

# Stabilization of Linear Flow Solver for Turbomachinery Aeroelasticity Using Recursive Projection Method

M. S. Campobasso\* and Michael B. Giles†  
Oxford University, Oxford, England OX1 3QD, United Kingdom

The linear analysis of turbomachinery aeroelasticity relies on the assumption of small level of unsteadiness and requires the solution of both the nonlinear steady and the linear unsteady flow equations. The objective of the analysis is to compute a complex flow solution that represents the amplitude and phase of the unsteady flow perturbation for the frequency of unsteadiness of interest. The solution procedure of the linear harmonic Euler/Navier–Stokes solver of the HYDRA suite of codes consists of a preconditioned fixed-point iteration, which in some circumstances becomes numerically unstable. Previous work had already highlighted the physical origin of these numerical instabilities and demonstrated the code stabilization achieved by wrapping the core part of the linear code with a Generalized Minimal Residual (GMRES) solver. The implementation and the use of an alternative algorithm, namely, the Recursive Projection Method, is summarized. This solver is shown to be well suited for both stabilizing the fixed-point iteration and improving its convergence rate in the absence of numerical instabilities. In the framework of the linear analysis of turbomachinery aeroelasticity, this method can be computationally competitive with the GMRES approach.

## I. Introduction

THE aeroelastic phenomena of concern in the turbomachinery industry are blade flutter and forced response,<sup>1</sup> as they can both lead to dramatic mechanical failures if not properly accounted for in the design of the engine. The blades of an assembly can undergo flutter vibrations when the aerodynamic damping associated with certain flow regimes becomes negative and is not counterbalanced by the mechanical damping. In such circumstances, the free vibration of the blades triggered by any temporary perturbation is sustained through the energy fed into the structure by the unsteady aerodynamic forces.<sup>2</sup> The high cycle fatigue (HCF) caused by these vibrations can shorten the life of the blades below the target life of the engine. Blade forced response can also lead to HCF and is caused by the relative motion of adjacent frames of reference, which transforms steady circumferential variations of the flowfield in one frame into periodic time-varying forces acting on the blades in the other. Well-known examples include forcing caused by the wakes shed by an upstream blade row<sup>3</sup> and circumferential nonuniformities produced by distressed upstream vanes.<sup>4</sup>

The estimation of both the mean energy flux between fluid and structure in the flutter case and the unsteady forces acting on the blades in the forced response problem requires knowledge of the unsteady flowfield. Over the past two decades, a number of approaches have emerged to carry out the analysis of turbomachinery aeroacoustics and aeroelasticity.<sup>5</sup> These methods vary from uncoupled linearized potential flow solvers in which the structural equations are solved independently of the aerodynamics<sup>6,7</sup> to fully coupled nonlinear three-dimensional unsteady viscous methods in which the structural and aerodynamic time-dependent equations are solved simultaneously.<sup>8</sup> Within this range the uncoupled linear harmonic Euler and Navier–Stokes (NS) methods<sup>9–12</sup> have proved to be a successful compromise between accuracy and cost. This method views the aerodynamic unsteadiness as a small perturbation of the

space-periodic mean steady flow. Hence the unsteady flowfield can be linearized about it and because of linearity can be decomposed into a sum of harmonic terms, each of which can be computed independently. The cyclic periodicity of both the steady and unsteady flow leads to a great reduction of computational costs because the analysis can focus on one blade passage rather than the whole blade row by making use of suitable periodic boundary conditions. The assumption of small amplitude of the aerodynamic unsteadiness often allows one to neglect both the coupling and variations of structural eigenmodes caused by the aerodynamic forces.<sup>1</sup> Therefore the investigation can be carried out considering one structural mode at a time, determined by a finite element program and used as an input for calculating the unsteady aerodynamic forces. The complete aerodynamic analysis consists of two phases: 1) calculation of the nonlinear steady flowfield about which the linearization is performed and 2) solution of the linear harmonic equations.

The HYDRA suite of parallel codes<sup>13–16</sup> includes both a nonlinear (*hyd*) and a linear harmonic (*hymlin*) Euler/NS solver. The solution procedure for both *hyd* and *hymlin* can be viewed as a preconditioned fixed-point iteration. Usually the linear code converges without difficulty, but problems have been encountered in situations in which the mean flow calculation itself failed to converge to a steady state but instead finished in a low-level limit cycle, often related to some physical phenomenon such as vortex shedding at a blunt trailing edge, unsteady shock/boundary layer, or shock/wake interaction. In these circumstances the linear fixed-point iteration on which *hymlin* is based becomes unstable, leading to an exponential growth of the residuals. The relationship between these instabilities and the physical features of the underlying base flow is discussed in Ref. 17, which also summarizes the successful implementation of a Generalized Minimal Residual (GMRES) algorithm<sup>18</sup> aimed at retrieving the numerical stability of the linear code. For large three-dimensional problems, however, the restarted GMRES solver can become computationally too expensive if the number of Krylov vectors per restarted cycle needed to prevent the residual from stagnating becomes larger than 30. To overcome this problem, an alternative algorithm has been implemented in *hymlin*, namely, the Recursive Projection Method (RPM).<sup>19</sup> The main objectives of this paper are to 1) summarize the main features of this algorithm and its implementation in *hymlin* and 2) compare the numerical performance of the RPM and GMRES stabilized iterations. Section II presents an overview of the steady nonlinear and unsteady linear equations, whereas the RPM solver is discussed in Sec. III. Finally, the numerical performance of the RPM and GMRES algorithms are compared in Sec. IV, in which the two methods are applied to the

Received 21 March 2003; revision received 13 February 2004; accepted for publication 17 April 2004. Copyright © 2004 by M. S. Campobasso and Michael B. Giles. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/04 \$10.00 in correspondence with the CCC.

\*Research Officer, Computing Laboratory, Parks Road.

†Professor of Computational Fluid Dynamics, Computing Laboratory, Parks Road. Lifetime Member AIAA.

flutter analysis of a two-dimensional turbine section and of a civil engine fan.

## II. Linear Analysis of Flow Unsteadiness

The time-dependent Euler and Reynolds-averaged NS equations in conservative form are approximated on unstructured hybrid grids, using an edge-based discretization.<sup>20</sup> Considering the computational domain consisting of all of the passages of a blade row leads to a system of nonlinear ordinary differential equations (ODE) of the form

$$T \frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}, \mathbf{U}_b, \mathbf{X}, \dot{\mathbf{X}}) = 0 \quad (1)$$

where  $t$  is the physical time,  $T$  is the Jacobian of the transformation from primitive to conservative variables,  $\mathbf{U}$  is the vector of primitive flow variables,  $\mathbf{R}$  is the nodal residual, and  $\mathbf{X}$  and  $\dot{\mathbf{X}}$  are the vectors of nodal coordinates and velocities, respectively. The vector  $\mathbf{U}_b$  is used to enforce time-dependent disturbances at the inflow and outflow boundaries such as wakes shed by an upstream blade row, and the residual vector  $\mathbf{R}$  depends also on the nodal velocities  $\dot{\mathbf{X}}$  because the grid can deform conforming to the blade vibration.

The first stage of the linear analysis requires the computation of the mean steady flow about which the linearization of the unsteady terms will be carried out. Time-averaging the governing equation (1) yields

$$\mathbf{R}(\bar{\mathbf{U}}, \bar{\mathbf{X}}) = 0 \quad (2)$$

where  $\mathbf{X}$  is the vector of nodal coordinates and the bar overlining  $\mathbf{U}$  and  $\mathbf{X}$  denotes time-averaged quantities. The mean steady flow  $\bar{\mathbf{U}}$  is obtained by solving Eq. (2) for a single blade passage, because the mean flow is circumferentially periodic. The boundary conditions to which the system (2) is subject can be of three types: inflow/outflow, periodic and inviscid/viscous wall. The far-field boundaries are handled through fluxes that incorporate prescribed flow information, and thus they become part of the residual vector  $\mathbf{R}$ . At matching pairs of periodic nodes, the periodicity condition for linear cascades is enforced by setting the flow state on the upper boundary equal to that on its lower counterpart. In the case of annular domains because of the use of Cartesian coordinates, the velocity vectors on the upper boundary are obtained by rotating those on the lower one. Combining flux residuals at the two periodic nodes in a suitable manner to maintain periodicity, this boundary condition can also be included in the definition of the flux residual vector  $\mathbf{R}$ . The treatment of the wall boundaries introduces some additional terms in Eq. (2). These terms are not reported here for brevity, and the interested reader is referred to Refs. 17 and 21 for more details. The discrete equation (2) is then solved using Runge–Kutta time-marching accelerated by Jacobi preconditioning and multigrid.<sup>20</sup>

