AIAA JOURNAL
Vol. 46, No. 12, December 2008

Block-Jacobi Implicit Algorithms for the Time Spectral Method

Frédéric Sicot,* Guillaume Puigt,† and Marc Montagnac‡
European Center for Research and Advanced Training in Scientific Computation, F-31057 Toulouse Cedex 01, France

DOI: 10.2514/1.36792

Many industrial applications involve flows periodic in time. Such flows are not simulated with enough efficiency when using classical unsteady techniques, as a transient regime must be bypassed. New techniques, dedicated to time-periodic flows and based on Fourier analysis, have been developed recently. Among these, the time spectral method casts a time-periodic flow computation in several coupled steady computations, corresponding to a uniform sampling of the period. Up until now, the steady states were reached using an explicit pseudotime algorithm. Thus, very small time steps were needed and the convergence rate slowed down when increasing the number of harmonics. In this paper, a block-Jacobi approach is presented to solve the stationary problems with an implicit algorithm. Numerical simulations show, on one hand, the good quality of the results and, on the other hand, the interest of the proposed method to reduce the sensitivity of computations to a large number of harmonics.

I. Introduction

EVEN if three-dimensional steady turbulent flow simulations begin to be handled routinely in the aircraft industry, three-dimensional unsteady turbulent flow simulations still require large amounts of computing time, and a substantial acceleration of the calculations is needed to reduce design cycles. Depending on the spatial and time scales to be resolved, numerous nonlinear time-marching methods are available. Direct numerical simulations and even large eddy simulations are still too expensive with respect to the best computing resources available today to satisfy industrial requirements. So far, unsteady Reynolds-averaged Navier–Stokes (U-RANS) techniques have proven to be the most efficient to meet industrial needs. Efficiency is not an absolute notion because it results from a tradeoff between the quality of the physics and the time needed to complete the simulation. In industry, U-RANS techniques are generally predictive enough and require relatively short time simulations.

To build an efficient method for unsteady flows, it is interesting to take into consideration all the flow characteristics. As an example, a large range of applications leads to time-periodic flows: turbomachinery, pitching wings, helicopter blades, wind turbines, etc. Several dedicated methods have been developed during the last years. They consider flow variables either in the time domain or in the frequency domain. The frequency-domain techniques are extensively reviewed in [1,2]. Linearized methods [3] form an important group among these methods. They superimpose perturbations over a steady flow and do not really rely on a time-marching procedure. Consequently, they are very inexpensive to compute. However, when the flow presents strong shock discontinuities, for instance, the linearity assumption is no longer true. Ning and He [4] extend these techniques to take account of the nonlinearities, yielding the nonlinear harmonic method. This one is limited to only one harmonic of the flow and requires a specific treatment for the time stepping.

Received 23 January 2008; revision received 9 July 2008; accepted for publication 3 September 2008.

*Doctoral Candidate, Computational Fluid Dynamics/Advanced Aerodynamics and Multiphysics, 42 Avenue Coriolis.

†Senior Researcher, Computational Fluid Dynamics/Aérodynamique et Applications Multiphysiques, 42 Avenue Coriolis.

‡Research Engineer, Computational Fluid Dynamics/Aérodynamique et Applications Multiphysiques, 42 Avenue Coriolis.

In recent years, a more efficient time-domain method dedicated to time-periodic flows has been developed. Hall et al. introduced a harmonic balance (HB) method [5] for blade cascades computations. Then Gopinath and Jameson presented the time spectral method (TSM) [6] for external aerodynamic applications. Both methods are essentially the same and allow one to capture the fundamental frequency of the flow and a given number of its harmonics. They cast the unsteady governing equations in a set of coupled steady equations corresponding to a uniform sampling of the flow within the time period. These steady equations can then be solved using standard steady RANS methods with convergence acceleration techniques such as local time stepping and multigrid. The convergence of a steady computation is better mastered than the transient needed by an unsteady computation to reach the periodic state. This method proved to be efficient in periodic problem computations such as vortex shedding [7,8], flutter [9], and turbomachinery applications [1,10]. Later, the HB method was extended for multistage turbomachinery [11] where several frequencies appear, not necessarily integer multiples of each other. For sake of clarity, the notation TSM is retained in this paper to refer to computations with a single fundamental frequency.

In [5–11], explicit algorithms such as Runge–Kutta methods are used to advance the calculations in pseudotime. This makes the pseudotime steps relatively small and therefore requires a large number of iterations to reach the steady state of all the instants. Furthermore, it has been observed that the convergence rate decays as the number of harmonics is increased [5]. To circumvent this, van der Weide et al. [10] use a spectral interpolation of computations with a low number of harmonics to produce good initial conditions for higher harmonics computations. This requires one to make several computations and to interpolate between each. An implicit time integration scheme, such as backward-Euler, could enable much larger time steps and make the TSM even more efficient. For this reason, the goal of the present paper is to derive and implement an implicit version of the TSM.

The following section recalls the formulation of the time spectral method and its stability criteria. Then, the new implicit treatment is described and two solving processes are derived. A numerical study is then carried out, followed by an application of a pitching wing in forced harmonic oscillations.

