Multigrid multiblock hovering rotor solutions

C. B. Allen
Department of Aerospace Engineering,
University of Bristol,
Bristol, UK

ABSTRACT

The effect of multigrid acceleration implemented within an upwind-biased Euler method for hovering rotor flows is presented. Previous work has considered multigrid convergence for structured single block rotor solutions. However, for forward flight simulation a multiblock approach is essential and, hence, the flow-solver has been extended to include multigrid acceleration within a multiblock solver. The requirement to capture the vortical wake development over several turns means a long numerical integration time is required for hovering rotors, and the solution (wake) away from the blade is significant. Hence, the solution evolution and convergence is different to a fixed wing case where convergence depends primarily on propagating errors away from the surface as quickly as possible, and multigrid acceleration is shown here to be less effective for hovering rotor flows. Previous single block simulations demonstrated that a simple multigrid *V*-cycle was the most effective, smoothing in the decreasing mesh density direction only, with a relaxed trilinear prolongation operator. This is also shown to be the case for multiblock simulations. Results are presented for multigrid computations with 2, 3, and 4, mesh levels, and a CPU reduction of approximately 80% is demonstrated for 4 mesh levels.

NOMENCLATURE

A	cell face area
E	total energy
$\mathbf{f}$	forcing function
$\underline{\mathbf{F}}$	flux vector
$\overline{\mathbf{F}}$	flux vector normal to cell face
$\overline{\mathbf{F}}^{\pm}$	split flux vector components normal to cell
$\mathbf{G}$	source term vector
I	number of iterations
$\mathbf{n}$	outward cell face normal vector
P	static pressure
$\mathbf{q}$	velocity vector
$\mathbf{r}$	co-ordinate vector
$\boldsymbol{R}$	rotation matrix
t	time
Δt	time step
u, v, w	velocity components
$\mathbf{U}$	conserved quantity vector
V	cell volume
x, y, z	inertial co-ordinates
α_j	time-step factor
γ	ratio of specific heats
ξ, η, ζ	parametric co-ordinates

Paper No. 2828. Manuscript received 10 February 2003, accepted 16 March 2004.

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

ρ	density
Θ	relaxation factor
ω	angular velocity vector
Ω	angular velocity

Subscripts

r	rotating reference frame

1.0 INTRODUCTION

Numerical simulation of the flow about a helicopter rotor is one of the most complex and expensive problems for CFD methods. The flow is non-linear, highly three-dimensional and unsteady and, furthermore, each blade moves into previously disturbed fluid. To capture the vortical wake, and particularly the tip vortex, requires a much higher grid density away from the surface than for a fixed wing case, where far-field mesh and solution are normally of little significance. Hence, rotary-wing flow simulations require finer meshes than fixed-wing flows, and a large numerical integration time for the wake to develop over several turns. These simulation can, therefore, be impractically expensive in terms of run times, and a multigrid approach can be valuable in reducing the cost. In this approach, which originates from works by Fedorenko[1], Bakhvarov[2] and later by Brandt[3,4], errors computed on the finest mesh are transferred to progressively coarser meshes to compute the corrections to the fine mesh solution. As larger time-steps are allowed on the coarser meshes, errors are propagated out of the domain faster than is possible on the fine mesh, resulting in faster convergence, see for example Refs 5,6,7,8. This paper describes the implementation and effectiveness of a multigrid scheme applied to hovering rotor flows.

Numerical solutions for hovering rotors have been considered previously, for inviscid flows, see for example Refs 9-18, and for viscous flows, see for example Refs 19-21. Some of this work has considered central-difference schemes, and in Ref. 13 it was shown that the numerical dissipation required has a significant effect on the solution, in terms of wake capturing and hence convergence rate. An upwind Euler solver is used in the current work, since this accurately models the physics of the flow, in terms of characteristic behaviour, and so is naturally dissipative. A finite-volume solver is presented, based on the flux-vector splitting of Van-Leer[22].

Structured grids are used here for rotary-wing grid generation. For simple blade geometries in hover, single block meshes can be used. However, for more complex blades and/or forward flight simulations, a multiblock approach is essential. Previous work has considered multigrid acceleration of a single block code[23, 24], and compared solution convergence and wake capturing for hovering rotors using single and multiblock meshes[25, 26]. This work demonstrated that the multiblock solution capture was sharper than the single block, and convergence was both faster and to a lower residual level than the single block. Hence, the flow-solver has since been extended to include multigrid within the multiblock scheme, and results are presented here.

In this paper the flow-solver, grid generation aspects for multiblock grids, and multigrid implementation procedures are first presented. Then results of multigrid computations of hovering multibladed rotor flows are considered for varying numbers of grid levels. Reductions in required CPU times of approximately 80% are demonstrated for a four-level computation. As presented in[23] this is a smaller CPU reduction than would normally be expected for a fixed wing case where reductions of 95% are common[24]. The difference is due to the requirement to capture the vortical wake over several turns, which requires a much longer numerical integration time than a fixed-wing case. Hence, convergence is not simply dependent on propagating all errors away from the solid surface as quickly as possible, as is the case for fixed-wing flows, but on the development of the complex wake over several turns.