The second stage of the analysis is the linearization of the unsteady flow equations. Assuming that the flow unsteadiness is small, the time-dependent variables can be written as the sum of a mean steady part and a small-amplitude perturbation:

$$\begin{aligned} \mathbf{X}(t) &= \bar{\mathbf{X}} + \tilde{\mathbf{x}}(t), & \|\tilde{\mathbf{x}}\| &\ll \|\bar{\mathbf{X}}\| \\ \mathbf{U}_b(t) &= \bar{\mathbf{U}}_b + \tilde{\mathbf{u}}_b(t), & \|\tilde{\mathbf{u}}_b\| &\ll \|\bar{\mathbf{U}}_b\| \\ \mathbf{U}(t) &= \bar{\mathbf{U}} + \tilde{\mathbf{u}}(t), & \|\tilde{\mathbf{u}}\| &\ll \|\bar{\mathbf{U}}\| \end{aligned}$$

where the perturbations are overlined with a tilde symbol. Linearizing Eq. (1) about the mean steady conditions ( $\bar{\mathbf{X}}, \bar{\mathbf{U}}$ ) yields

$$T \frac{d\tilde{\mathbf{u}}}{dt} + L\tilde{\mathbf{u}} = \tilde{\mathbf{f}}_1 + \tilde{\mathbf{f}}_2 \quad (3)$$

where the linearization matrix  $L$  and the vectors  $\tilde{\mathbf{f}}_1$  and  $\tilde{\mathbf{f}}_2$  are given by

$$L = \frac{\partial \mathbf{R}}{\partial \mathbf{U}}, \quad \tilde{\mathbf{f}}_1 = -\left( \frac{\partial \mathbf{R}}{\partial \bar{\mathbf{X}}} \bar{\mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \bar{\mathbf{X}}} \bar{\dot{\mathbf{x}}} \right), \quad \tilde{\mathbf{f}}_2 = -\frac{\partial \mathbf{R}}{\partial \mathbf{U}_b} \tilde{\mathbf{u}}_b$$

Because of linearity, the linear unsteady flowfield can be decomposed into a sum of complex harmonics of the form  $\tilde{\mathbf{u}}_k(t) =$

$\Re(e^{ik\omega t} \hat{\mathbf{u}}_k)$ , each of which can be computed separately. The complex elements of  $\hat{\mathbf{u}}_k$  define the amplitude and phase of the unsteadiness at frequency  $k\omega$ . Analogous expansions hold for  $\tilde{\mathbf{x}}(t)$ ,  $\tilde{\dot{\mathbf{x}}}(t)$ , and  $\tilde{\mathbf{u}}_b(t)$ . Inserting them in Eq. (3) and considering only the mode  $k = 1$  for simplicity yields the harmonic equation

$$(i\omega T + L)\hat{\mathbf{u}} = \hat{\mathbf{f}}_1 + \hat{\mathbf{f}}_2 \quad (4)$$

which are complex and can be viewed as the frequency-domain counterpart of Eq. (3). The right-hand-side vectors  $\hat{\mathbf{f}}_1$  and  $\hat{\mathbf{f}}_2$  give the sensitivity of the residuals to harmonic deformations of the mesh and to incoming harmonic perturbations respectively. Based on an idea of Ni and Sisto,<sup>22</sup> the linear equations are solved with the same pseudo-time-marching approach adopted for the solution of the nonlinear steady equations, that is, by introducing a fictitious time derivative  $d\hat{\mathbf{u}}/d\tau$  and time marching the solution of the system of linear ODEs:

$$\frac{d\hat{\mathbf{u}}}{d\tau} = -[(i\omega T + L)\hat{\mathbf{u}} - \hat{\mathbf{f}}_1 - \hat{\mathbf{f}}_2]$$

until  $d\hat{\mathbf{u}}/d\tau$  vanishes. Discretizing this time derivative leads to the linear fixed-point iteration discussed in greater detail in the following section.

In the flutter case, the object of the analysis is to assess the stability of a particular structural mode. The frequency  $\omega$  and the blade mode shape are calculated with a finite element program and used to determine  $\hat{\mathbf{f}}_1$ , which is nonzero throughout the computational domain because the grid deforms conforming to the harmonic vibration of the blade, whereas  $\hat{\mathbf{f}}_2$  is set to zero. The phase between the motion of adjacent blades (interblade phase angle or IBPA) is an additional parameter of the analysis. It is given by  $\phi_j = 2\pi j/N_{\text{blades}}$ , and the index  $j$  usually called *nodal diameter* can take any integer value between 0 and  $(N_{\text{blades}} - 1)$ , though the critical values are usually the first few ones, as shown in Ref. 1. Equation (4) can then be solved for a single passage, introducing the complex phase shift  $e^{i\phi_j}$  between the two periodic boundaries. The output of interest is the net energy flux from the structure to the working fluid over one cycle of vibration, defined by the worksum integral

$$W = \int_0^{T_v} \int_S p u_{\text{blade}} \cdot d\mathbf{S} dt$$

in which  $T_v$  is the period of vibration,  $p$  and  $u_{\text{blade}}$  are the time-dependent blade static pressure and velocity respectively,  $d\mathbf{S}$  is the elemental blade surface with outward normal, and  $S$  is the overall blade surface. A positive sign indicates stability as energy is transferred from the structure to the fluid, whereas a negative sign indicates the occurrence of flutter. In the engineering community, the logarithmic decrement  $\delta$  is a more frequently used stability parameter, which depends on the ratio between the amplitude  $V$  of two consecutive cycles of vibration. It is defined as  $\delta = V(t + T_v)/V(t)$ , and it can be proved that

$$\delta = W/\omega^2$$

In forced response, the object of the analysis is to determine the unsteady forces acting on the blade as a result of any of the harmonic components, into which the incoming time-periodic gust can be decomposed. The IBPA depends on the geometric properties of the problem. In the case of forcing coming from circumferentially periodic wakes, the blades and the wakes can have different pitches, and hence there is a difference in the times at which neighboring wakes strike neighboring blades. Therefore the IBPA of the fundamental harmonic is  $2\pi N_{\text{wakes}}/N_{\text{blades}}$ . Again the linear harmonic equation (4) can be solved for a single blade passage using complex periodic boundary conditions. The vector  $\hat{\mathbf{f}}_1$  is zero throughout the domain because the mesh is stationary, and the vector  $\hat{\mathbf{f}}_2$  is nonzero only at the inlet or outlet boundaries, where the harmonic perturbation is prescribed. The unsteady aerodynamic force acting on the blade can be calculated in a postprocessing step for each structural mode using the unsteady pressure field determined with the harmonic analysis.

The linear unsteady analysis is completed by enforcing suitable linearized boundary conditions. The inflow, outflow, and (complex) periodic boundary conditions can all be symbolically included into Eq. (4), whereas the additional terms as a result of the wall boundary condition are omitted here and reported in Refs. 17 and 21. The implementation of the far-field boundary conditions is based on one-dimensional nonreflecting boundary conditions.<sup>23</sup> Equation (4) are then solved using the same preconditioned pseudo-time-marching method as for the nonlinear equations.

### III. RPM Stabilization

The linearized harmonic flow equation (4) can be viewed as a simple linear system of the form

$$Ax = b \quad (5)$$

with  $A = i\omega T + L$ ,  $b = \hat{f}_1 + \hat{f}_2$ , and  $x = \hat{u}$ . This system has dimension  $k = (2 \times N_{\text{eqs}} \times N)$ , where  $N$  is the number of grid nodes,  $N_{\text{eqs}}$  is 5 for inviscid flows and 6 for turbulent flow analyses using a one-equation turbulence model, and the factor 2 accounts for real and imaginary part of the complex flowfield. Though Eq. (4) is complex, *hydlin* has been written using real arithmetic, that is, considering real vectors of size  $k$  rather than complex vectors of size  $k/2$ . This choice has been made because of errors often introduced by highly optimized FORTRAN compilers when dealing with complex arithmetic. The linear code for the solution of these equations can be regarded as the fixed-point iteration:

