Time-accurate simulation of helicopter rotor flows including aeroelastic effects

Scott G. Sheffer  
Princeton Univ., NJ

Juan J. Alonso  
Princeton Univ., NJ

Luigi Martinelli  
Princeton Univ., NJ

Antony Jameson  
Princeton Univ., NJ

AIAA, Aerospace Sciences Meeting & Exhibit, 35th, Reno, NV, Jan. 6-9, 1997

The following work presents recent developments in the simulation of helicopter rotor flows in the hover and forward flight regimes. The simulations are carried out by solving the full time-accurate compressible flow equations, both Euler and Navier-Stokes, on a moving multiblock mesh around a 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. The results demonstrate that the combination of highly efficient algorithms and high performance parallel computing yields an accurate and efficient method for the computation of complex vortical flows of interest to the rotorcraft community. (Author)
Abstract

The following work presents recent developments in the simulation of helicopter rotor flows in the hover and forward flight regimes. The simulations are carried out by solving the full time-accurate compressible flow equations, both Euler and Navier-Stokes, on a moving multiblock mesh around a 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. The results demonstrate that the combination of highly efficient algorithms and high performance parallel computing yields an accurate and efficient method for the computation of complex vortical flows of interest to the rotorcraft community.

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 use 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 integrate 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 in forward flight 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.

All numerical algorithms, regardless of their inception, carry a certain amount of numerical dissipation, which can be intrinsic to the discretization, or can be explicitly added to provide numerical stability. The amount of dissipation is usually proportional to the mesh cell size, and has disastrous effects on the resolution of helicopter wakes, which experimentally are shown to have concentrated tip vortex structures. It is then clear that more refined methods to provide vortex capturing are necessary in order to properly resolve a helicopter rotor wake. Jameson [10] has developed high accuracy, low dissipation schemes for fixed wing applications, and their performance in the calculation of helicopter flows must be assessed.

Furthermore, the simulation of the forward flight problem has not yet been thoroughly attempted 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 [14]. Alonso and Sheffer [4] have shown 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, they computed the flowfield of a helicopter rotor in hover with a moving mesh strategy. With this in mind, the solution of the forward flight problem only requires an additional factor in computational time equal to the number of blades in the rotor, since periodic boundary conditions can no longer be used.

The results presented in this paper are computed using a fully-implicit discretization of the Euler and Navier-Stokes 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 [9, 1, 2, 6]. 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 impractical.

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. 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 the blade tip area and the inclusion of full helicopter geometries (rotor and fuselage). Multiblock meshes used in Euler calculations are generated by the decomposition of an O-H mesh while Navier-Stokes meshes are constructed by a mesh generation package [20].

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 [11, 6, 4].

Governing Equations and Discretization

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

\[
\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial (f - f_v)}{\partial x} + \frac{\partial (g - g_v)}{\partial y} + \frac{\partial (h - h_v)}{\partial z} = 0 \tag{1}
\]

where \( \mathbf{w} \) is the vector of flow variables

\[
\mathbf{w} = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
\rho w \\
\rho E
\end{bmatrix},
\]

\( f, g, h \) are the convective flux vectors

\[
f = \begin{bmatrix}
\rho(u - x_t) \\
\rho u(u - x_t) + p \\
\rho v(u - x_t) + p \\
\rho w(u - x_t) + p \\
\rho E(u - x_t) + p u
\end{bmatrix},
\]

\[
g = \begin{bmatrix}
\rho(v - y_t) \\
\rho u(v - y_t) + p \\
\rho v(v - y_t) + p \\
\rho w(v - y_t) + p v \\
\rho E(v - y_t) + p v
\end{bmatrix},
\]

\[
h = \begin{bmatrix}
\rho(w - z_t) \\
\rho u(w - z_t) + p \\
\rho v(w - z_t) + p v \\
\rho w(w - z_t) + p v \\
\rho E(w - z_t) + p w
\end{bmatrix},
\]

and \( f_v, g_v, h_v \) are the viscous flux vectors

\[
f_v = \begin{bmatrix}
0 \\
\tau_{xx} \\
\tau_{xy} \\
-\tau_{yx} + \tau_{xx} u + \tau_{xy} v + \tau_{xz} w \\
\tau_{xz}
\end{bmatrix},
\]