II. Time Spectral Method

A. Governing Equations

The Navier–Stokes equations in Cartesian coordinates are written in semidiscrete form as

$$V\frac{\partial W}{\partial t}+R(W)=0 \tag{1}$$

where V is the volume of a cell, and W is the vector of conservative variables

$$W=(\rho,\rho u_1,\rho u_2,\rho u_3,\rho E)^T$$

complemented with an arbitrary number of turbulent variables as within the RANS framework. $R(W)$ is the residual vector resulting from spatial discretization of the convective f_{ci} and viscous f_{vi} fluxes

$$R(W)=\frac{\partial}{\partial x_i}f_i(W)$$

with $f_i=f_{\mathrm{ci}}-f_{\mathrm{vi}}$ and

$$f_{\mathrm{ci}}=\begin{pmatrix}\rho u_i\\ \rho u_i u_1+p\delta_{i1}\\ \rho u_i u_2+p\delta_{i2}\\ \rho u_i u_3+p\delta_{i3}\\ \rho u_i E+p u_i\end{pmatrix},\qquad f_{\mathrm{vi}}=\begin{pmatrix}0\\ \tau_{i1}\\ \tau_{i2}\\ \tau_{i3}\\ u\cdot\tau_i-q_i\end{pmatrix} \tag{2}$$

Here, δ denotes the Kronecker symbol. The components of the stress tensor are

$$\begin{aligned}
\tau_{11}&=\frac{2}{3}\mu\left(2\frac{\partial u_1}{\partial x_1}-\frac{\partial u_2}{\partial x_2}-\frac{\partial u_3}{\partial x_3}\right)\\
\tau_{12}=\tau_{21}&=\mu\left(\frac{\partial u_2}{\partial x_1}+\frac{\partial u_1}{\partial x_2}\right)\\
\tau_{22}&=\frac{2}{3}\mu\left(-\frac{\partial u_2}{\partial x_2}+2\frac{\partial u_2}{\partial x_2}-\frac{\partial u_3}{\partial x_3}\right)\\
\tau_{13}=\tau_{31}&=\mu\left(\frac{\partial u_3}{\partial x_1}+\frac{\partial u_1}{\partial x_3}\right)\\
\tau_{33}&=\frac{2}{3}\mu\left(-\frac{\partial u_3}{\partial x_3}-\frac{\partial u_2}{\partial x_2}+2\frac{\partial u_3}{\partial x_3}\right)\\
\tau_{23}=\tau_{32}&=\mu\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right)
\end{aligned}$$

The heat flux vector q components are $q_i=-\kappa\partial T/\partial x_i$, where T is the temperature and

$$\kappa=C_p\left(\frac{\mu_{\mathrm{lam}}}{Pr_{\mathrm{lam}}}+\frac{\mu_{\mathrm{turb}}}{Pr_{\mathrm{turb}}}\right)$$

The total viscosity μ is the sum of the laminar μ_{lam} and turbulent μ_{turb} viscosities. Pr_{lam} and Pr_{turb} are the associated Prandtl number. For an ideal gas, the closure is provided by the equation of state

$$p=(\gamma-1)\rho\left(E-\frac{u_i u_j}{2}\right)$$

B. Fourier-Based Time Discretization

If W is periodic with period $T=2\pi/\omega$, then so is $R(W)$, and the Fourier series of Eq. (1) is

$$\sum_{k=-\infty}^{\infty}(ik\omega V\hat{W}_k+\hat{R}_k)e^{ik\omega t}=0 \tag{3}$$

where $\hat{W}_k$ and $\hat{R}_k$ are the Fourier coefficients of W and R corresponding to mode k. Because the set of complex exponential functions forms an orthogonal basis, the only way for Eq. (3) to be true is that the weight of every mode k is zero. An infinite number of steady equations in the frequency domain is obtained as expressed by

$$ik\omega V\hat{W}_k+\hat{R}_k=0,\quad \forall\ k\in\mathbb{Z} \tag{4}$$

McMullen et al. [12] solve a subset of these equations up to mode N, $-N\le k\le N$, yielding the nonlinear frequency-domain method.

The time spectral method [6] uses a discrete inverse Fourier transform (DIFT) to cast back in the time domain this subset of $2N+1$ equations from Eq. (4). The DIFT induces linear relations between Fourier's coefficients $\hat{W}_k$ and a uniform sampling of W within the period

$$W_n=\sum_{k=-N}^{N}\hat{W}_k\exp(i\omega n\Delta t),\qquad 0\le n<2N+1$$

with $W_n\equiv W(n\Delta t)$ and $\Delta t=T/(2N+1)$. This leads to a time discretization with a new time operator D_t as follows:

$$R(W_n)+VD_t(W_n)=0,\qquad 0\le n<2N+1 \tag{5}$$

These steady equations correspond to $2N+1$ instants equally spaced within the period. The new time operator connects all time levels and can be expressed analytically by

$$D_t(W_n)=\sum_{m=-N}^{N}d_m W_{n+m}$$

with

$$d_m=\begin{cases}\frac{\pi}{T}(-1)^{m+1}\csc(\frac{\pi m}{2N+1}), & m\neq 0\\ 0, & m=0\end{cases}$$

A similar derivation can be made for an even number of instants, but it is proven in [10] that it can lead to an odd–even decoupling and, as a consequence, the method can become unstable. Time-dependent boundary conditions could also benefit from such a derivation, but this is not an issue for external aerodynamic applications and has not been done yet.

A pseudotime derivative τ_n is added to Eq. (5) to time march the equations to the steady-state solutions of all instants:

$$V\frac{\partial W_n}{\partial\tau_n}+R(W_n)+VD_t(W_n)=0,\qquad 0\le n<2N+1 \tag{6}$$

The term $VD_t(W_n)$ appears as a source term that represents a high-order formulation of the initial time derivative in Eq. (1). For stability reasons, the computation of the local time step is modified [10] to take into account this additional source term:

$$\Delta\tau=\mathrm{CFL}\frac{V}{\|\lambda\|+\omega NV} \tag{7}$$

where CFL denotes the Courant–Friedrichs–Lewy number. An extra term ωNV is added to the spectral radius $\|\lambda\|$ to restrict the time step. Equation (7) implies that a high frequency and/or a high number N of harmonics can considerably constrain the time step. Actually, it has been observed [5] that the convergence of the method slows down for increasing N. Explicit schemes are used in [5–11], such as Runge–Kutta, to carry out the pseudotime integration. Their limited stability criteria on CFL numbers is very sensitive to such a restriction. Conversely, implicit schemes are more stable and allow larger CFL numbers, reducing this sensitivity. Such schemes would have the same behavior when the frequency of the unsteadiness increases. The following section describes the backward-Euler algorithm for the TSM.