$$x_{n+1} = F(x_n) = (I - M^{-1}A)x_n + M^{-1}b \quad (6)$$

in which  $M^{-1}$  is a preconditioning matrix resulting from the Runge–Kutta time-marching algorithm, the Jacobi preconditioner, and one multigrid cycle. Linear stability analysis of Eq. (6) shows that a necessary condition for its convergence is that all of the eigenvalues of  $(I - M^{-1}A)$  lie within the unit circle centered at the origin in the complex plane or equivalently that all of the eigenvalues of  $M^{-1}A$  lie in the unit disk centered at  $(1, 0)$ . For most aeroelastic problems of practical interest, this condition is fulfilled, and the linear code converges without difficulty. However, an exponential growth of the residual has been encountered in situations in which the steady flow calculation itself failed to converge to a steady state but instead finished in a small-amplitude limit cycle, related to some physical phenomenon such as separation bubbles, corner stalls, and vortex shedding at a blunt trailing edge. The solution procedure of the nonlinear steady equation (2) is not time accurate, but it nevertheless reflects some time-dependent physical properties of the flowfield because of the pseudo-time-marching strategy associated with the Runge–Kutta algorithm. Physical small-amplitude limit cycles do not prevent the steady solver from converging to an acceptable level, and their effect is sometimes visible in small oscillations of the residual. However these periodic instabilities result in a small number of complex conjugate pairs of eigenvalues of the linearization matrix  $M^{-1}A$  lying outside the unit circle (outliers) and thus causing the exponential growth of the residual of the linear equations. This problem had been previously solved by implementing a restarted GMRES algorithm in *hydlin*.<sup>17</sup> The drawback of this approach is that the GMRES code can become computationally very expensive when dealing with large three-dimensional problems. This is because each Krylov vector has the same size of the linear flowfield  $x$ , and the extra memory requirement with respect to the standard code grows linearly with the number of Krylov vectors ( $n_{\text{Kr}}$ ) per restarted cycle. Furthermore  $n_{\text{Kr}}$  cannot be chosen below a case-dependent threshold to prevent the residual from stagnating. The memory requirement of the GMRES code is already about twice that of the standard code if  $n_{\text{Kr}} = 30$ .

To stabilize the linear code reducing the additional memory requirement, the RPM first introduced by Shroff and Keller to stabilize unstable iterative procedures for nonlinear parameter dependent problems<sup>19</sup> has been implemented in *hydlin*. This algorithm is based on the projection of Eq. (6) onto the orthogonal subspaces  $\mathcal{P}$  and  $\mathcal{Q}$  of  $\mathcal{R}^k$  associated respectively with the subset of  $m$  outliers and that of the remaining  $(k - m)$  eigenvalues lying in the unit disk. At

each RPM iteration, only the projection of Eq. (6) onto the subspace  $\mathcal{Q}$  is solved with the standard fixed-point iteration; the projection onto the typically low-dimensional subspace  $\mathcal{P}$  is instead solved with Newton's method. Denoting by  $Z$  an orthonormal basis of  $\mathcal{P}$ , the orthogonal projectors  $P$  and  $Q$  of the subspaces  $\mathcal{P}$  and  $\mathcal{Q}$  are defined, respectively, as  $P = ZZ^T$  and  $Q = I - P$ . Each time the calculation is diverging, the basis  $Z$  is augmented with the current dominant eigenmode, and the projectors  $P$  and  $Q$  are updated accordingly. The projections  $f$  and  $g$  of Eq. (6) onto  $\mathcal{P}$  and  $\mathcal{Q}$  are defined, respectively, as

$$f = PF = P[(I - M^{-1}A)x + M^{-1}b]$$

$$g = QF = Q[(I - M^{-1}A)x + M^{-1}b]$$

and the stabilized iteration can be written as follows:

$$p_{v+1} = p_v + (I - f_p)^{-1}[f(p_v, q_v) - p_v] \quad (7a)$$

$$q_{v+1} = g(p_v, q_v) \quad (7b)$$

where

$$p = Px, \quad q = Qx, \quad f_p \equiv PF_xP, \quad F_x = I - M^{-1}A$$

It is easily verified that

$$(I - f_p)^{-1} = Z[I - Z^T(I - M^{-1}A)Z]^{-1}Z^T = Z[I - H]^{-1}Z^T \quad (8)$$

where  $(I - H)$  is a small matrix of size  $m$ , whose inversion requires minimum computational effort. The stability analysis of this algorithm shows that its spectral radius is smaller than 1, that is, the stabilized RPM iteration is stable.<sup>19</sup>

The basis  $Z$  is updated directly from the iterates  $q_v$  of the modified iteration (7b), without computing Jacobians. This is done by monitoring the rate of convergence of the iterates  $q_v$ . If the residual starts growing, it is argued that some of the eigenvalues of  $g_q = QF_xQ$  lie outside the unit circle. The generic case is that either an isolated real eigenvalue  $\lambda_{m+1}$  or a complex conjugate pair  $(\lambda_{m+1}, \lambda_{m+2})$  cause the instability. One has to decide which is the case and determine the one or two vectors to append to  $Z$  to make it span the larger invariant subspace of  $F_x$  associated with the augmented set of eigenvalues  $\{\lambda_1, \lambda_2, \dots, \lambda_{m+1}\}$  or  $\{\lambda_1, \lambda_2, \dots, \lambda_{m+1}, \lambda_{m+2}\}$ . It can be shown<sup>19</sup> that the vectors  $\{\Delta q_v = q_{v+1} - q_v\}$  are a power iteration with the matrix  $g_q$  applied to the starting vector  $\Delta q_0$ . Asymptotically these vectors will tend to be in the dominant eigenspace of  $g_q$ , provided that  $\Delta q_0$  has a nonzero component in this direction. If the iteration starts diverging, the two difference vectors  $\{\Delta q_v, \Delta q_{v-1}\}$  (i.e., the most recent power iterates) are used to compute the Gram–Schmidt factorization<sup>24</sup>

$$D \equiv [\Delta q_v, \Delta q_{v-1}] = \hat{D}T \quad (9)$$

with  $T \in \mathcal{R}^{2 \times 2}$  upper triangular and  $\hat{D} \in \mathcal{R}^{k \times 2}$  orthogonal. If  $T_{1,1} \gg T_{2,2}$ , the dominant eigenmode of  $g_q$  is real, and only the first column of  $\hat{D}$  is included in  $Z$ . Otherwise the instability is caused by a complex conjugate pair, and both columns of  $\hat{D}$  are included in  $Z$ . This procedure can be used to append to  $Z$  not only the unstable eigenmodes, but also the stable ones whose eigenvalues are very close to the unit circle, allowing one to speed up the convergence rate even in the absence of outliers. In the current implementation, we determine recursively and include in  $Z$  one real eigenmode or one complex conjugate eigenpair at a time, until a satisfactory convergence rate is obtained. However one could also scrutinize more than one dominant mode at a time, carrying out the QR factorization (9) with more than two  $\Delta q$  looking at the modulus of the eigenvalues of  $T$  and including in  $Z$  all of the modes needed for a satisfactory convergence rate of the stabilized iteration. The experience gained so far makes us believe that this approach would require fewer iterations to achieve the desired level of convergence. Looking for one dominant mode at a time, in fact, can result in a longer numerical transient, as the RPM iteration continues to diverge until all outliers have been included in  $\mathcal{P}$ . On the other hand, the single-mode search results in a reduced memory usage, as the memory for storing new

vectors of  $Z$  can be allocated “dynamically” when the convergence rate needs to be improved.

In the actual computation, one introduces coordinate variables  $\mathbf{z} \in \mathcal{R}^{m \times 1}$  for the representation of  $\mathbf{p}$  in the basis  $Z$ :

$$\mathbf{z} \equiv Z^T \mathbf{p} = Z^T \mathbf{x}$$

from which it follows that

$$\mathbf{p} = Z\mathbf{z}, \quad \mathbf{x} = Z\mathbf{z} + \mathbf{q}$$

The iteration (7a) in the subspace  $\mathcal{P}$  can be written in these variables using Eq. (8) and observing that  $Z^T Z = I$ :

$$\mathbf{z}_{v+1} = \mathbf{z}_v + (I - H)^{-1} [Z^T \mathbf{F}(\mathbf{x}_v) - \mathbf{z}_v]$$

The implementation of the RPM solver in *hydlin* has been carried out at the top routine level. At this level, the pseudocode of the preconditioned multigrid iteration without RPM looks like