\[
g_v = \begin{bmatrix}
0 \\
\tau_{yx} \\
\tau_{yy} \\
\tau_{yz} \\
-\tau_{yx} u + \tau_{yy} v + \tau_{yz} w
\end{bmatrix},
\]

\[
h_v = \begin{bmatrix}
0 \\
\tau_{xz} \\
\tau_{yz} \\
\tau_{xz} \\
-\tau_{yx} u + \tau_{yy} v + \tau_{yz} w
\end{bmatrix}.
\]
in each of the coordinate directions. The equations of motion of the fluid can then be written in integral form as

\[
\frac{d}{dt} \int \int w \frac{dV}{\partial V} + \int \int w(u - b) \cdot n \frac{dA}{\partial V} = \int \int \Pi \frac{dA}{\partial V}.
\]

(2)

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

\[
\frac{d}{dt}(w_{ij} V_{ij}) + E(w_{ij}) + D(w_{ij}) + NS(w_{ij}) = 0, \quad (3)
\]

where \( E(w_{ij}) \) are the convective Euler fluxes, \( D(w_{ij}) \) are the artificial dissipation fluxes added for numerical stability reasons and \( NS(w_{ij}) \) are the Navier-Stokes viscous fluxes. This equation (3) can be discretized implicitly as follows (drop the \( i,j \) subscripts for clarity):

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

where \( R \) is the sum of the three 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

\[
\frac{3}{2\Delta t} [w^{n+1} V^{n+1}] - \frac{3}{2\Delta t} [w^n V^n] + \frac{1}{2\Delta t} [w^{n-1} V^{n-1}] + R(w^{n+1}) = 0. \quad (5)
\]

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

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

In this case, the vector of flow variables \( w \) which satisfies the equation \( R^*(w) = 0 \) is the \( w^{n+1} \) 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, \( t^* \):

\[
\frac{dw}{dt^*} + R^*(w) = 0, \quad (6)
\]

to which one can apply the fast convergence techniques used for steady-state calculations, such as multigrid and Runge-Kutta time stepping. In order to increase the robustness of this numerical algorithm, the diagonal terms in the modified residual \( R^*(w) \) are treated in a point implicit fashion within the Runge-Kutta integration following Melson et al. [15]. In addition, one could use various preconditioning techniques to significantly enhance the convergence properties of the entire scheme as was demonstrated by Alonso and Pierce [3]. Applying this process repeatedly one can advance the flow field solution forward in time in a very efficient fashion.

If we wish to solve for the absolute velocities in the hover case without physically rotating the computational grid, we may follow the formulation of Holmes and Tong [8]. This involves adding source terms to equation (2) so that our governing equation becomes:

\[
\int \int \frac{\partial w}{\partial t} \frac{dV}{\partial V} + \int \int w(u - b) \cdot n \frac{dA}{\partial V} = \int \int \Pi \frac{dA}{\partial V} + \int \int \mathbf{T} \frac{dV}{\partial V}. \quad (7)
\]

The grid velocity vector is \( b = (-\Omega z, 0, \Omega x) \), while the additional term on the right hand side of equation (7) is introduced to account for the actual motion of the grid and has the form

\[
\mathbf{T} = \begin{pmatrix}
0 \\
\rho \Omega w \\
-\rho \Omega \nu
\end{pmatrix}
\]

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.

**Turbulence Model**

To model turbulence in the flow field the laminar viscosity is replaced by

\[ \mu = \mu_t + \mu_t \]

where the turbulent viscosity \( \mu_t \) is computed using the model of Baldwin and Lomax [5]. The Baldwin-Lomax model is an algebraic scheme that makes use of a two-layer, isotropic eddy viscosity formulation. This turbulence model is used as a starting point for future turbulent rotor simulations and is intended to demonstrate the concept of turbulent hover calculations.
Numerical Dissipation

Two numerical dissipation schemes are used in this work. The first is the standard Jameson-Schmidt-Turkel scalar switched scheme [13]. The second scheme is the Convective Upwind and Split Pressure (CUSP) scheme [10, 21]. Both of these schemes have been shown to have adequate accuracy. The CUSP scheme yields sharper shocks and less background dissipation than the scalar switched scheme at a slightly higher computational cost.

Parallel Multiblock Flow Solver: ROTOR87

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 [9]. 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, Navier-Stokes 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 [9] 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.