2.0 EULER SOLVER

The computation of flows about a rotor blade in hover requires a rigid but rotating grid. However, instead of solving an unsteady problem, the three-dimensional unsteady Euler equations are transformed to a blade-fixed rotating reference frame. In the rotating frame the hover case is then a steady problem. The equations may be transformed in terms of absolute or relative velocities. Absolute velocities are used here. If the frame rotates with angular velocity of $\boldsymbol{\omega} = [\Omega_x, \Omega_y, \Omega_z]^T$, and the absolute velocity vector in the rotating frame is denoted by $\mathbf{q_r} = [u_r\ v_r\ w_r]^T$, the resulting Euler equations in integral form are then:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_r} \mathbf{U_r} dV_r + \int_{\partial V_r} \mathbf{F_r}.\mathbf{n_r} \mathrm{d}S_r + \int_{V_r} \mathbf{G_r} \mathrm{d}V_r = 0 \qquad \ldots (1)$$

where

$$\mathbf{U_r} = \begin{bmatrix} \rho \\ \rho u_r \\ \rho w_r \\ \rho w_r \\ E \end{bmatrix}, \mathbf{F_r} = \begin{bmatrix} \rho[\mathbf{q_r} - (\omega \times \mathbf{r})] \\ \rho u_r[\mathbf{q_r} - (\omega \times \mathbf{r})] + P\mathbf{i_r} \\ \rho v_r[\mathbf{q_r} - (\omega \times \mathbf{r})] + P\mathbf{j_r} \\ \rho w_r[\mathbf{q_r} - (\omega \times \mathbf{r})] + P\mathbf{k_r} \\ E[\mathbf{q_r} - (\omega \times \mathbf{r})] + P\mathbf{q_r} \end{bmatrix}, \mathbf{G_r} = \begin{bmatrix} 0 \\ \rho(\omega \times \mathbf{q_r}).\mathbf{i_r} \\ \rho(\omega \times \mathbf{q_r}).\mathbf{j_r} \\ \rho(\omega \times \mathbf{q_r}).\mathbf{k_r} \\ 0 \end{bmatrix} \qquad \ldots (2)$$

Here $\mathbf{G_r}$ is the source term resulting from the transformation, $\mathbf{r} = [x_r, y_r, z_r]^T$ is the co-ordinate vector, and $\mathbf{n}$ is the outward unit normal at each cell face. The equation set is closed by:

$$P = (\gamma - 1)[E - \frac{\rho}{2} q^2]. \qquad \ldots (3)$$

2.1 Upwind difference scheme

A finite-volume upwind scheme is used to solve the integral form of the Euler equations (Equation (1)), since by correctly modelling the characteristic behaviour of the flow upwind schemes are naturally dissipative. The flux-vector splitting of Van-Leer[22, 27] is used.

For each cell face a local orthogonal coordinate system (ξ, η, ζ) is adopted, where the principal coordinate direction ξ is normal to the cell face. The unit normal to each cell face is dfined as the unit vector in the ξ direction, $\mathbf{n}_\xi$. Unit vectors in two directions, lying in the cell face, $\mathbf{n}_\eta$ and $\mathbf{n}_\zeta$, are then defined to form an orthogonal axis system. The total flux across each cell face is calculated, using the upwind representation, in the local axis system. A third order spatial discretisation is adopted, along with a continuously differentiable flux limiter[28].

Once each cell face flux has been computed the flux integral is evaluated by rotating each component back to cartesian coordinates using the local rotation matrix. More details of the flow-solver can be found in, Refs 23, 24, 26.

2.2 Time-stepping scheme

An explicit multi-stage Runge-Kutta scheme is used to integrate the equations forward in time. However, the four-stage scheme of Jameson *et al*[29] is not efficient for an upwind scheme, since the stability limit is greatly reduced from the $2\sqrt{2}$ value for a central difference scheme[30]. A three-stage scheme is used, which can operate at a CFL number of 1·5. As this is a steady flow local time-stepping is used to accelerate convergence.

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

The time-stepping scheme used for each computational cell to integrate from time level n to n+1 is:

$$\mathbf{U}_{\mathbf{r}}^{\,n+\alpha_j} = \mathbf{U}_{\mathbf{r}}^{\,n} - \alpha_j \frac{\Delta t}{V}\left\{ V\mathbf{G}_{\mathbf{r}}(\mathbf{U}_{\mathbf{r}}^{\,n+\alpha_{j-1}}) + \sum_{k=1}^{6}[R]_k^{-1}\left[\overline{\mathbf{F}}^{+}(\mathbf{U}_{\mathbf{r}}^{+})_k^{n+\alpha_{j-1}} + \overline{\mathbf{F}}^{-}(\mathbf{U}_{\mathbf{r}}^{-})_k^{n+\alpha_{j-1}}\right]A\right\} \quad \ldots (4)$$

with $\alpha_{(0,1,2,3)} = 0, \frac{1}{4}, \frac{1}{2}, 1$. In the above, V is the cell volume, F the split flux components, $[\boldsymbol{R}]$ the local rotation matrix, $\boldsymbol{A}$ the cell face area, and k represents the six cell faces.

2.3 Boundary conditions

The flow-solver has been coded such that each block boundary has a boundary condition tag (listed in Table 1), a neighbouring block number, and an orientation flag. The only restriction applied is that a boundary can only have one type of tag, as this ensures the connectivity/index algorithm could be fully automated for any multigrid level.