III. Implicit Time Integration

Let us recall the backward-Euler algorithm for the Navier–Stokes equations and the standard solving lower upper-symmetric successive overrelaxation [13] (LU-SSOR) method.

A. Algorithm for Steady Navier–Stokes Equations

The time derivative in Eq. (1) is discretized by a first-order scheme

$$V\frac{\Delta W}{\Delta t} = -R(W) \tag{8}$$

where $\Delta W = W^{q+1} - W^q$ is the increment of the conservative variables between the iterations q and $q+1$. By considering $R(W)$ at iteration $q+1$, the implicit backward-Euler scheme is derived. As $R(W^{q+1})$ is unknown, it is linearized. Let J be the Jacobian matrix of the residual vector $J = \partial R(W)/\partial W$. The linearization of $R(W^{q+1})$ is then

$$R(W^{q+1}) = R(W^q) + J\Delta W + \mathcal{O}(\Delta W^2) \tag{9}$$

Equations (8) and (9) lead to the following linear system:

$$\left(\frac{V}{\Delta t}I + J\right)\Delta W = -R(W^q)$$

The LU-SSOR method is used to approximate the solution of this system. Formally, the matrix A of the linear system is split into three matrices:

$$A\Delta W = (\mathcal{L} + \mathcal{D} + \mathcal{U})\Delta W = -R(W^q) \tag{10}$$

with $\mathcal{L}$ a lower triangular matrix, $\mathcal{D}$ a diagonal matrix, and $\mathcal{U}$ an upper triangular matrix. One LU-SSOR step is composed of the forward and backward sweeps of the iterative symmetric successive overrelaxation (SSOR) method [Eq. (11)], performed one after the other for $s \geq 0$,

$$\begin{cases} (\mathcal{L} + \mathcal{D})\Delta W^{s+1/2} = -R(W^q) - \mathcal{U}\Delta W^s, \\ (\mathcal{U} + \mathcal{D})\Delta W^{s+1} = -R(W^q) - \mathcal{L}\Delta W^{s+1/2} \end{cases} \tag{11}$$

with $\Delta W^0 = 0$. These two sweeps are repeated several times and $W^{q+1} = W^q + \Delta W^{s_{\max}}$, $s_{\max}$ corresponding to the maximum number of LU-SSOR steps.

Convective fluxes are written with a first-order Steger and Warming [14] flux vector splitting for the residual linearization to end up with a diagonally dominant implicit matrix, which ensures that the method is convergent. Viscous terms are also linearized and preserve this diagonal dominance. Artificial dissipation is added for stability issues. The boundary conditions could be linearized in a same manner as the residual operator. As the scalar LU-SSOR is used in this paper, this point is not required to get convergence. The relaxation parameter is set to unity as it gives the best performances. This is equivalent to removing overrelaxation and using lower upper-symmetric Gauss–Seidel. Nevertheless, in the following sections, we keep the LU-SSOR designation but, for the sake of simplicity, the derived equations do not mention the relaxation parameter. This method has proven its efficiency in an industrial context for several years.

B. Extension for the Time Spectral Method

To introduce an implicit algorithm in the TSM, the first approach is to linearize only the residual $R(W_n)$ of Eq. (6), but not the source term $VD_t(W_n)$. This leads to the augmented system

$$\begin{pmatrix} \frac{V}{\Delta\tau_0}I + J_0 & 0 & \dots & 0 \\ 0 & \frac{V}{\Delta\tau_1}I + J_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{V}{\Delta\tau_{2N}}I + J_{2N} \end{pmatrix} \begin{pmatrix} \Delta W_0 \\ \Delta W_1 \\ \vdots \\ \Delta W_{2N} \end{pmatrix} = -\begin{pmatrix} R_{\mathrm{TSM}}(W_0^q) \\ R_{\mathrm{TSM}}(W_1^q) \\ \vdots \\ R_{\mathrm{TSM}}(W_{2N}^q) \end{pmatrix} \tag{12}$$

with $R_{\mathrm{TSM}}(W_n^q) = R(W_n^q) + VD_t(W_n^q)$ on the right-hand side of the TSM equations, and J_n the Jacobian of the standard residual operator at instant n, $J_n = \partial R(W_n)/\partial W_n$. The augmented matrix is block diagonal and an LU-SSOR algorithm can be applied independently on each instant n. In other words, $2N+1$ steady flows are computed, and they are only coupled through the explicit residuals. This is clearly an advantage because this approach has no impact on message passing and does not need any new development on the implicit side. However, it will be shown in Sec. IV that convergence is not achieved easily with this technique, and the present paper proposes another alternative.