Parallelization Strategy

The original single block solver was parallelized using a static domain decomposition along the three coordinate directions of the block. The global mesh was partitioned into equal sized pieces, each of which was assigned to a single processor. 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 parallelization 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 [11]. Additional algorithms which account for communication costs can be developed in order to further improve the load balancing of the computation.

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, com-
munication is necessary. In order to minimize com-
munication cost, it was decided to pack all data that
needed to be sent from one processor to another
in a 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 commu-
ication, during the pre-processing step, each nodes
compiles a pointer list with all the entries in these
large arrays that need to be sent to all other process-
sors. 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 com-
municates all the necessary data for the blocks that
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.

**Boundary Conditions**

Four different types of boundary conditions are im-
posed 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 in-
viscid flows and the usual no-slip and adiabatic wall
boundary condition for Navier–Stokes solutions.
For moving meshes, the velocity of the mesh cells
must be taken into account in order to properly imple-
ment these kinds of boundary conditions.

At the top inflow boundary, and on the far-field side
walls of the mesh, non-reflecting boundary condi-
tions based on one-dimensional Riemann invariants
normal to the boundary are imposed. At the bottom
outflow boundary, a direct extrapolation of the flow
quantities is performed, as reported elsewhere in the
literature [19, 16, 18, 22].

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 ex-
perimental hover thrust coefficients. Every effort is
made to achieve boundary conditions that resemble
the conditions of a rotor in free hover.

**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 trans-
formation 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 ro-
tor 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 in-
teractive and time consuming process, and thus can-
not be embedded in the calculation process. Since
the mesh deflections are typically small, an auto-
matic procedure to achieve mesh deformations was
pursued.

Reuther et al. [17] have developed 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 chang-
ing values of the design variables in the opti-
mization problem. The mesh motion requirements for
the aeroelastic rotor simulation are perfectly addressed
by this regridding 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 cor-
respond 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 al-
ows 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 [17] for more details.

**Structural Equations and Coupling**

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

\[
[M] \ddot{q} + [C] \dot{q} + [K] q = F, \tag{8}
\]

where \([M], [C], \text{ and } [K]\) 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

\[
\ddot{\eta}_i + 2\zeta_i \omega_i \dot{\eta}_i + \omega_i^2 \eta_i = f_i, \quad i = 1, \ldots, N \tag{9}
\]

where \(\eta_i\) is the \(i\)-th normal coordinate, \(\omega_i\) is the
natural frequency of the \(i\)-th mode, \(\zeta_i\) 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 (9) 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:

$$\eta_i = \frac{f_i}{u_i^2}, \quad i = 1, \ldots, N$$  \hspace{1cm} (10)

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 to which global structural cell number they correspond.

Results

This section presents results of Euler and Navier-Stokes calculations for helicopter rotors in hover including aeroelastic deflections and forward flight Euler simulations including aeroelastic effects. The hover test cases 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 hover results in this paper were computed using the quasi-steady code.

Rigid Rotor–Euler Hover

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 (see Figure 1). Experimental results for this rotor at varying collective pitch angles and rotational speeds have been obtained by Caradonna and Tung [7]. Both the scalar switched and CUSP dissipation schemes were used for these test cases. 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 and blade-vortex interactions). Figure 2 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 3 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 outer portion of the blade. Figure 4 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 shock location is not predicted accurately. In addition, it may be seen that the CUSP results have a slight overacceleration after the shock which, for the inviscid case, has been observed in fixed wing calculations. The shocks produced using the CUSP scheme are crisper than those calculated with the scalar switched scheme. The CUSP result may represent the “true” inviscid solution better than the scalar switched scheme due to the lower amount of dissipation that it adds to the solution. In these last two hover 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 5 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 6 shows the contours of downwash velocity on a vertical cutting plane that passes through the rotor hub. The calculation corresponds to the same rotor in Figure 5 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 7 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.