At upper, lower, and spanwise farfield boundaries, characteristic based conditions are applied. At periodic planes, the conserved variables required in neighbouring cells (i.e. the cells adjacent to the periodic planes are treated as internal cells) are evaluated using the following transformations:

$$\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix}_{Downstream} = \begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & \mathrm{Cos}(2\pi/N_b) & \mathrm{Sin}(2\pi/N_b) & 0 & 0\\ 0 & -\mathrm{Sin}(2\pi/N_b) & \mathrm{Cos}(2\pi/N_b) & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix}_{Upstream} \quad \ldots (5)$$

$$\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix}_{Upstream} = \begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & \mathrm{Cos}(2\pi/N_b) & -\mathrm{Sin}(2\pi/N_b) & 0 & 0\\ 0 & \mathrm{Sin}(2\pi/N_b) & \mathrm{Cos}(2\pi/N_b) & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{bmatrix}_{Downstream} \quad \ldots (6)$$

where N_b is the number of blades.

3.0 MULTIBLOCK GRID GENERATION

For N-bladed hovering rotor calculations only one blade, and 1/N of the cylindrical domain, need be considered, and the up- and downstream boundary planes can be treated as periodic boundaries. The grid is generated by first generating a small extent blade-fixed C-H grid around the solid surface, generated in a similar manner to a single block grid, i.e. incorporating a periodic tranformation for up- and downstream planes, but with a variable fraction of the disk and a small farfield distance. See Ref. 31 for details of single block grid generation.

Table 1
Boundary tags

–1 Solid surface (include in loads calculation)
0 Solid surface
1 Farfield
2 Internal face (needs neighbour number and orientation)
3 Periodic downstream (needs neighbour number and orientation)
4 Periodic upstream (needs neighbour number and orientation)

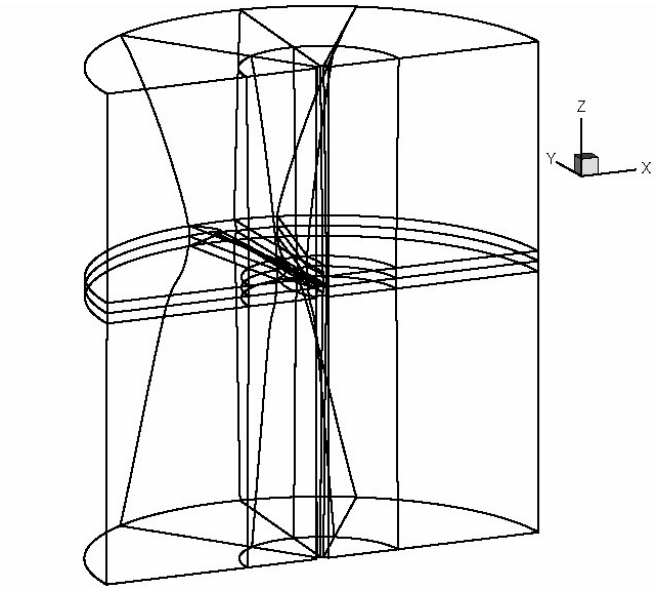


Figure 1. Block boundaries for 36 block mesh.

From this near blade grid, which has a rectangular type farfield, cylindrical grids are generated upstream, downstream, above and below the blade to fill the required domain. A transfinite interpolation scheme is used to generate the mesh within each block[32]. A simple Laplcian-type smoothing scheme is then applied to the entire grid, including across block boundaries, to improve smoothness[33,31].

The geometry considered is that of the well known Caradonna-Tung two-bladed rotor[34]. The rotor has no twist or taper, an aspect ratio six, with a constant NACA0012 section. The incidence is 8°. In the hover case only one blade, and half the physical domain, need be considered. The axis system used is z vertical, y spanwise (from root to tip) and x to give an orthogonal system. Hence the rotation vector is $\omega = [0, 0, \Omega_z]$, and the $y = 0$ plane is the periodic plane.

The multiblock grid was generated to contain the minimum number of blocks possible to ensure the maximum number of cells within each block, to optimise the multigrid effectiveness. The mesh contains 36 blocks, and the block boundaries are shown in Fig. 1. The dimensions are chosen so that each block may be split in each direction three times, i.e. four levels of multigrid are used. The surface grid has dimensions 113×49, and blade-fixed C-H has dimensions $161 \times 73 \times 25$, the upstream and downstream blocks have 49 radial points, there are 25 vertical points above the blade, and 57 vertical points below the blade. Hence, the total mesh contains around $1{\cdot}6 \times 10^6$ points. The hub plane, blade, and grid in a plane near the blade tip is shown in Fig. 2, along with a magnified view.

4.0 MULTIGRID SCHEME

Explicit time-stepping schemes lend themselves well to multigrid acceleration, see for example[5]. In a multigrid approach, errors computed on the finest mesh are transferred to progressively coarser meshes to compute the corrections to the fine mesh solution. As larger time-steps are allowed on the coarser meshes, errors are propagated out of the domain faster than is possible on the fine mesh, resulting in faster convergence.