C. Full Implicitation Method for the Time Spectral Method

To improve the performances, the source term of the TSM needs to be taken into account. The TSM equations with W considered at iteration $q+1$ read

$$V\frac{\Delta W_n}{\Delta\tau_n} = -\left[R\left(W_n^{q+1}\right) + VD_t\left(W_n^{q+1}\right)\right], \qquad 0 \leq n < 2N+1 \tag{13}$$

As the operator D_t is linear, applying it on W_n at iteration $q+1$ gives

$$D_t\left(W_n^{q+1}\right) = D_t\left(W_n^q\right) + D_t(\Delta W_n) \tag{14}$$

In the same manner as the TSM new time operator D_t couples together the conservative variables at all instants, Eq. (14) leads to a coupling of the increments ΔW at all instants. Equation (13) turns into

$$\left(\frac{V}{\Delta\tau_n}I + J_n\right)\Delta W_n + VD_t(\Delta W_n) = -R_{\mathrm{TSM}}\left(W_n^q\right), \qquad 0 \leq n < 2N+1$$

As $d_0 = 0$, the diagonal terms are identical to the diagonal terms of Eq. (12). The matrix of the system becomes

$$A^* = \begin{pmatrix} \frac{V}{\Delta\tau_0}I + J_0 & Vd_1I & \dots & Vd_NI & Vd_{-N}I & \dots & Vd_{-1}I \\ Vd_{-1}I & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Vd_1I & \ddots & \ddots & \vdots \\ Vd_{-N}I & \dots & Vd_{-1}I & \frac{V}{\Delta\tau_N}I + J_N & Vd_1I & \dots & Vd_NI \\ \vdots & \ddots & \ddots & Vd_{-1}I & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & Vd_1I \\ Vd_1I & \dots & Vd_NI & Vd_{-N}I & \dots & Vd_{-1}I & \frac{V}{\Delta\tau_{2N}}I + J_{2N} \end{pmatrix}$$

The new matrix A^* is not block sparse anymore and couples all the increments ΔW_n of all the instants n. This probably explains why the adapted LU-SSOR scheme of Sec. III.B fails to converge for a high number of harmonics, as shown in Sec. IV: the linearization error grows as the number of harmonics increases, and so the convergence rate decays.

A^* could be decomposed as a sum of three matrices $A^* = \mathcal{L}^* + \mathcal{D}^* + \mathcal{U}^*$, with $\mathcal{L}^*$ a lower block triangular matrix, $\mathcal{D}^*$ a block diagonal matrix, and $\mathcal{U}^*$ an upper block triangular matrix. Then, a classical SSOR algorithm could be applied on the whole system, but it would necessitate one to go all over the blocks and thus would break down code efficiency in terms of CPU requirement. To remove this drawback, two solving algorithms based on the block-Jacobi method are now presented.

D. Block-Jacobi Strategies for Full Implicit Time Spectral Method

Applied to the TSM, the iterative block-Jacobi method [15] allows one to move the implicit coupling term $VD_t(\Delta W_n)$ to the right-hand side and yields $2N + 1$ independent linear systems. A Jacobi step l reads

$$\left(\frac{V}{\Delta\tau_n}I + J_n\right)\Delta W_n^{l+1} = -R_{\mathrm{TSM}}\left(W_n^q\right) - VD_t\left(\Delta W_n^l\right), \qquad 0 \le n < 2N+1 \tag{15}$$

with $l \ge 0$, $\Delta W_n^0 = 0$, and at the end of the $l_{\max}$ block-Jacobi iterations, the increments ΔW_n allow one to compute W at the next iteration: $W_n^{q+1} = W_n^q + \Delta W_n^{l_{\max}}$. For every block-Jacobi step, a linear system has to be solved. This system could be solved with any direct or iterative method. The classical SSOR technique is actually used, as it allows minimum efforts to be adapted from the LU-SSOR method.

1. Block-Jacobi Symmetric Successive Overrelaxation Strategy

Each equation of the block-Jacobi system, Eq. (15), could be solved with an iterative SSOR technique, classically decomposed in a forward sweep

$$(\mathcal{L}_n + \mathcal{D}_n)X^{s+1/2} = -\left[R_{\mathrm{TSM}}\left(W_n^q\right) + VD_t\left(\Delta W_n^l\right)\right] - \mathcal{U}_n X^s \tag{16}$$

followed by a backward sweep

$$(\mathcal{D}_n + \mathcal{U}_n)X^{s+1} = -\left[R_{\mathrm{TSM}}\left(W_n^q\right) + VD_t\left(\Delta W_n^l\right)\right] - \mathcal{L}_n X^{s+1/2} \tag{17}$$

for $s \ge 0$ with $X^0 = \Delta W_n^l$. At the end of the SSOR iterations, $X^{s_{\max}}$ is updated into the block-Jacobi steps: $\Delta W_n^{l+1} = X^{s_{\max}}$, $s_{\max}$ being the number of SSOR forward and backward sweeps inside a block-Jacobi step. It should be noticed that $\mathcal{L}_n$, $\mathcal{D}_n$, and $\mathcal{U}_n$ refer to Eq. (10), where the splitting of the implicit matrix was obtained for one instant n. The block-Jacobi method imposes the implicit coupling term $D_t(\Delta W_n^l)$ to be updated at each step l. In other words, the implicit coupling term is computed every $2s_{\max}$ sweeps and frozen over the following $2s_{\max} - 1$ sweeps. As $\Delta W_n^0 = 0$, it remains null during all the sweeps in the first block-Jacobi step and, consequently, at least two steps are needed to ensure the coupling of the increments of all instants $l_{\max} \ge 2$. If $l_{\max} = 1$, no implicit coupling occurs and Eq. (12) is recovered. The solving algorithm uses two nested loops as described by Algorithm 1.

To reinforce the influence of the implicit coupling, the next method is proposed.

2. Block-Jacobi Successive Overrelaxation Strategy

The system Eq. (15) could also be solved in a special way with alternate successive overrelaxation (SOR) techniques. Only a loop is needed and the imposed constraint is to have an even number $l_{\max}$ of block-Jacobi steps to balance forward and backward sweeps. Indeed, when l is even, the system is solved with only one forward SOR sweep [Eq. (16)], and when l is odd, the system is solved with only one backward SOR sweep [Eq. (17)]. Algorithm 2 describes this strategy.