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_{\text{start}} \\ \mathbf{x} &= mg(A, \mathbf{x}, \mathbf{b}, n_{\text{cl}}) \\ \mathbf{x}_{\text{finish}} &= \mathbf{x} \end{aligned}$$

where  $mg$  is the core routine that performs the preconditioned fixed-point iteration (6) and  $\mathbf{x}_{\text{finish}}$  is the solution after  $n_{\text{cl}}$  multigrid cycles. The RPM solver does not require any change to  $mg$ , which is still used as in the standard fixed-point iteration to determine the projection  $\mathbf{q}$  of the solution  $\mathbf{x}$  onto  $\mathcal{Q}$ . This operation corresponds to Eq. (7b). The computationally cheap inversion of the matrix  $(I - H)$  along with the other matrix-vector products is carried out at the top routine level and the pseudocode of the main *hydlin* using RPM is

$$Z = [], \quad m = 0, \quad v = 0, \quad \mathbf{x}_v = \mathbf{x}_{\text{start}}$$

while  $|\mathbf{b} - A\mathbf{x}_v| > \text{tolerance}$

% stabilized iteration

if  $m > 0$

$$\mathbf{z}_v = Z^T \mathbf{x}_v, \quad \zeta_v = Z^T \mathbf{x}_{v+1}$$

$$\mathbf{z}_{v+1} = \mathbf{z}_v + (I - H)^{-1} (\zeta_v - \mathbf{z}_v)$$

$$\mathbf{q}_{v+1} = \mathbf{x}_{v+1} - Z\mathbf{z}_v$$

$$\mathbf{x}_v = Z\mathbf{z}_{v+1} + \mathbf{q}_{v+1}$$

endif

$$\mathbf{x}_{v+1} = mg(A, \mathbf{x}_v, \mathbf{b}, 1)$$

$$v = v + 1$$

% increase basis size

if not converging

$T$  and  $\hat{D}$  from equation (9)

$$Z = [Z \hat{D}(:, 1 : \delta m)], \quad m = m + \delta m$$

% update  $H$

for  $j = 1 : m$

$$H(:, j) = Z^T mg(A, Z(:, j), \mathbf{0}, 1)$$

end

$$v = 0$$

endif

end

$$\mathbf{x}_{\text{finish}} = \mathbf{q}_{v+1} + Z\mathbf{z}_{v+1}$$

The section of the pseudocode labeled with “stabilized iteration” corresponds to the stabilized iteration (7). Only one multigrid cycle is performed at each step of the stabilized iteration by setting  $n_{\text{cl}} = 1$  in the arguments of  $mg$ . The variable  $\delta m$  in the section “increase basis size” depends on the outcome of the QR factorization (9): it is 1 if the dominant mode is real and 2 if it is a complex conjugate pair. Thereafter the subroutine  $mg$  is used to determine the columns of  $F_x Z$  by setting  $\mathbf{b} = \mathbf{0}$  and performing one multigrid cycle on each column of  $Z$ . The additional CPU time of the RPM iteration is that required for the matrix-vector products of the stabilized iteration and for calling  $mg$  to update  $H$ . The order of magnitude of  $m$  based on the problems investigated so far varies between 1 and 10, and both operations are fairly inexpensive for values of  $m$  in this range. Including this extra cost in that required for a multigrid cycle, the CPU time for executing a given number of multigrid cycles using RPM is about 1% higher than using the standard iteration. In the absence of unstable or slowly converging modes, the algorithm reduces to the standard iteration (6), as at each step one simply performs one multigrid cycle using the appropriate vector  $\mathbf{b}$  and  $n_{\text{cl}} = 1$  in the arguments of  $mg$ .

Each column of  $Z$  has the same size of the linear flowfield  $\mathbf{x}$ , and the implementation of RPM in *hydlin* has led to an extra memory allocation of  $[(m + 1) \times k]$  needed for the vectors of the basis  $Z$  and for a new work array. By comparison, the extra memory allocation required by GMRES is  $[(n_{\text{Kr}} + 1) \times k]$  and is independent of the number of outliers.

The RPM algorithm also allows the straightforward determination of the dominant eigenmodes, as they are isolated in the small eigenspace  $\mathcal{P}$ . Similarly to the GMRES case,<sup>17</sup> this enables one to relate the numerical instability to the physical unsteadiness that causes it. To establish the relationship between the dominant modes and the vectors of  $Z$ , let us start by considering the restriction of  $F_x$  to  $\mathcal{P}$  defined as

$$f_p \equiv P F_x P = P(I - M^{-1}A)P$$

The eigenvalues of  $f_p$  are given by

$$P(I - M^{-1}A)PV = V\Theta \quad (10)$$

where  $\Theta$  is the diagonal matrix of the eigenvalues and  $V$  the matrix whose columns are the eigenvectors of  $f_p$ . The eigenvalues of the dominant modes are the first  $m$  diagonal entries of  $\Theta$ , and the associated eigenvectors are the first  $m$  columns of  $V$ . Note that the remaining  $k - m$  diagonal entries of  $\Theta$  and columns of  $V$  are zero, as  $f_p$  has rank  $m$ . Inserting the expression  $P = ZZ^T$  in Eq. (10), it then follows that

$$HY = Y\Theta'$$

in which  $\Theta'$  is the upper-left portion of  $\Theta$  of size  $m \times m$  and the columns of the matrix  $Y = Z^T V$  are the right eigenvectors of  $H$ . The sought dominant modes are given by

$$V = ZY \quad (11)$$

because  $ZZ^T V = PV = V$ . The relationship between the dominant eigenvalues  $\lambda_j$  of  $M^{-1}A$  and the eigenvalues  $\lambda'_j$  of  $H$  (i.e., the diagonal entries of  $\Theta'$ ) is

$$\lambda_j = 1 - \lambda'_j, \quad j = 1, \dots, m$$

#### IV. Results

In this section, the investigation of the flutter stability of a two- and three-dimensional turbomachinery test case is carried out by means of the RPM-stabilized linear flow solver to highlight the main features of the algorithm. A comparative study of the numerical performance of RPM and GMRES is also presented. The interested reader is referred to Refs. 13, 14, 17, and 25 for other flow analyses aiming at the validation of both the nonlinear and linear code.

### A. Two-Dimensional Turbine Section

The two-dimensional turbine section of the 11th Standard Configuration is the midspan blade-to-blade section of an annular turbine cascade with 20 blades. The test rig and cascade geometry are described in Ref. 26, which also provides experimental measurements and various computed results of the steady and unsteady flow field caused by blade-plunging with prescribed IBPA. Two steady working conditions are considered: a subsonic one with exit Mach number of 0.68 and a transonic one with exit Mach number of 0.96. This test case had already been used to demonstrate the predictive capabilities of *hydlin* and test the GMRES solver.<sup>17</sup> In this paper,

it will be used to demonstrate the effectiveness of the RPM solver. The computational grid that we have used for the investigation is a quasi-orthogonal H-type mesh with medium refinement: it has 273 nodes in the streamwise and 65 nodes in the pitchwise direction, for a total of 17,745 grid nodes. A preliminary mesh-refinement analysis carried out using a coarser 7869-node ( $183 \times 43$ ) and finer 39673-node ( $409 \times 97$ ) mesh has shown no difference of practical interest between the results obtained with the medium and finer grids. The coarser mesh is shown in Fig. 1. The Mach contours in Fig. 2 refer to the transonic mean flow. They reveal the presence of a separation bubble on the suction side close to the leading edge and

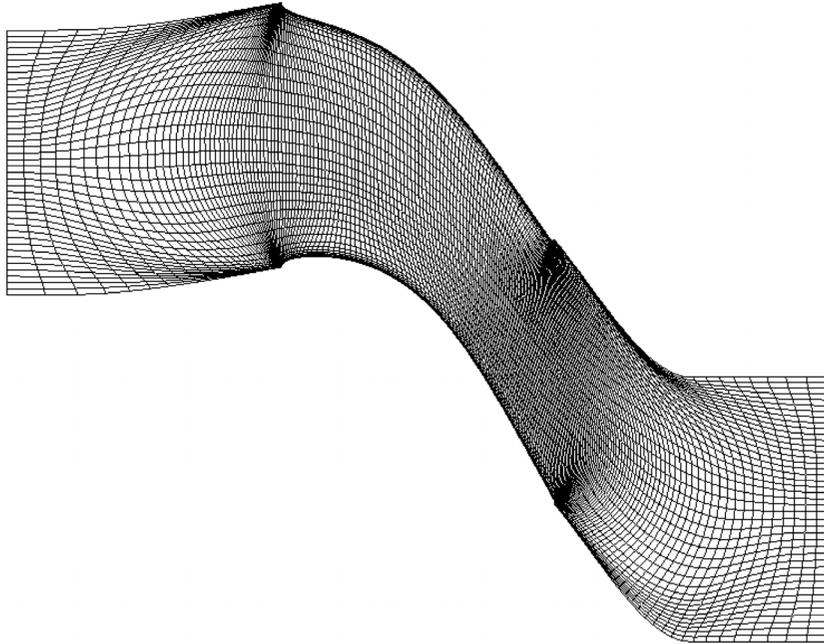


Fig. 1 Mesh for the two-dimensional turbine section.