The scheme is presented here using N to represent the mesh number ($N = 1$ is finest), V is the cell volume, $\mathbf{R}$ represents the residual, $\mathbf{U}^R$ and $\mathbf{U}^S$ represent the restricted and smoothed (updated) solution respectively, and I_N and I^N are the number of iterations performed on mesh N in the decreasing and increasing mesh density directions (The $\mathbf{r}$ subscript is omitted from $\mathbf{U}$ for clarity). For a simple 'V' cycle, the scheme can be written as:

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

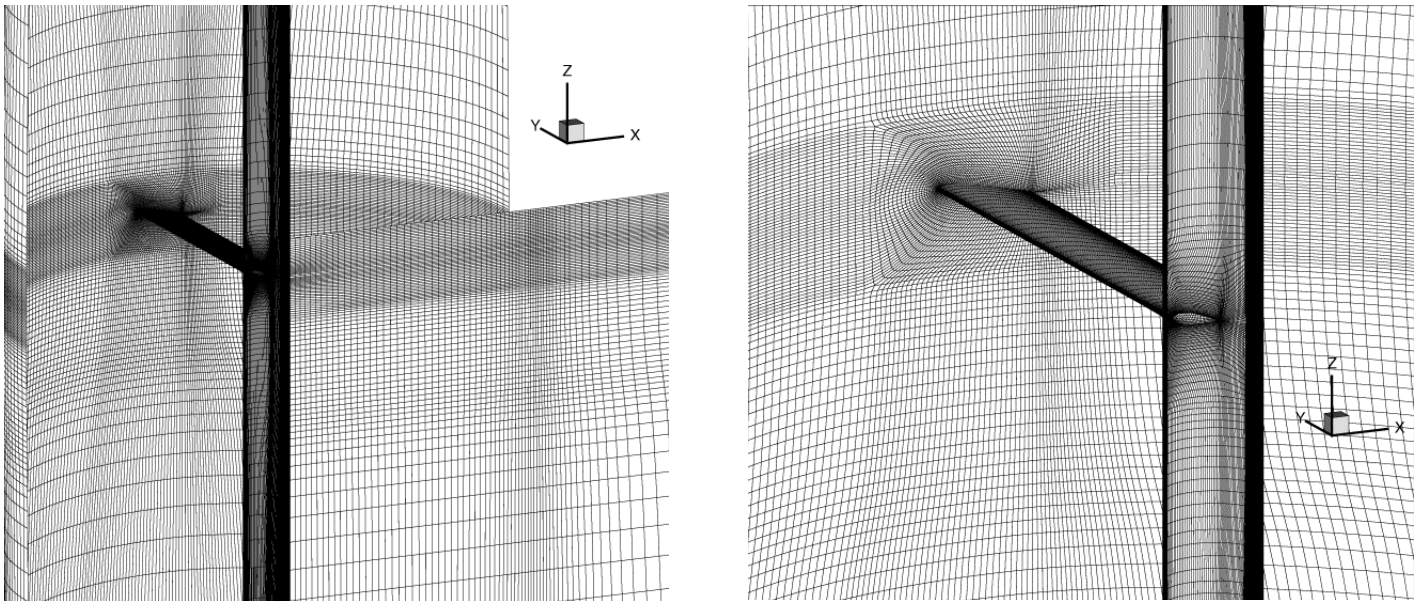


Figure 2. Hub, blade, plane near blade tip and part of downstream plane.

DO FOR $NB = 1 -> NBLOCK$

Perform I_1 iterations (smoothing sweeps) on mesh 1 (finest)

$$\mathbf{U}_1^{n+\alpha_j} = \mathbf{U}_1^n - \alpha_j \frac{\Delta t}{V_N}\mathbf{R}(\mathbf{U}_1^{n+\alpha_{j-1}}) \qquad j = 1...3$$

$$\Rightarrow \mathbf{U}_1^S$$

set $\mathbf{f}_1 = 0$

DO FOR $N = 2 -> NMESH$

Restrict solution to next finest mesh

$$\mathbf{U}_N^R = \frac{\sum_{cell=1}^{8} \mathbf{U}_{N-1}^S V_{N-1}}{\sum_{cell=1}^{8} V_{N-1}}$$

Evaluate forcing function

$$\mathbf{f}_N = \mathbf{R}_N(\mathbf{U}_N^R) - \Sigma_{cell=1}^{8}\left\{\mathbf{R}_{N-1}(\mathbf{U}_{N-1}^S) - \mathbf{f}_{N-1}\right\}$$

Perform I_N iterations on mesh N

$$\mathbf{U}_N^{n+\alpha_j} = \mathbf{U}_N^n - \alpha_j \frac{\Delta t}{V_N}\left(\mathbf{R}(\mathbf{U}_N^{n+\alpha_{j-1}}) - \mathbf{f}_N\right) \qquad j = 1...3$$

$$\Rightarrow \mathbf{U}_N^S$$

ENDDO

DO FOR $N = NMESH - 1 -> 1$

Compute corrections on coarser mesh

$$\Delta\mathbf{U}_{N+1} = \mathbf{U}_{N+1}^S - \mathbf{U}_{N+1}^R$$

Pass, prolong corrections to current mesh

$$\Rightarrow \mathbf{U}_N$$

$$\mathbf{U}_N^S = \mathbf{U}_N^R + \Delta\mathbf{U}_N$$

Perform I^N on mesh N

$$\mathbf{U}_N^{n+\alpha_j} = \mathbf{U}_N^n - \alpha_j \frac{\Delta t}{V_N}\left(\mathbf{R}(\mathbf{U}_N^{n+\alpha_{j-1}}) - \mathbf{f}_N\right) \qquad j = 1...3$$

$$\Rightarrow \mathbf{U}_N^S$$

ENDDO

ENDDO

The forcing function, $\mathbf{f}_N$, is essential for the success of the multigrid approach. This can be considered a correction to the residual computed on one level from the restricted solution from the next mesh upwards. It ensures that the errors on all mesh levels are driven by the residuals on the finest mesh, i.e. if the solution has converged on the finest mesh, subsequent multigrid cycles will result in no changes to the solution.