Algorithm 1 Block-Jacobi-SSOR algorithm for the time spectral method

Require: W_n^q, $l_{\max} \ge 2$, $s_{\max} \ge 1$
  $\Delta W_n^0 = 0$
  for $l = 0$ to $l_{\max} - 1$ do
    compute $D_t(\Delta W_n^l)$
    $X^0 = \Delta W_n^l$
    for $s = 0$ to $s_{\max} - 1$ do
      solve Eq. (16) {Forward sweep}
      solve Eq. (17) {Backward sweep}
    end for
    $\Delta W_n^{l+1} = X^{s_{\max}}$
  end for
Ensure: $W_n^{q+1} = W_n^q + \Delta W_n^{l_{\max}}$

Algorithm 2 Block-Jacobi-SOR algorithm for the time spectral method

Require: W_n^q, $l_{\max}$ even
  $\Delta W_n^0 = 0$
  for $l = 0$ to $l_{\max} - 1$ do
    compute $D_t(\Delta W_n^l)$
    if l is even then {Forward sweep}
      $X^s = \Delta W_n^l$, solve Eq. (16), $\Delta W_n^{l+1} = X^{s+1/2}$
    else {l is odd, Backward sweep}
      $X^{s+1/2} = \Delta W_n^l$, solve Eq. (17), $\Delta W_n^{l+1} = X^{s+1}$
    end if
  end for
Ensure: $W_n^{q+1} = W_n^q + \Delta W_n^{l_{\max}}$

The implicit coupling term $VD_t(\Delta W_n)$ is computed before every sweep (but the first one as $\Delta W_n^0 = 0$), and thus this strategy ensures the strongest coupling. If $s_{\max} = 2$ in the block-Jacobi symmetric successive overrelaxation (BJ-SSOR) method, for instance, the implicit coupling term is computed before the fifth (forward) sweep and frozen over the three following sweeps. Table 1 enables the comparison between the two methods in terms of SOR sweeps.

The LU-SSOR and the new block-Jacobi methods are now compared. The influence of the number of block-Jacobi iterations on convergence rate is also studied.

IV. Validation of the Implicit Time Spectral Method

The TSM technique has been implemented in the parallel structured multiblock solver elsA [16]. The code capability is wide, as it can simulate steady and unsteady, internal and external flows, in a relative or fixed motion. The solver uses a conservative cell-centered finite volume approach for the spatial discretization. Several spatial and time integration schemes are available. In this paper, the second-order Jameson–Schmidt–Turkel centered scheme [17] is used for convective terms and a central second-order scheme is used for diffusive terms. In combination with local time stepping, a

Table 1 Example of implicit coupling term update up to eight sweeps (values of the loop indexes l and s before sweep, and if the implicit coupling term is updated)

Number of sweeps	BJ-SSOR $s_{\max} = 2$			BJ-SOR	
	l	s	Update	l	Update
1	0	0	No	0	No
2	0	0	No	1	Yes
3	0	1	No	2	Yes
4	0	1	No	3	Yes
5	1	0	Yes	4	Yes
6	1	0	No	5	Yes
7	1	1	No	6	Yes
8	1	1	No	7	Yes

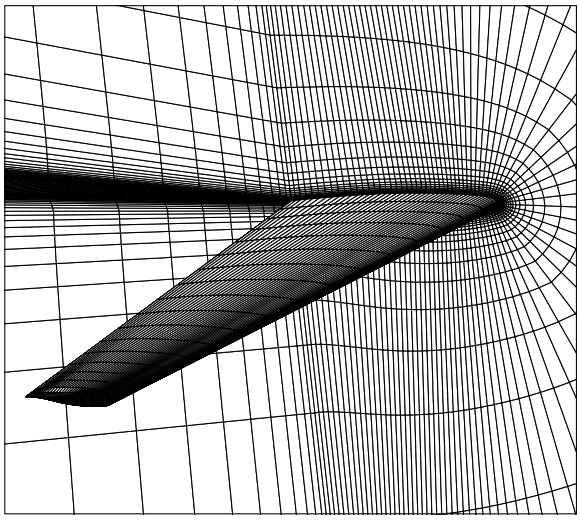


Fig. 1 Navier–Stokes mesh of the LANN wing (for a better readability, the mesh is coarsened twice in every direction).

V-cycle multigrid technique with two levels of coarse grids is used to accelerate the convergence of the steady computations. The Spalart–Allmaras [18] turbulence model is used in all the simulations.

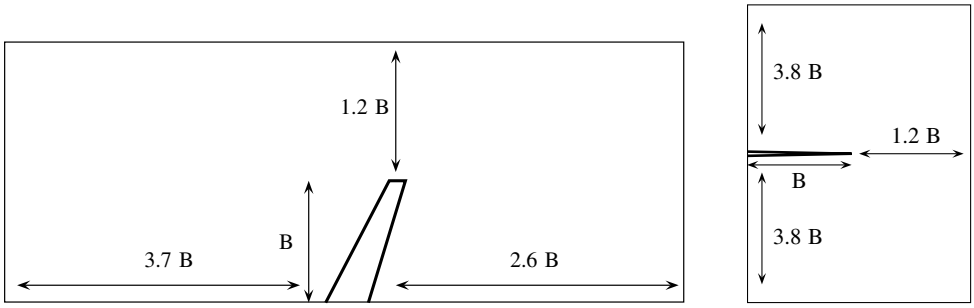
The new implicit algorithm has been validated with the flow simulation around a transonic wing in a forced pitching movement. This wing has been designed by Lockheed Georgia, Air Force Flight Dynamic Laboratory, NASA Langley, and National Aerospace Laboratory (The Netherlands), yielding the name of LANN wing. Experimental data [19] are provided for the frequency $f = 24$ Hz. The angle of attack α oscillates as $\alpha(t) = \alpha_0 + \alpha_m \sin(2\pi f t)$ with $\alpha_0 = 0.6$ deg and $\alpha_m = 0.25$ deg. The flow conditions are $M_\infty = 0.822$ and $Re = 5.43 \times 10^6$. Experimental data are available for the time-averaged and the first harmonic values of the wall pressure coefficient C_p at different wing cross sections.