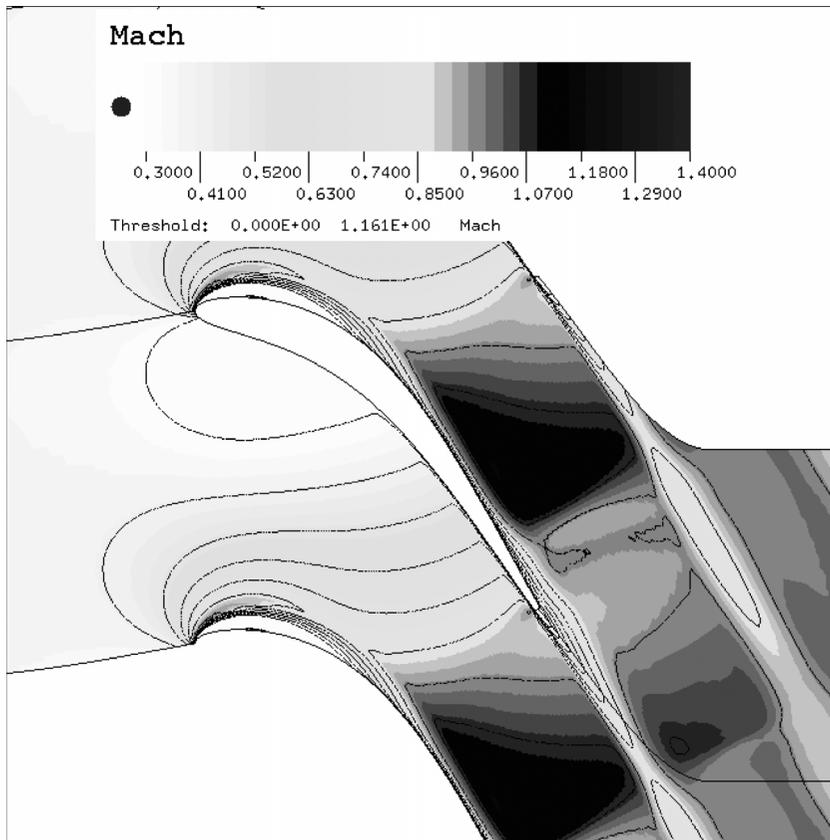
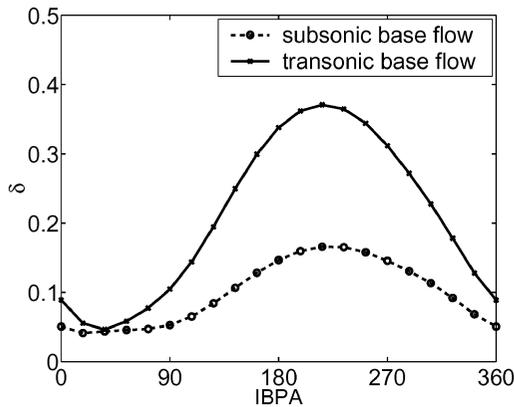


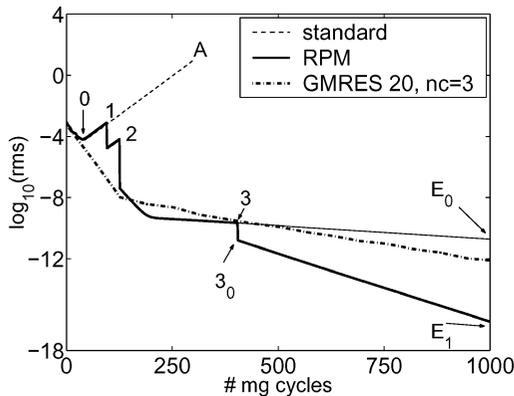
Fig. 2 Mach contours for transonic conditions of the two-dimensional turbine.

a shock, impinging on the suction side close to the trailing edge and crossing the wake shed by the upper blade. Note also that both the wake and the boundary layer on the suction side thicken remarkably after passing through the shock.

The stability curves ( $\delta$  vs IBPA) for both flow regimes are provided in Fig. 3a, which shows that the system never becomes aeroelastically unstable. The nonlinear calculations of both the subsonic and transonic base flow converge without difficulties to machine



a)



b)

Fig. 3 Flutter analysis of the two-dimensional turbine: a) logarithmic decrement vs IBPA and b) convergence histories using the standard, RPM, and GMRES iteration (transonic base flow, IBPA = 180 deg).

epsilon ( $10^{-18}$ ). However all of the linear calculations based on the transonic base flow diverge using the standard code, and convergence can be retrieved only by using RPM or GMRES, as shown in the convergence histories of *hydlin* in Fig. 3b, which refer to IBPA = 180 deg. The variable on the  $x$  axis is the number of multigrid cycles and that on the  $y$  axis is the logarithm in base 10 of the rms of all nodal residuals ( $rms$ ). The number at the right of the label GMRES in the legend is  $n_{Kr}$ , whereas  $n_{cl}$  is the number of multigrid cycles per GMRES step. The exponential growth of the residual using the standard iteration and the discontinuities in the slope of the convergence history using RPM can be explained looking at the dominant eigenmodes of  $M^{-1}A$  for  $n_{cl} = 1$ . The first 150 dominant eigenvalues determined by means of GMRES as described in Ref. 17 are plotted in the complex plane of Fig. 4, which highlights the presence of two complex conjugate pairs of outliers. These have been determined also by means of RPM following the procedure reported in the preceding section and are plotted in the same figure using empty circles. Note that there is a very good agreement between the GMRES and RPM estimates. The complex conjugate pair of outliers labeled with 1 is responsible for the exponential growth of the residual associated with the fixed-point iteration (6). In fact, its asymptotic convergence rate is determined by the spectral radius  $\rho$  of the linear operator  $M^{-1}A$ , and it can be proved that the relationship between the asymptotic slope of the residual curve and  $\rho$  is

$$\Delta[\log(rms)]/N_{mg} \approx \log \rho \quad (12)$$

where  $N_{mg}$  is the number of multigrid cycles across which the variation of  $rms$  is considered. This equation provides the theoretical relationship between the slope of the exponentially growing residual curve of the standard iteration (branch 0A in Fig. 3b) and the spectral radius of the linear operator (radius of the outlier 1). Inserting the computed values in it yields  $46.90e - 3 \approx 47.53e - 3$ , which demonstrates the correctness of the mathematical analysis. The discontinuities in the slope of the convergence history using the RPM solver labeled with 1, 2, and 3 in Fig. 3b occur when the complex conjugate eigenpairs 1, 2, and 3 in Fig. 4 are appended to the unstable eigenspace  $\mathcal{P}$ . Note that the RPM-stabilized iteration becomes stable once the eigenmodes 1 and 2 are included in  $\mathcal{P}$ , as demonstrated by the negative slope of the branch  $3E_0$ . The inclusion of the eigenmode 3 has only the effect of increasing the convergence rate, because this is controlled by the spectral radius of the projection of  $M^{-1}A$  onto the subspace  $\mathcal{Q}$ , namely, the distance of the eigenmode 4 from the center of the unit disk. In fact, inserting in Eq. (12) this spectral radius

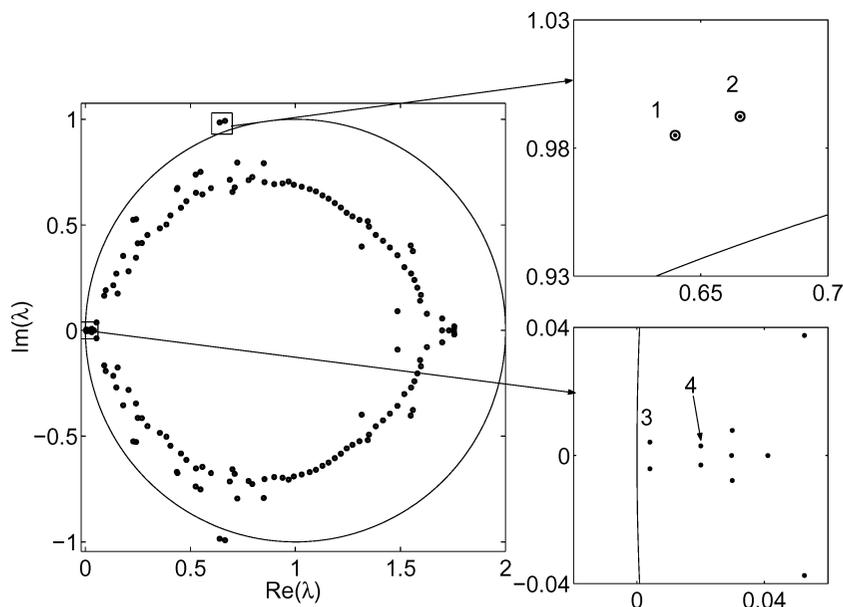
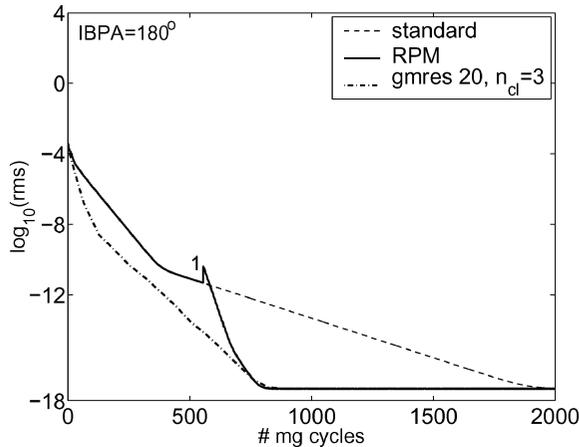