The prolongation step, i.e. passing corrections up to next finest mesh, can be performed using several methods. The simplest way is to use direct injection, wherein all eight cells surrounding each coarser mesh cell are given the same correction. More commonly a trilinear interpolation scheme is used. As an upwind scheme is used here, an upwind solution correction interpolation can be used. However, although theoretically an upwind-biased prolongation results in a more robust and stable scheme, previous work has been shown that this does not have a significant effect on convergence, see for example[35,36]. Hence, the extra computational requirement of using such a prolongation does not seem justified and a trilinear interpolation scheme is used here.

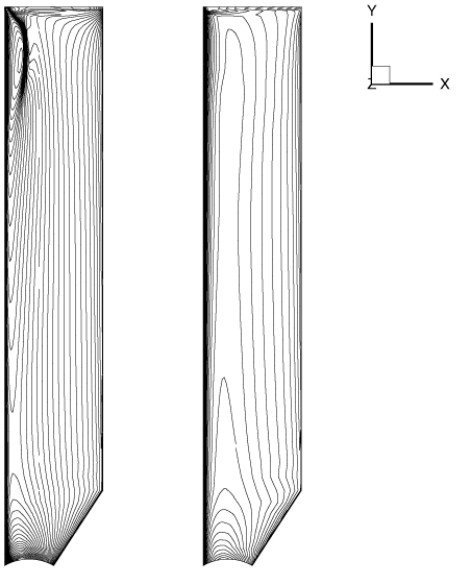


Figure 3. Upper and lower surface pressure coefficient contours.

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

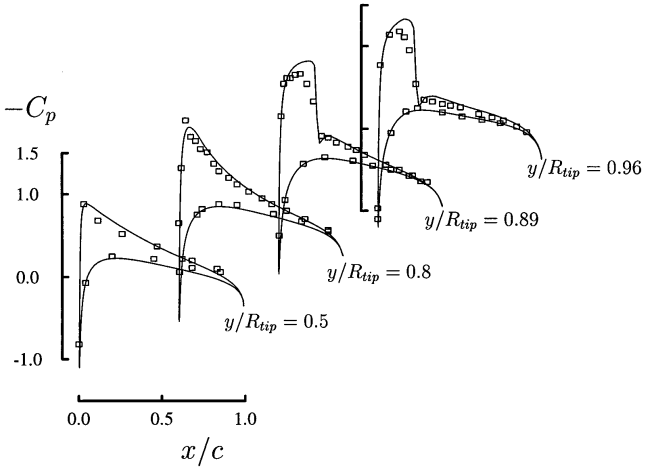


Figure 4. Computed and experimental pressure coefficient at radial stations.

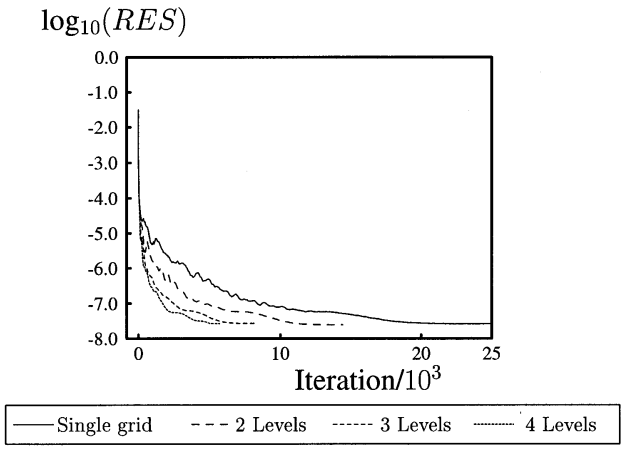


Figure 5. Multigrid convergence history.

5.0 RESULTS

The Caradonna-Tung hovering rotor test case with a tip Mach number of 0·794 was run using a single mesh (finest), and as multigrid with 2, 3, 4 mesh levels. The upper and lower surface pressure coefficient contours shown in Fig. 3 (57 contour levels are plotted between –1·5 and 1·2 for C_P).

Also shown, in Fig. 4, are surface pressure coefficient variations at several spanwise stations compared with experimental results (plotted as squares). The results compare well, except for the computed result showing stronger shock strength, which is expected since viscosity will weaken the shock strength compared to an inviscid result. The vorticity field, shown in a later figure, demonstrates that the tip vortex is diffused fairly rapidly by the numerical scheme, and this could also contribute to the difference in shock strength. The induced velocity from the vortex may be smaller than in reality, which would lead to a slightly higher effective incidence of the blade at the outboard stations.

5.1 Effect of multigrid scheme

The convergence history, in terms of average global residual level against iteration number for the single mesh, and with 2, 3, 4 mesh levels, is shown in Fig. 5. (It should be noted here that one 'iteration' in any multigrid scheme takes the same amount of CPU time. For example one cycle of a 3-level calculation, with two iterations per mesh, would take the equivalent of (2 + 2/8 + 2/64 + communication overhead) iterations on a single mesh. Hence, the x-axis could just as easily be CPU time.)