The numerical simulations are conducted on a mesh composed of 1,122,816 cells as shown in Fig. 1. The grid extent is shown Fig. 2 where B denotes the wingspan.

A. Numerical Study

First, a numerical study of the different parameters is conducted. The convergence curves for the first approach described in Sec. III.B are given in Fig. 3a. The solution residual is defined as the root mean square of the residual operator $R(W)$ on all the mesh cells, averaged by the number of instants. The curves indicate the residual on the density residual ρ and are normalized by the residual at first iteration to enable comparison. It is observed that the CFL needs to be decreased to converge high-harmonic computations. For $N = 4$, the CFL must be decreased to 20 (dotted line) because, with 30, the computation does not converge (dashed line). The five-harmonic computation needs a few thousand iterations at CFL = 5 to lose 5 orders of magnitude: the convergence rate is very slow. The first block-Jacobi strategy used is the BJ-SOR, Sec. III.D.2, as it should ensure the best coupling. The results with $l_{\max} = 4$ are presented in Fig. 3b. The benefits of the full implicitation are clear as all the computations are now performed at CFL = 100. Furthermore, almost no differences are found between the normalized convergence curves. Not all test cases show such a good matching, but it is observed that the convergence rate is nearly the same for any number of harmonics.

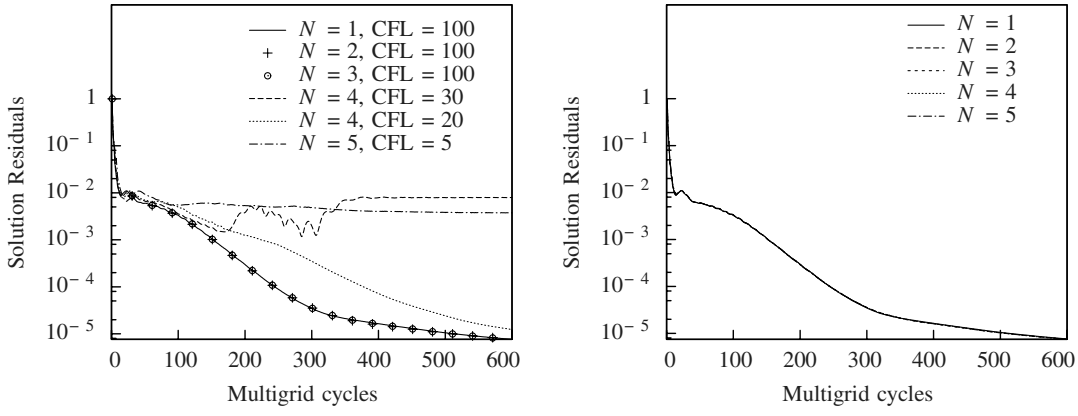
Up until now, results have been obtained using four SOR sweeps, leading finally to the first forward sweep without implicit coupling ($\Delta W = 0$ initially) and the three other sweeps with implicit coupling (cf. Table 1). The influence of the derived strategies is shown in Fig. 4 for the most difficult case $N = 5$. It is observed that the BJ-SOR



a) Top view: upstream, downstream, and spanwise grid boundaries

b) Front view: upper, lower, and spanwise grid boundaries

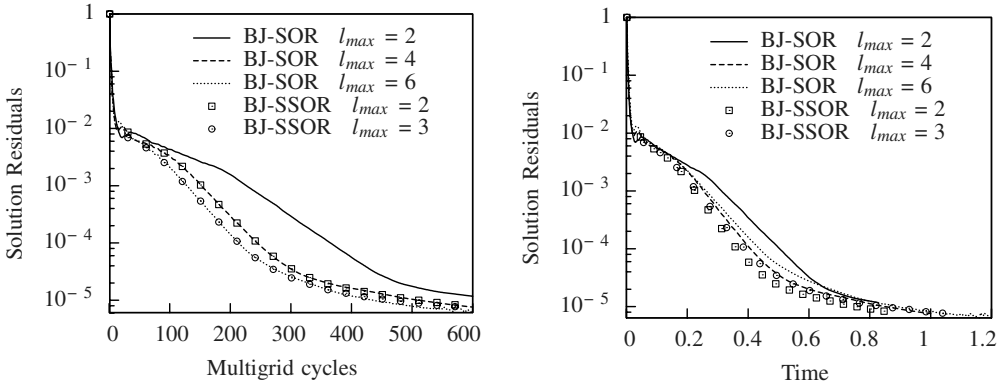
Fig. 2 Grid extent (not to scale).



a) Standard LU-SSOR

b) BJ-SOR (all CFL = 100)

Fig. 3 Convergence of the computations.