**Rigid Rotor–Navier-Stokes Hover**

A Navier-Stokes calculation was performed on the Caradonna rotor at a collective pitch of 8 degrees and a tip Mach number of 0.877. Shock-free cases including viscous effects produced results that were very similar to the inviscid and experimental results and are not reproduced here. The grid used in this case was an H-H grid with \( 256 \times 64 \times 64 \) cells, with 128 cells on the surface of the airfoil in the chordwise direction and 48 cells in the spanwise direction. The Baldwin-Lomax turbulence model was used for a tip chord Reynolds number of 3,930,000. In order to have a well resolved boundary layer without the effects of spurious numerical dissipation, the CUSP scheme was used to compute this flow. Approximately 24 cells lie in the boundary layer of the rotor. This level of resolution has been shown to be satisfactory for these types of calculations [12, 21, 2] when using CUSP. Figure 8 shows experimental and numerical pressure coefficient distributions for this viscous test case. Contrary to expectations, the shock remains in approximately the same location as that predicted by the inviscid computations. The most likely causes for this disagreement with experimental measurements are the inadequacy of the Baldwin-Lomax turbulence model for flow cases which include shock-boundary layer interaction such as the present case, and the differences between transition locations in the computation and experiment. Transition in this calculation was fixed at the leading edge of the blade, which may not correspond to the experimental location of transition (which was not specified). Therefore, the prediction of the boundary layer thickness may not be correct causing a deviation in shock location from the experimental data. The small oscillation in the pressure coefficient near the aft of the airfoil sections may have been caused by a small separation bubble due to the turbulence model. The grid in this section varies smoothly and is not likely to have caused this oscillation. This result encourages us to pursue implementation of improved turbulence models which can adequately model the flow physics.

To reach an adequate level of convergence (five orders of magnitude reduction in the RMS residual of density), this calculation took 6.5 hours on 16 processors of an IBM SP-2. The computation was perfectly load balanced with 64 blocks of \( 32 \times 32 \times 16 \) cells.

**Rigid Rotor–Euler Forward Flight**

A series of time dependent forward flight calculations for the Caradonna rotor was made using the scalar switched scheme. The full two bladed rotor (24 blocks) is simulated in this case. The freestream conditions are set appropriately, while the rotor and attached grid are rotated at the correct angular velocity. The collective pitch of the blade was 8 degrees. The tip Mach number for this flight condition was 0.628 while the advance ratio was 0.30. A second order accurate discretization for the time derivative was chosen and a refinement study was done in time. Three calculations using 36, 72 and 144 time steps per revolution (corresponding to 10, 5 and 2.5 degrees per step) were made. Between 20 and 25 multigrid cycles were used at each time step in order to converge the pseudo-time iteration to an acceptable level. The results are presented in Figure 9 which shows the lift coefficient of the rotor as a function of the azimuthal angle. As would be expected, the series of lift coefficient histories converges as the number of time steps per revolution is increased. Figures 10 and 11 present pressure coefficient plots for three different span stations at two different azimuthal locations, 90 degrees (advancing side, blade perpendicular to freestream) and 270 degrees (retreating side, blade perpendicular to freestream). Approximately 4-6 revolutions were needed to attain a periodic solution for the lift coefficient. For the 144 time step per revolution case, approximately
<table>
<thead>
<tr>
<th>Mode number</th>
<th>Modal frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.103</td>
</tr>
<tr>
<td>2</td>
<td>82.487</td>
</tr>
<tr>
<td>3</td>
<td>647.200</td>
</tr>
<tr>
<td>4</td>
<td>2397.528</td>
</tr>
<tr>
<td>5</td>
<td>2495.157</td>
</tr>
<tr>
<td>6</td>
<td>6878.083</td>
</tr>
<tr>
<td>7</td>
<td>15572.949</td>
</tr>
</tbody>
</table>

Table 1: Modal frequencies for the first seven modes of vibration of a helicopter rotor blade

4 hours on 12 processors of an IBM SP-2 were used for each full revolution.

Additional calculations not presented here indicate that, at lower advance ratios, more revolutions of the blade are needed in order to achieve a periodic solution. This is because as the advance ratio is lowered, the wake is not convected as far away from the blade and therefore has a larger effect on the blade loading.

Aeroelastic Rotor—Euler Hover

In order to document the aeroelastic capabilities of the Rotor87 code, the aeroelastic model described in the section discussing the structural equations was applied to the same five-bladed rotor from Figure 5. 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 free vibration 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. As mentioned before, these modes will not reproduce the exact aeroelastic characteristics of the rotor, instead, they are intended to be used as a first estimate and to demonstrate the present numerical method.

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.