The residual is defined as:

$$RES = \sqrt{\frac{1}{ni.nj.nk}\sum_{i=1}^{ni}\sum_{j=1}^{nj}\sum_{k=1}^{nk}\left\{\left(\frac{\Delta\rho}{\Delta t}\right)^2+\left(\frac{\Delta\rho u}{\Delta t}\right)^2+\left(\frac{\Delta\rho v}{\Delta t}\right)^2+\left(\frac{\Delta\rho w}{\Delta t}\right)^2+\left(\frac{\Delta E}{\Delta t}\right)^2\right\}_{i,j,k}} \quad \ldots (7)$$

The most effective multigrid scheme was found to be to use a simple V-cycle solving on the way down only, i.e. the 'sawtooth' cycle as shown in Fig. 6. Although also smoothing on the way up resulted in fewer multigrid cycles required for convergence, in fact more total iterations and hence CPU time were required. The fastest convergence was achieved smoothing only on the way down, with a relaxed prolongation operator. When passing the error from mesh $N+1$ to mesh N the correction is relaxed by:

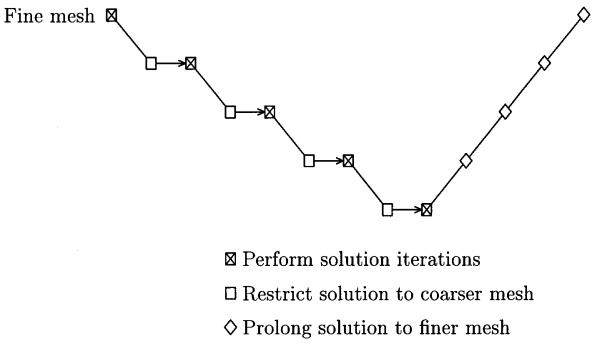


Figure 6. Multigrid cycle.

$$\mathbf{U}_N^S = \mathbf{U}_N^R + \Theta\Delta\mathbf{U}_N \quad \ldots (8)$$

where $\Theta = 0{\cdot}8$ is the optimum value regardless of number of mesh levels. The optimum iterations scheme was found to be:

$$I_{1,2,3,4} = 2, 8, 16, 32 \qquad I^{1,2,3,4} = 0, 0, 0, 0 \quad \ldots (9)$$

However, the residual history is not always an accurate measure of convergence for rotary flows, and neither are the blade loads. The vortical wake takes many iterations to develop, and the development may not have a huge effect on either the global residual level or the blade loads. Hence, it is interesting to examine the flowfield development.

Figure 7 shows vorticity contours for the downstream periodic plane, i.e. the $y = 0$ plane, 90° behind the blade, during the solution development for the various simulations. The vorticity in each cell is defined as:

$$\omega = |\nabla \times \mathbf{q}_r| \quad \ldots (10)$$

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

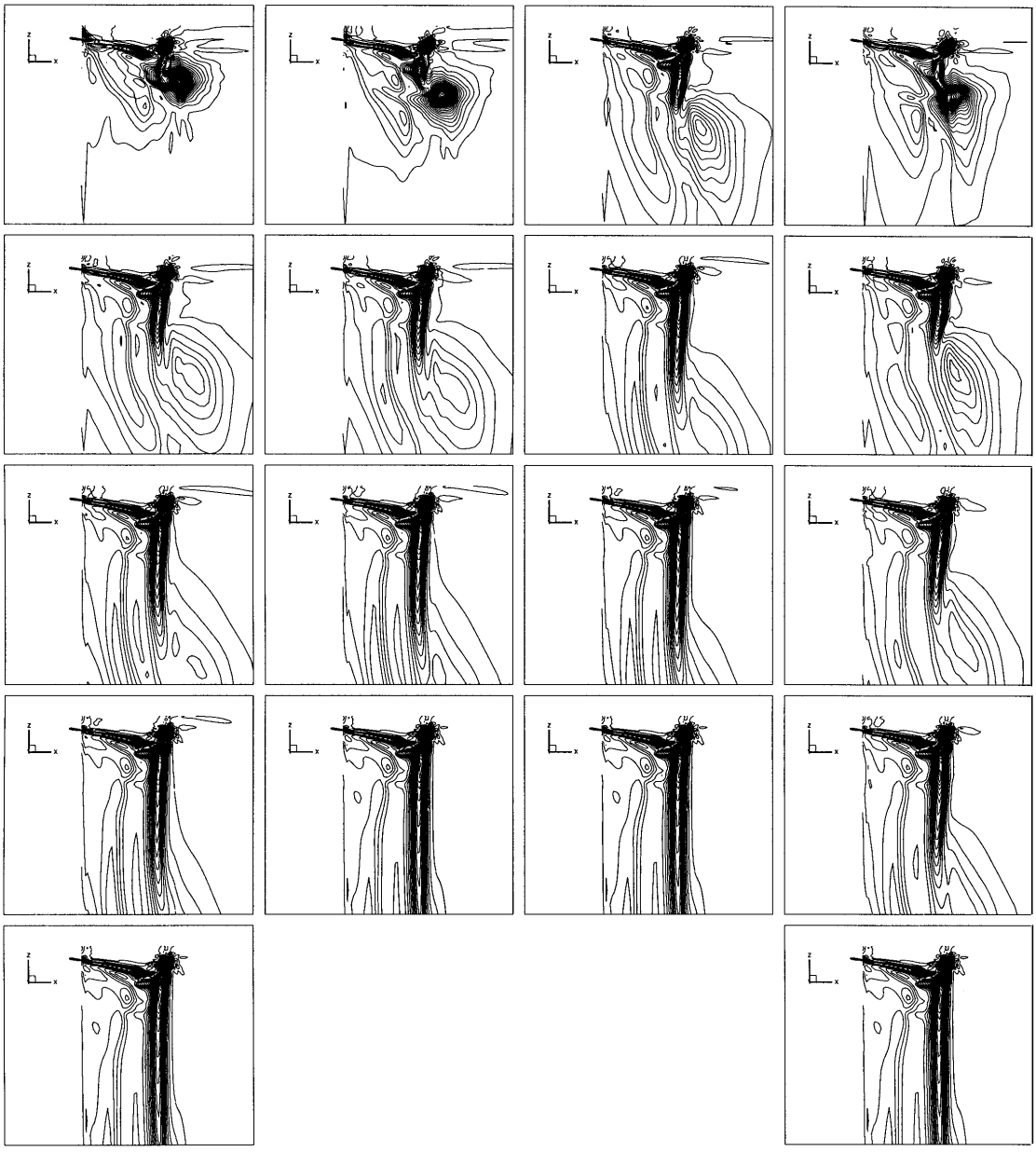


Figure 7. Vorticity Contours in Plane 90º behind blade. (a) Single level every 5,000 timesteps, (b) two-level every 3,000, (c) three-level every 2,000, (d) four-level every 1,000.

and this is transformed to a surface integral for evaluation. 100 contour levels are plotted between 0 and 0·8 in each picture. The single block solution is shown every 5,000 time-steps on the left, the two-level multigrid solution every 3,000 effective timesteps down the second column, three-level multigrid solution every 2,000 effective timesteps third column, and four-level multigrid solution every 1,000 effective timesteps in the right column. Detailed examination indicates that the speed-up's obtained by the multigrid scheme are around 2·2 × for two levels, 3·5 × for three levels, and 5 × for four levels.

The accelerations obtained are not as large as would be expected for fixed-wing cases. Previous works have shown that accelertions of over ten would be expected for inviscid aerofoil computations using central-difference and upwind schemes, see for example Refs 5, 37, 38, 39. In three dimensions, fixed-wing computations can result in accelerations approaching 20, see for example Refs 6, 24, 40. The lower accelerations exhibited here are due to the slow development of the vortcial wake and the tip vortex over several turns in the rotary case. This accurate wake capture is a requirement not present in a fixed-wing case, where the errors associated with the solid surface boundary conditions simply have to be propagated away as quickly as possible.

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press

6.0 CONCLUSIONS

An upwind Euler solver has been presented, with multigrid convergence acceleration, and applied to multibladed hovering rotor flow using a multiblock approach. A reduction in required CPU time of over 80% has been demonstrated using four multigrid levels. However, the convergence of fixed-wing and rotary-wing computations are vastly different due to the different wakes. The vortical wake of a hovering rotor must develop over several turns before convergence is achieved, whereas for fixed-wing computations the far-field grid and solution have little significance. Hence, the multigrid approach is less effective and the CPU reduction is not as large as for a fixed-wing case, where 90%+ would be expected in three dimensions.

The most effective convergence has been achieved using a simple V-cycle, with increasing solution iterations on each level, solving when moving in the decreasing mesh density direction only. A trilinear prolongation operator is employed with a relaxation parameter which has an optimum value of 0·8, regardless of the number of mesh levels employed.

It should also be noted that implementing a multiblock approach limits the number of multigrid levels possible. A $1{\cdot}6 \times 10^6$ grid point single block mesh would allow at least five levels of multigrid, whereas only four are possible here.