a) Convergence curves with respect to the number of multigrid cycles

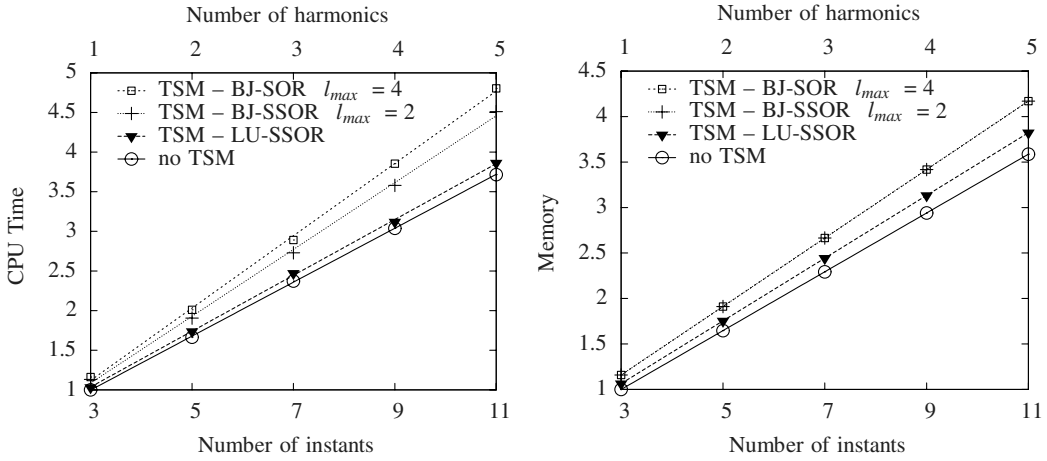
b) Convergence curves with respect to a nondimensional CPU time

Fig. 4 Effect of the number of modified LU-SSOR steps on convergence (all CFL = 100).

strategy with $l_{\max} = 2$ (with only the backward sweep ensuring the implicit coupling) is sufficient to obtain convergence at CFL = 100. Thanks to the three coupling sweeps, the BJ-SOR method with $l_{\max} = 4$ substantially speeds up the convergence with respect to the number of multigrid cycles (Fig. 4a). The nondimensional time of computation (Fig. 4b) remains mostly in favor of the BJ-SOR method. The best convergence rate in terms of multigrid cycles is obtained with the BJ-SOR strategy with $l_{\max} = 6$, but this advantage is lost when considering the time spent. The extra CPU cost induced by the two supplementary sweeps is not worth the gain in convergence rate.

Results from the BJ-SSOR algorithm described in Sec. III.D.1 are represented by marks in Fig. 4. As $\Delta W_n = 0$ for the first block-Jacobi step, at least two steps are needed to ensure the coupling of increments (see Table 1). As shown previously with the BJ-SOR method, six sweeps are already expensive. To ensure an implicit coupling as often as possible with this few numbers of sweeps, $s_{\max}$ is set to one for the BJ-SSOR algorithm. Even though the coupling occurs less often, the convergence rate is almost the same for the BJ-SOR method with $l_{\max} = 4$ as for the BJ-SSOR method with $l_{\max} = 2$, and slightly slowed down with two additional sweeps for each method (respectively, $l_{\max} = 6$ and $l_{\max} = 3$) with respect to the number of multigrid cycles. As the term $D_t(\Delta W_n)$ is only computed every two sweeps, the CPU time required by the BJ-SSOR method is notably reduced compared with the BJ-SOR approach with the same number of sweeps.

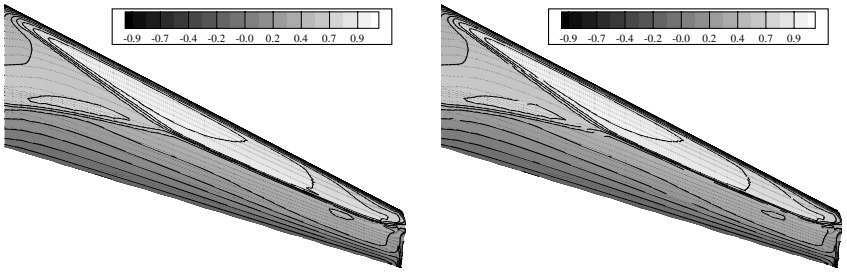
The CPU and memory costs of the implicitation are not negligible, as shown in Fig. 5. The lines show the trend and do not necessarily pass exactly through the data points. All the computations are performed on a parallel computer, which shows a variation of up to 5% in CPU time, so that this statistic is averaged over four runs of simulation. They are all performed with four sweeps, either $l_{\max} = 4$ for the BJ-SOR method or $l_{\max} = 2$ for the BJ-SSOR method. All the curves are normalized by the cost of three uncoupled steady



a) CPU cost

b) Memory cost

Fig. 5 Costs of the different implicitation strategies.