Figure 12 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. Figure 13 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.

Aeroelastic Rotor—Euler Forward Flight

A preliminary aeroelastic calculation was made using the five bladed rotor at a tip Mach number of 0.628 and an advance ratio of 0.30. The same mesh used in the hover cases was repeated at 72 degree intervals resulting in a total mesh size of \(5 \times 96 \times 32 \times 56 = 860,160\) cells with \(5 \times 18 = 90\) blocks. Aeroelastic deflections were computed for all blades, but only modal deflections for one of these blades are reported. The structural model is the same one used for the hover calculations. The reader is reminded once more that this is a contrived model that does not intend to accurately describe the structural properties of the real blade.
A total number of 36 time steps per revolution was used allowing for the motion of the blades at 10 degree intervals. Within each time step, 50 multigrid cycles were used to fully converge the coupled fluid/aeroelastic system. Information between equation systems was exchanged after every 5 multigrid cycles of the flow solver.

Figure 14 shows the time evolution of three of the bending modes during the last computed rotor revolution. For the first mode of vibration, a negative modal coordinate represents an upward tip displacement. As one can see, after 6 revolutions the modal coordinates have nearly reached a periodic state. In particular, it is interesting to note that the maximum modal deflections are achieved on the retreating side, which is not unreasonable given the assumptions made in the modeling of the structural properties of the blades.

The problem was solved using 30 processors of an IBM SP-2 system (6 processors per blade), achieving almost perfect load balance (4% variation between processors). Nine hours were required to compute a total of 6 revolutions.

Further verification using more realistic structural models and experimental data will follow in the coming months. It is important to point out that this calculation shows that forward flight rotor calculations including aeroelastic effects are indeed feasible on high performance parallel computing platforms.

Parallel Performance

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 inviscid meshes used in this work. 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. The total number of internal cells in one sector of the five-bladed rotor is 172032, decomposed into 18 blocks of varying sizes. Figure 15 presents the parallel speedup for this hover 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. More importantly, the load balancing that can be accomplished with 18 blocks in 12 processors is rather poor. For Navier-Stokes calculations, the granularity of the solution is much lower, and more blocks are used which allows a load balanced calculation with a larger number of processors. These two effects sustain the high parallel performance of Rotor87 for a much larger number of nodes.

Conclusions

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. 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. Forward flight results indicate a convergent numerical scheme with reasonable pressure distributions and aeroelastic responses. However, comparison with accurate experiments is still needed to fully validate the forward flight results.

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 first author was supported in part by a Fannie and John Hertz Foundation/Princeton Research Center Fellowship. Support provided by the IBM-CRA grant is also gratefully acknowledged by the authors.


Figure 1: Diagram of two-bladed rotor with spanwise cut locations.

Figure 2: Pressure distribution on a nonlifting rotor in hover, $\theta_c = 0^\circ$, $M_t = 0.520$. 
Figure 3: Pressure distribution on a rotor in hover, \( \theta_c = 8^\circ, M_t = 0.439 \).

Figure 4: Pressure distribution on a rotor in hover, \( \theta_c = 8^\circ, M_t = 0.877 \).
Figure 5: Perspective view of the O-H mesh used in one sector of a five-bladed rotor.

Figure 6: Contours of downwash velocity on a cutting plane through the rotor hub.

Figure 7: Thrust versus collective pitch for a five-bladed rotor, $M_t = 0.576$, $\circ$ = numerical, $-$ = experimental.
Figure 8: Pressure distribution on a rotor in hover including viscous effects, $\theta_c = 8^\circ$, $M_t = 0.877$, $\circ =$ experimental, $-$ = numerical.

Figure 9: Two bladed rotor lift coefficient versus azimuth for advance ratio of 0.30, $\circ =$ 36 steps per revolution, $-$ = 72 steps per revolution, $-$ = 144 steps per revolution.
Figure 10: Pressure distribution at azimuth of 90° for advance ratio of 0.30.

Figure 11: Pressure distribution at azimuth of 270° for advance ratio of 0.30.
Figure 12: Aeroelastic evolution of the first three modes of vibration in hover.

Figure 14: Time history of three bending modes in forward flight for a five bladed rotor.

Figure 13: Blade tip deflection at successive aeroelastic iterations in hover.

Figure 15: Parallel speedup for 18 block mesh.