REFERENCES

1. FEDORENKO, R.P. The speed of convergence of one iterative process, *Zh Vych Mat,* **3**, (4), 1964, pp 559-564 (*USSR Comp Math and Math Phys*, 1964, (4), pp 227-235).
2. BAKHVALOV, N.S. On the convergence of a relaxation method with natural constraints of the elliptic operator, *Zh Vych Mat,* 1966, **5**, (6), pp 861-885. (*USSR Comp Math and Math Phys,* 1966, (6), pp 101-135).
3. BRANDT, A. Multi-level adaptive solutions to boundary-value problems, *Math Comp*, 1977, **31,** (138), pp 333-390.
4. BRANDT, A. Guide to Multigrid Development, *Multigrid Methods* (HACKBUSCH and TROTTENBERG, Eds), Springer-Verlag, *Lecture Notes in Mathematics,* 960, 1982.
5. JAMESON, A. Transonic flow calculations, Princeton University, Report MAE 1751, 1984.
6. JAMESON, A. Time-dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, AIAA Paper 91-1596, 1991.
7. NI, R.H. A multiple-grid scheme for solving the Euler equations, *AIAA J*, 1982, **20,** (11), pp 1565-1571.
8. ARNONE, A., LIOU, M-S. and POVINELLI, L.A. Multigrid time-accurate integration of Navier-Stokes equations, AIAA Paper 93-3361, 1993.
9. KROLL, N. Computation of the flow fields of propellers and hovering rotors using Euler equations, Paper 28, 12th European Rotorcraft Forum, Garmisch-Partenkirchen, Germany, September 1986.
10. BONIFACE, J.C. and SIDES, J. Numerical simulation of steady and unsteady Euler flows around multibladed helicopter rotors, Paper C10, 19th European Rotorcraft Forum, Cernobbio (Como), Italy, September 1993.
11. PAHLKE, K. and RADDATZ, J. 3D Euler methods for multibladed rotors in hover and forward flight, Paper 20, 19th European Rotorcraft Forum, Cernobbio (Como), Italy, September 1993.
12. RADDATZ, J. and PAHLKE, K. 3D Euler calculations of multibladed rotors in hover: Investigations of the wake capturing properties, 75th AGARD Fluid Dynamics Panel Meeting and Symposium on Aerodynamics and Aeroacoustics of Rotorcraft, Berlin, Germany, October 1994.
13. RADDATZ, J. and ROUZARD, O. Calculations of multibladed rotors in hover using 3D Euler methods of DLR and Onera, Paper 11, 21st European Rotorcraft Forum, St Petersburg, Russia, August 1995.
14. BONIFACE, J.C. and SIDES, J. Extension and improvement of existing Euler code of ONERA for multibladed rotors in hover, HELISHAPE Technical Report, 1995.
15. SANKAR, L.N., WAKE, B.E. and LEKOUDIS, S.G. Solution of the unsteady Euler equations for fixed and rotating wing configurations, *AIAA J Aircr*, April 1986, **23,** (4), pp 283-289.
16. SANKAR, L.N., WAKE, B.E. and LEKOUDIS, S.G., Computation of rotor blade flows using the Euler equations, *AIAA J of Aircr*, July 1986, **23,** (7), pp 582-588.
17. STRAWN, R.C. and BARTH, T.J. A finite-volume Euler solver for computing rotary-wing aerodynamics on unstructured meshes, In Proceedings of the 48th Annual American Helicopter Society Forum, Washington, DC, June 1992.
18. ALLEN, C.B. The effect of grid topology and density on inviscid hovering rotor solutions, *I Mech E J of Aero Eng*, Part G7, 1999.
19. SRINIVASAN, G.R., BAEDER, J.D., OBAYASHI, S. and MCCROSKEY, W.J. Flowfield of a lifting rotor in hover. A Navier-Stokes simulation, *AIAA J Aircr*, October 1992, **30,** (10), pp 2371-2377.
20. WAKE, B.E. and EGOLF, T.A. Implementation of a rotary-wing Navier-Stokes solver on a massively parallel computer, *AIAA J Aircr*, January 1991, **29,** (1), pp 58-67.
21. ZHONG, B. and QIN, N. Non-inertial multiblock Navier-Stokes calculation for hovering rotor flowfields using high order upwind scheme, Proceedings Royal Aeronautical Society Aerodynamics Conference, London, April 2000.
22. VAN-LEER, B. Flux-vector splitting for the Euler equations, *Lecture Notes in Physics,* 1982, **170,** pp 507-512.
23. ALLEN, C.B. Multigrid acceleration of an upwind Euler code for hovering rotor flows, *Aeronaut J*, September 2001, **105,** (1051), pp 517-524.
24. ALLEN, C.B. Multigrid convergence of inviscid fixed- and rotary-wing flows, *Int J for Numerical Methods in Fluids*, 2002, **39,** (2), pp 121-140.
25. ALLEN, C.B. Time-stepping and grid effects on convergence of inviscid rotor flows, AIAA paper 2002-2817, proceedings 20th Applied Aerodynamics Conference, St Louis, June 2002.
26. ALLEN, C.B. Convergence of steady and unsteady inviscid hovering rotor solutions, *Int J for Numerical Methods in Fluids*, 2003, **41,** (9), pp 931-949.
27. PARPIA, I. H. Van-Leer flux-vector splitting in moving coordinates, *AIAA J,* January 1988, **26,** pp 113-115.
28. ANDERSON, W.K., THOMAS, J.L. and VAN-LEER, B., Comparison of finite volume flux vector splittings for the Euler equation, *AIAA J*, September 1986, **24,** pp 1453-1460.
29. JAMESON, A., SCHMIDT, W. and TURKEL, E. Numerical solution of the Euler equations by finite-volume methods using Runge-Kutta time-stepping schemes', AIAA Paper 81-1259, 1981.
30. TURKEL, E. and VAN-LEER, B. Runge-Kutta methods for partial differential equations, ICASE Report, 1983.
31. ALLEN, C.B. 'CHIMERA volume grid generation within the EROS code, *I Mech E J Aerosp Eng*, Part G, 2000.
32. GORDON, W.J. and HALL, C.A. Construction of curvilinear coordinate systems and applications of mesh generation, *Int J Numerical Methods in Eng*, **7,** 1973, pp 461-477.
33. THOMPSON, J.F. A general three dimensional elliptic grid generation system on a composite block-structure, *Computer Methods in Appl Mech and Eng,* **64,** 1987, pp 377-411.
34. CARADONNA, F.X. and TUNG, C. Experimental and analytical studies of a model helicopter rotor in hover, NASA TM-81232, September 1981.
35. GRASSO, F. and MARINI, M. Multigrid techniques for hypersonic viscous flows, *AIAA J*, 1993, **31,** pp 1729-1731.
36. GRASSO, F. and MARINI, M. Solutions of hypersonic viscous flows with total variation diminishing multigrid techniques, *Computers and Fluids,* 1995, **24,** pp 571-592.
37. DICK, E. Multigrid formulation of polynomial flux-difference splitting for steady Euler equations, *JCP,* 1990, **91,** pp 161-173.
38. DICK, E. Multigrid methods for steady Euler and Navier-Stokes equations based on polynomial flux-difference splitting, In *Multigrid Methods II*, Birkhauser Verlag, 1-20, 1991.
39. SATOFUKA, N. and MORINISHI, K. Method of lines approach to CFD using rational Runge-Kutta time-stepping scheme, In *Frontiers of Computational Fluid Dynamics,* CAUGHEY and HAFEZ (Eds), 1995, pp 189-213.
40. MAVRIPLIS, D.J. Three-dimensional multigrid for the Euler equations, *AIAA J,* 1992, **30,** pp 1753-1761.

https://doi.org/10.1017/S000192400000511X Published online by Cambridge University Press