Fig. 4 Flutter analysis of the two-dimensional turbine: first 150 dominant eigenvalues of  $M^{-1}A$  ( $n_{cl} = 1$ ) determined with GMRES ( $\cdot$ ) and two pairs of complex conjugate outliers determined with RPM ( $\circ$ ) (transonic base flow, IBPA = 180 deg).

**Table 1** First three dominant eigenvalues for three different IBPAs

IBPA, deg	Mode	$\Re(\lambda)$	$\Im(\lambda)$
36	1	0.640143	0.984962
	2	0.665552	0.992178
	3	0.004075	0.004150
180	1	0.640016	0.984956
	2	0.665476	0.992322
	3	0.004076	0.004150
270	1	0.639930	0.984763
	2	0.665644	0.992444
	3	0.004075	0.004150

**Fig. 5** Flutter analysis of the two-dimensional turbine: convergence histories with standard, RPM, and GMRES iteration (subsonic base flow).

and the data relative to the slope of the branch  $3_0E_1$  in Fig. 3b yields  $-20.5e-3 \approx -19.8e-3$ . The first three dominant eigenvectors of  $M^{-1}A$  have been determined using the analysis presented at the end of the preceding section, namely, by means of Eq. (11). Such analysis has shown that the eigenmodes 1 and 2 correspond to the separation bubble on the suction side because this is the only location at which they have nonzero amplitude. This result proves that the origin of the numerical instability is the small limit cycle associated with this unstable flow separation. The eigenmode 3 has nonzero amplitude both in the separation and downstream shock regions. These conclusions are identical to those drawn on the basis of an Arnoldi-based eigenmode analysis of the same test case, discussed in Ref. 17.

It has also been found that the two dominant eigenmodes just described are independent of the IBPA, despite the fact that  $M^{-1}A$  depends on it. This phenomenon is probably caused by the high spatial localization of the unstable modes and is highlighted in Table 1, which reports the real and imaginary parts of the first three least stable modes for three different IBPAs. This feature can be easily exploited in the framework of the RPM solver, as the same basis  $Z$  can be used for linear calculations referring to different IBPAs, thus leading to significant computational savings. All linear calculations based on the subsonic flow regime converge also without RPM and GMRES. The convergence histories of the linear code (IBPA = 180 deg) using the standard iteration, RPM, and GMRES 20 are provided in Fig. 5. The memory requirements with three grid levels for the multigrid scheme are 52, 63, and 86 Mbytes, respectively. The extra memory requirement of RPM with respect to the standard code is used for the six vectors of the basis  $Z$  associated with the three dominant complex conjugate eigenpairs. All three calculations have been started from the same initial solution and run on eight processors of a computer cluster consisting of 24 four-processor Sun Ultra-80 nodes, with a Sun Blade-1000 front end. The 700 iterations for achieving a residual level of  $-17.5$  with RPM and GMRES have been carried out in about 27 min of CPU time, whereas the 1800 needed to obtain the same level with the standard iteration have required 69 min.

**Fig. 6** Blade geometry and surface mesh of the three-dimensional fan.

### B. Three-Dimensional Fan

The second test case considered is a three-dimensional fan rotor whose geometry and surface grid are shown in Fig. 6. This grid has only 157,441 nodes and is quite coarse, but all of the phenomena discussed in this section have been also observed with finer computational meshes and for other test cases. The linear flutter analysis has been carried out for four points of a constant-speed working line using *hyd* and *hydlin*. The computed pressure ratio  $\beta$  is plotted vs the computed mass flow  $\dot{m}$  in Fig. 7a. Their definition is

$$\beta = \frac{p_{02}}{p_{01}}, \quad \dot{m} = \int_{S_2} \rho_2 \mathbf{u}_2 \cdot d\mathbf{S}_2$$

All variables in the two preceding expressions refer to the base flow:  $p_{01}$  and  $p_{02}$  are, respectively, the inlet and outlet mass-averaged total pressure,  $S_2$  is the area of the outlet boundary,  $\rho_2$  is the outlet density, and  $\mathbf{u}_2$  is the outlet velocity vector. Note that the values of both  $\beta$  and  $\dot{m}$  in Fig. 7a are given as percentage deviations from their design values. For all four working conditions the residual of the nonlinear steady equations drops by four orders of magnitude (Fig. 7b), ending in a low-amplitude limit cycle.

The analysis of the flutter stability of the first flap mode has been carried out for all four working conditions, and the computed logarithmic decrement is plotted in Fig. 8a. As expected, the least stable aeroelastic modes are those associated with the first few IBPAs, and the blades undergo flutter in the 2 nodal diameter mode at the base flow conditions  $D$ , which are the closest to stall. All linear calculations have been performed using RPM or GMRES, as they were otherwise unstable. This is visible in the convergence plots of *hydlin* reported in Fig. 8b, which refer to the base flow conditions  $D$  and to IBPA = 180 deg. This figure shows that the GMRES solver stagnates if  $n_{cl} = 1$  and an acceptable convergence rate can be achieved only by using  $n_{cl} = 3$  and  $n_{Kr} \geq 30$ . As in the preceding example, both the exponential growth of the residual using the

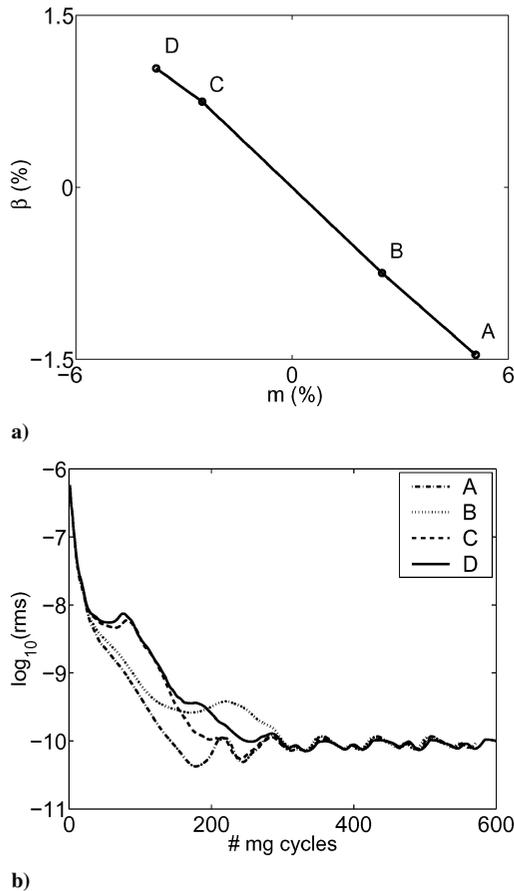


Fig. 7 Steady flow analysis of the three-dimensional fan: a) constant speed working line of the three-dimensional fan and b) convergence histories of *hyd* for four working points.

standard iteration and the discontinuities in the slope of the convergence history using RPM can be explained analyzing the dominant eigenmodes of  $M^{-1}A$  for  $n_{cl} = 1$ . The plot of the first 150 dominant eigenvalues in Fig. 9 (determined by means of GMRES) highlights the existence of four complex conjugate pairs of outliers, labeled from 1 to 4 in order of decreasing distance from the center of the unit disk in Fig. 9. The empty circles refer to the RPM estimates of the four dominant eigenpairs, and again a very good agreement with the GMRES result is observed. Inserting in Eq. (12) the computed data relative to the slope of the ascending branch 0A of the residual curve of the standard iteration (Fig. 8b) and the spectral radius of  $M^{-1}A$  (radius of the outlier 1) yields  $38.82e-3 \approx 40.17e-3$ , which confirms once more the correctness of the mathematical analysis. The discontinuities of the RPM convergence curve occurring at the iterations labeled from 1 to 4 in Fig. 8b occur when the first four dominant modes are included in the unstable eigenspace  $\mathcal{P}$ . The improvement of convergence rate at the iteration labeled with 5 occurs when also the eigenvalue 5 is appended to  $\mathcal{P}$ . In fact, the branch  $5E_1$  is steeper than the branch  $5E_0$  because their slope is determined by the eigenvalues 6 and 5 respectively, and the former is closer to the center of the unit disk. Inserting in Eq. (12) the computed data relative to the branch  $5E_1$  and the distance of the eigenvalue 6 from the center leads to  $-7.96e-3 \approx -8.16e-3$ .

