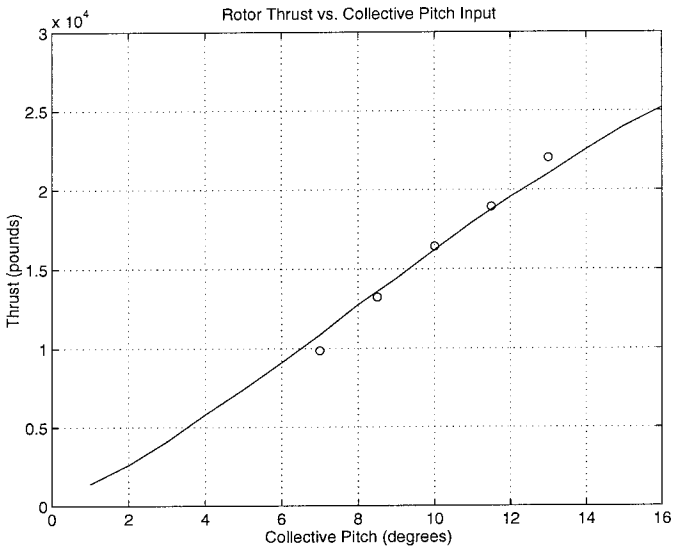


Figure 6: Thrust versus collective pitch for a five-bladed rotor, $M_t = 0.576$, o = numerical, – = experimental.

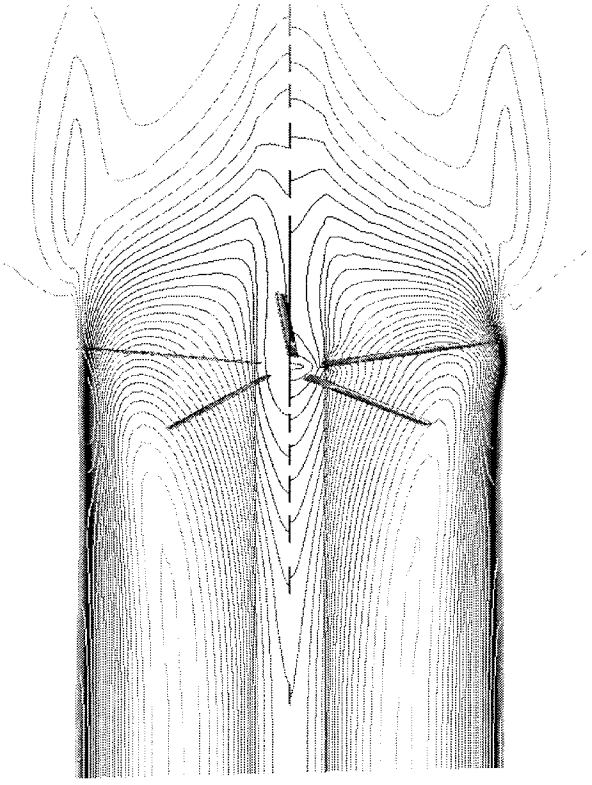


Figure 5: Contours of Downwash Velocity on a Cutting Plane Through the Rotor Hub.

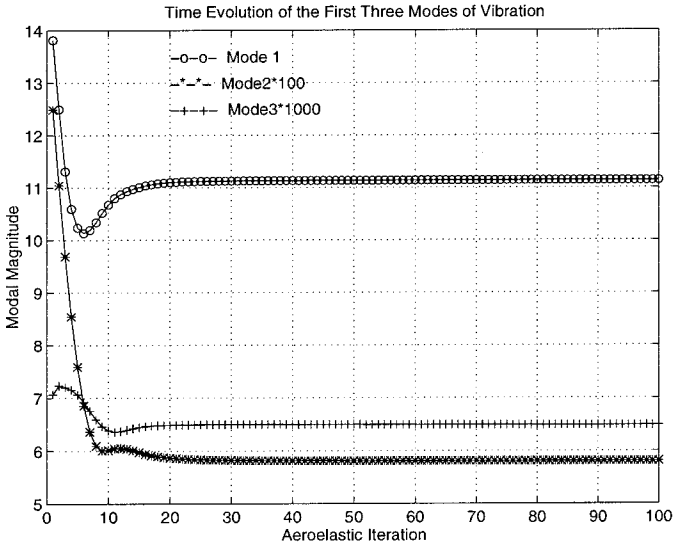


Figure 7: Aeroelastic Evolution of the First Three Modes of Vibration.

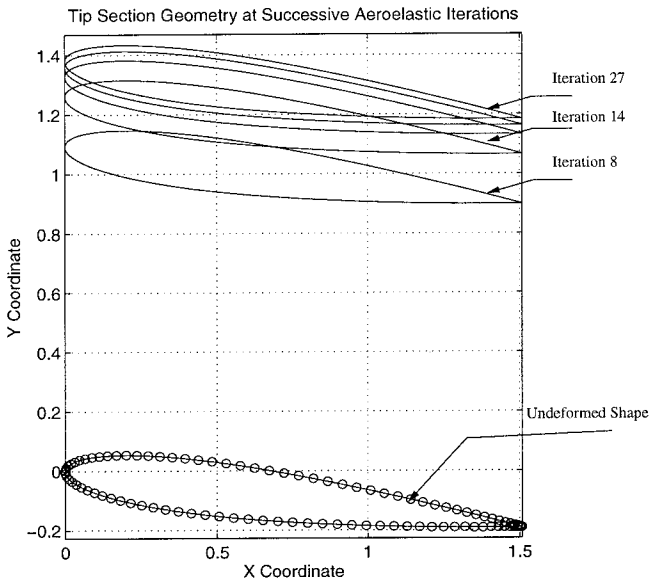


Figure 8: Blade Tip Deflection at Successive Aeroelastic Iterations.

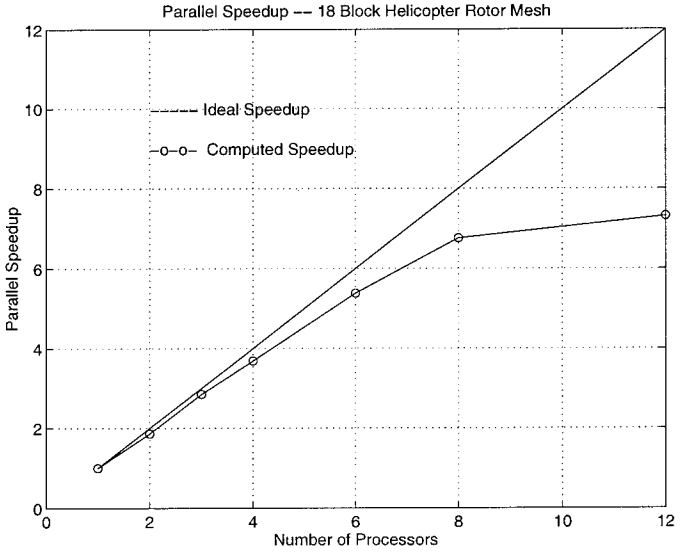


Figure 9: Parallel speedup for 18 block mesh.