Both the modal analysis based on Eq. (11) and that reported in Ref. 17 show that the eigenvectors associated with the eigenpairs 1 and 2 correspond to a mild hub corner stall, whereas those associated with the pairs 3 and 4 correspond to a separation bubble on the suction side close to the leading edge in the hub region. The numerical instabilities of the standard linear iteration are therefore due to the fact that the linearization of the unsteady equations is performed about of a base flow containing traces of these two unsteady phenomena. The eigenmode corresponding to the complex conjugate pair 5 takes nonzero values both at the same locations as the first four

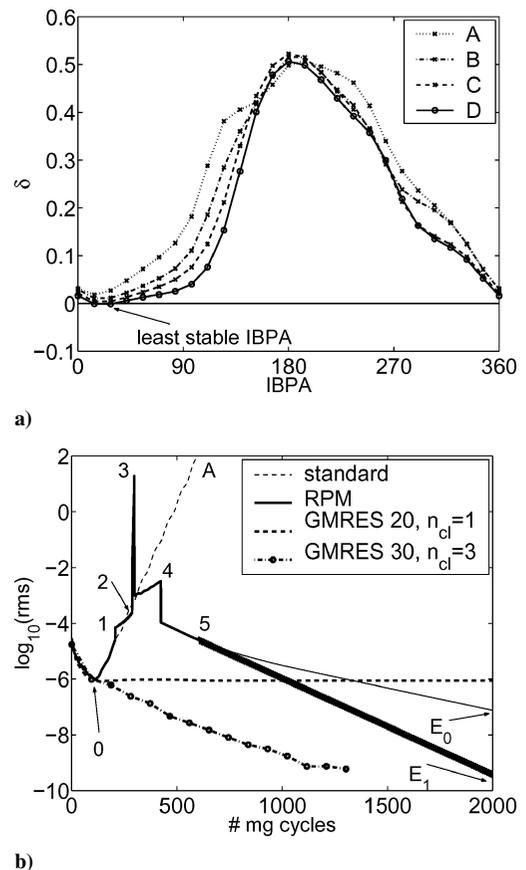


Fig. 8 Flutter analysis of the three-dimensional fan: a) logarithmic decrement for the four working conditions and b) convergence histories of *hydlin* using the standard, RPM, and GMRES iteration (mean flow D, IBPA = 180 deg).

and in the proximity of a shock on the suction side close to the blade tip. Similarly to the turbine test case, the dominant eigenmodes just described have been found to be independent of the IBPA, and this might be caused again by their high spatial localization.

Note that solving the linear equations about a pseudo-time-averaged base flow, obtained by averaging the nonlinear solution over one numerical limit cycle of *hyd*, has not removed the instability of the linear calculations using the standard iteration. It might be more appropriate to linearize the flow unsteadiness either about the stabilized solution of the nonlinear equations, determined by using GMRES<sup>27</sup> or about the true time-averaged flow. This can be obtained either by introducing unsteady stress terms in the nonlinear equations<sup>28</sup> or solving the time-dependent nonlinear equations and then time averaging the flow solution. It is the authors' view, however, that these approaches would also not remove the linear instabilities of the standard linear solver. In fact, the limit cycles under discussion are stable, and the theory of dynamical systems foresees that the underlying steady solutions are unstable<sup>29</sup> leading to the exponential growth of their linear perturbations.

All calculations have been run with four grid levels for the multigrid scheme on the SUN cluster described in the preceding subsection. The CPU time of one multigrid cycle depends on the number of iterations performed on each grid level. The values chosen for this test case have led to a CPU time of about 56 s for one multigrid cycle of *hydlin* using eight processors for both the RPM and GMRES code. (By comparison, the CPU time for one multigrid cycle of *hyd* is about half of that needed by *hydlin*.) The 800 cycles needed for a good convergence of each linear calculation have thus required an overall time of about 12 h using GMRES. The RPM code has needed about 1500 multigrid iterations to achieve the same convergence level of the GMRES code. This is because of the numerical transient of the RPM solver, during which the residual keeps growing until all outliers are included in  $\mathcal{P}$ . This transient

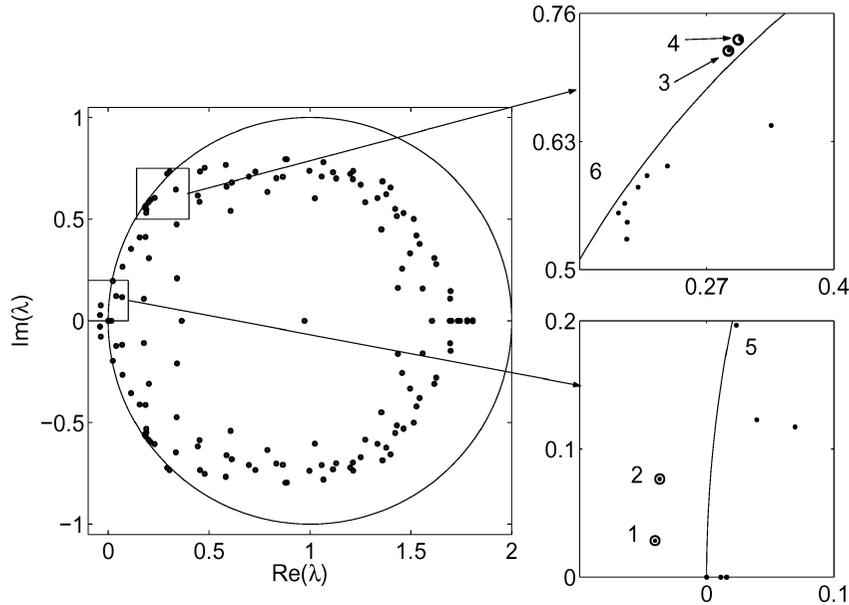


Fig. 9 Flutter analysis of the three-dimensional fan: first 150 dominant eigenvalues of  $M^{-1}A$  ( $n_{cl} = 1$ ) determined with GMRES (•) and four pairs of complex conjugate outliers determined with RPM (◦) (mean flow  $D$ , IBPA = 180 deg).

could be significantly shortened by searching more than two outliers at a time. The memory requirements of the standard, RPM, and GMRES 30 code using four grid levels are 441, 600, and 888 Mbytes, respectively. The extra memory requirement of RPM with respect to the standard code is used for the 10 vectors of the basis  $Z$  associated with the five dominant complex conjugate eigenpairs.

## V. Conclusions

This paper has presented the application of the RPM algorithm to stabilize an existing linear flow solver for the analysis of turbomachinery aeroelasticity. Such stabilization had previously been possible only by using GMRES. Similarly to the GMRES-stabilized iteration, the use of RPM allows the aeroelastic analysis to be carried out even in the presence of small periodic flow instabilities in the base flow, which are believed not to have any significant effect on the aeroelastic behaviour of the component under investigation.

The CPU time required for one multigrid cycle by both the RPM and GMRES solver is only from 1 to 3% higher than with the standard iteration. For a given set of multigrid parameters, the convergence rate of RPM depends on the spectral radius of the projection of the linear operator  $M^{-1}A$  onto the stable subspace  $\mathcal{Q}$ , and therefore a significant convergence speed up can be achieved by appending to the unstable subspace  $\mathcal{P}$ , not only the outliers but also the first few dominant modes in the unit circle. The extra memory allocation depends on the overall number of eigenmodes included in  $\mathcal{P}$ , which is greater or equal to the number of outliers. The additional memory required by a run in which 10 modes are included in  $\mathcal{P}$  is about 35% that of the standard code, and it corresponds to the extra memory usage of GMRES 10. The asymptotic convergence rate of the restarted GMRES solver depends on both the number of GMRES iterations per restarted cycle and the number of multigrid cycles per GMRES iteration. Increasing the former parameter always improves the convergence rate, whereas optimal case-dependent values seem to exist for the latter one.<sup>17</sup> The extra memory allocation depends only on the chosen number of Krylov vectors and not on the number of outliers. The extra memory allocations associated with GMRES 10 and GMRES 30 are about 35 and 100% of that used by the standard fixed-point iteration respectively.

For test cases without unstable modes, the same convergence level can be obtained in considerably fewer iterations using the RPM or the GMRES iteration rather than the standard code and in this circumstance RPM requires significantly less memory than GMRES. In the presence of outliers, the more convenient choice of the solver depends on the number of unstable modes and the number

of Krylov vectors required for avoiding numerical stagnation of the residual. Both parameters are case-dependent variables, but the experience gained so far shows that the number of outliers for typical three-dimensional turbomachinery problems lies usually below 10, and the code can be prevented from stagnating only by using 30 or more Krylov vectors. These numbers are in the range in which the use of RPM results in significant computational savings over that of GMRES.

## Acknowledgments

This research has been carried out in the framework of the GEODISE project supported by the Engineering and Physical Sciences Research Council under Grant GR/R67705/01. The permission of Rolls-Royce to publish results from the HYDRA codes is gratefully acknowledged. We also acknowledge the contributions of M. C. Duta, P. Moinier, J. D. Müller, L. Lapworth, and M. West to the development of the HYDRA codes and the very useful discussions with A. Wathen and M. Embree on the properties of RPM and GMRES.

## References

- Campobasso, M. S., and Giles, M. B., "Analysis of the Effect of Mistuning on Turbomachinery Aeroelasticity," *Proceedings of the IX International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity in Turbomachines*, edited by P. Ferrand and S. Aubert, Presses Univ. de Grenoble, Grenoble, France, 2001, pp. 885–896.
- Chew, J. W., Hamby, R. H., Marshall, J. G., and Vahdati, M., "Part Speed Flutter of Transonic Fans," *Proceedings of the Applied Vehicle Technology Panel Symposium: Design Principles and Methods for Aircraft Gas Turbine Engines*, NATO Research and Technology Organization, RTO-MP-8, Neuilly-Sur-Seine, France, 1999.
- Sayma, A. L., Vahdati, M., Sbardella, L., and Imregun, M., "Modelling of Three-Dimensional Viscous Compressible Turbomachinery Flows Using Unstructured Hybrid Grids," *AIAA Journal*, Vol. 38, No. 6, 2000, pp. 945–954.
- Panovsky, J., and Carson, S. M., "Prediction of Turbine Blade Vibratory Response Due to Upstream Vane Distress," *Journal of Turbomachinery*, Vol. 120, No. 3, 1998, pp. 515–521.
- Verdon, J. M., "Review of Unsteady Aerodynamic Methods for Turbomachinery Aeroelastic and Aeroacoustic Applications," *AIAA Journal*, Vol. 31, No. 2, 1993, pp. 235–250.
- Verdon, J. M., and Caspar, J. R., "A Linearized Unsteady Aerodynamic Analysis for Transonic Cascades," *Journal of Fluid Mechanics*, Vol. 149, Dec. 1984, pp. 403–429.
- Hall, K. C., "Deforming Grid Variational Principle for Unsteady Small Disturbance Flows in Cascades," *AIAA Journal*, Vol. 31, No. 5, 1993, pp. 891–900.

- <sup>8</sup>Breard, C., Imregun, M., Green, J. S., and Elliott, R., "A Study of Low Engine Order Excitation in Turbomachines," *Proceedings of the IX International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity in Turbomachines*, edited by P. Ferrand and S. Aubert, Presses Univ. de Grenoble, Grenoble, France, 2001, pp. 397–408.
- <sup>9</sup>Ning, W., Li, Y. S., and Wells, R. G., "Predicting Blade-Row Interactions Using a Multistage Time-Linearized Navier–Stokes Solver," *Journal of Turbomachinery*, Vol. 105, No. 1, 2003, pp. 25–32.
- <sup>10</sup>Clark, W. S., and Hall, K. C., "A Time-Linearized Navier–Stokes Analysis of Stall Flutter," *Journal of Turbomachinery*, Vol. 122, No. 3, 2000, pp. 467–476.
- <sup>11</sup>Hall, K. C., Clark, W. S., and Lorence, C. B., "A Linearized Euler Analysis of Unsteady Transonic Flows in Turbomachinery," *Journal of Turbomachinery*, Vol. 116, No. 3, 1994, pp. 477–488.
- <sup>12</sup>Sbardella, L., and Imregun, M., "Linearized Unsteady Viscous Turbomachinery Flows Using Hybrid Grids," *Journal of Turbomachinery*, Vol. 123, No. 3, 2001, pp. 568–582.
- <sup>13</sup>Moinier, P., Müller, J. D., and Giles, M. B., "Edge-Based Multigrid and Preconditioning for Hybrid Grids," *AIAA Journal*, Vol. 40, No. 10, 2002, pp. 1954–1960.
- <sup>14</sup>Duta, M. C., Giles, M. B., and Campobasso, M. S., "The Harmonic Adjoint Approach to Unsteady Turbomachinery Design," *International Journal for Numerical Methods in Fluids*, Vol. 40, Oct. 2002, pp. 323–332.
- <sup>15</sup>Campobasso, M. S., Duta, M. C., and Giles, M. B., "Adjoint Calculation of Sensitivities of Turbomachinery Objective Functions," *Journal of Propulsion and Power*, Vol. 19, No. 4, 2003, pp. 693–703.
- <sup>16</sup>Giles, M. B., Duta, M. C., Müller, J. D., and Pierce, N. A., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.
- <sup>17</sup>Campobasso, M. S., and Giles, M. B., "Effects of Flow Instabilities on the Linear Analysis of Turbomachinery Aeroelasticity," *Journal of Propulsion and Power*, Vol. 19, No. 2, 2003, pp. 250–259.
- <sup>18</sup>Saad, Y., and Schultz, M. H., "GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.
- <sup>19</sup>Shroff, G. M., and Keller, H. B., "Stabilization of Unstable Procedures: the Recursive Projection Method," *SIAM Journal of Applied Mathematics*, Vol. 30, No. 4, 1993, pp. 1099–1120.
- <sup>20</sup>Moinier, P., "Algorithm Developments for an Unstructured Viscous Flow Solver," Ph.D. Dissertation, Computing Lab., Oxford Univ., Oxford, Dec. 1999.
- <sup>21</sup>Duta, M. C., "The Use of the Adjoint Method for the Minimization of Forced Response," Ph.D. Dissertation, Computing Lab., Oxford Univ., Oxford, March 2002.
- <sup>22</sup>Ni, R. H., and Sisto, F., "Numerical Computation of Nonstationary Aerodynamics of Flat Plate Cascades in Compressible Flow," *Journal of Engineering for Power*, Vol. 98, No. 2, 1976, pp. 165–170.
- <sup>23</sup>Giles, M. B., "Nonreflecting Boundary Conditions for Euler Equation Calculations," *AIAA Journal*, Vol. 28, No. 12, 1990, pp. 2050–2058.
- <sup>24</sup>Golub, G. H., and Van Loan, C. F., "Orthogonalization and Least Squares," *Matrix Computations*, 3rd ed., Johns Hopkins Univ. Press, Baltimore, MD, 1996, pp. 306–374.
- <sup>25</sup>Duta, M. C., Campobasso, M. S., Giles, M. B., and Lapworth, L. B., "Adjoint Harmonic Sensitivities for Forced Response Minimization," American Society of Mechanical Engineers, Paper GT2003-38904, June 2003.
- <sup>26</sup>Fransson, T. H., Joecker, M., Boelcs, A., and Ott, P., "Viscous and Inviscid Linear/Nonlinear Calculations Versus Quasi Three-Dimensional Experimental Cascade Data for a New Aeroelastic Turbine Standard Configuration," *Journal of Turbomachinery*, Vol. 121, No. 4, 1999, pp. 717–725.
- <sup>27</sup>Pueyo, A., and Zingg, D. W., "Efficient Newton–Krylov Solver for Aerodynamic Computations," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1991–1997.
- <sup>28</sup>Ning, W., and He, L., "Some Modelling Issues on Trailing-Edge Vortex Shedding," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 787–793.
- <sup>29</sup>Glendinning, P., "Stability, Instability and Chaos," *Bifurcation Theory: Stationary Points*, Cambridge Univ. Press, Cambridge, England, U.K., 1994, pp. 224–246.

E. Livne  
Associate Editor