a) U-RANS

b) TSM

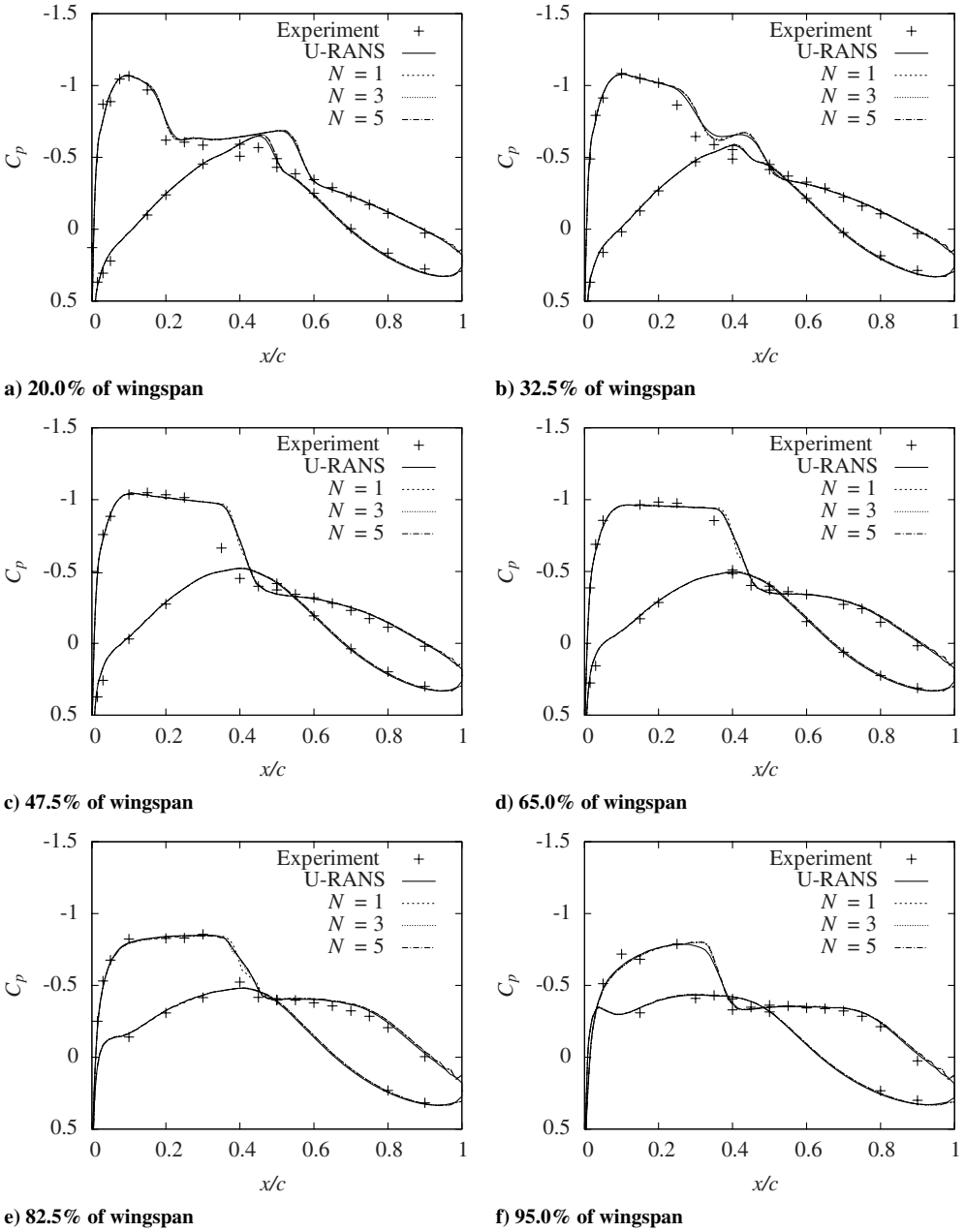
Fig. 6 LANN wing upper surface. Instantaneous pressure coefficient at $\alpha = 0.6$ deg for increasing α.

computations. The circles indicate the CPU time and memory consumption required for $2N + 1$ uncoupled steady computations. The inverted triangles denote the TSM with the first approach of Eq. (12) with standard LU-SSOR. The complexity of the computation of D_t is quadratic with respect to the number of instants, but it is only involved in a small part of the global calculations, and the whole time remains linear with respect to the number of instants. Finally, the first approach adds penalties of about 3.5% in CPU time and 6.5% in memory. When considering the BJ-SOR method (squares), D_t is applied several times over the conservative variable increment ΔW_n inside the SOR sweeps. Nevertheless, the CPU time remains linear. The extra cost is significant as the CPU time is increased by 30% and the memory by 10% compared with the LU-SSOR method. With the BJ-SSOR method (plus signs), the implicit coupling term is less often computed so that the extra CPU cost is reduced to 20%. The memory consumption remains identical, as the same information is stored though not computed at the same moment.

Finally, the BJ-SSOR scheme enables a fast convergence rate of the computations at a cost of about one-fifth more CPU time and 10% more memory requirements compared with the LU-SSOR method. Nevertheless, the time steps allowed are much larger and the TSM is far less sensitive to either high frequencies or an important number of harmonics than with explicit schemes. It can thus be concluded that the extra numerical cost of the implicitation is greatly counterbalanced by the larger time step enabled.

B. Pitching Wing

The quality of the presented implicit time spectral method is now studied. The BJ-SSOR method with 300 multigrid cycles and $l_{\max} = 2$ is retained in the following simulations of the LANN wing in forced harmonic oscillations. The time spectral method is compared with a reference U-RANS computation [20] on the same mesh shown in Fig. 1. A dual-time-stepping backward-difference-formula scheme advances the equations in time with 50 inner iterations that take advantage of the same acceleration techniques as for the previous TSM computations. Indeed, an implicit backward-Euler time integration method is used for the inner iterations. The resulting linear system is solved with a scalar LU-SSOR method. Four periods



a) 20.0% of wingspan

b) 32.5% of wingspan

c) 47.5% of wingspan

d) 65.0% of wingspan

e) 82.5% of wingspan

f) 95.0% of wingspan

Fig. 7 Time average of the wall wingspan pressure coefficient C_p.

of the flow are discretized by 30 time steps each, leading to 6000 multigrid cycles.

The TSM computations can be carried out in two ways: in a wing-relative or an absolute reference frame. For the first one, the mesh remains rigid around the wing and the variation of incidence is induced by different far-field boundary conditions applied at each instant. In this case, the inertial force has to be taken into account through a source term of the Navier–Stokes equations. In an absolute reference frame, the incidence variation is produced by deforming the mesh around the wing skin while the far-field boundary conditions remain fixed. In an arbitrary Lagrangian–Eulerian formulation, the deformation velocity of the mesh is introduced in the computation of the fluxes, Eq. (2), and as the cell volume V also varies in time, the TSM operator is applied on VW, leading to the following semidiscrete equation:

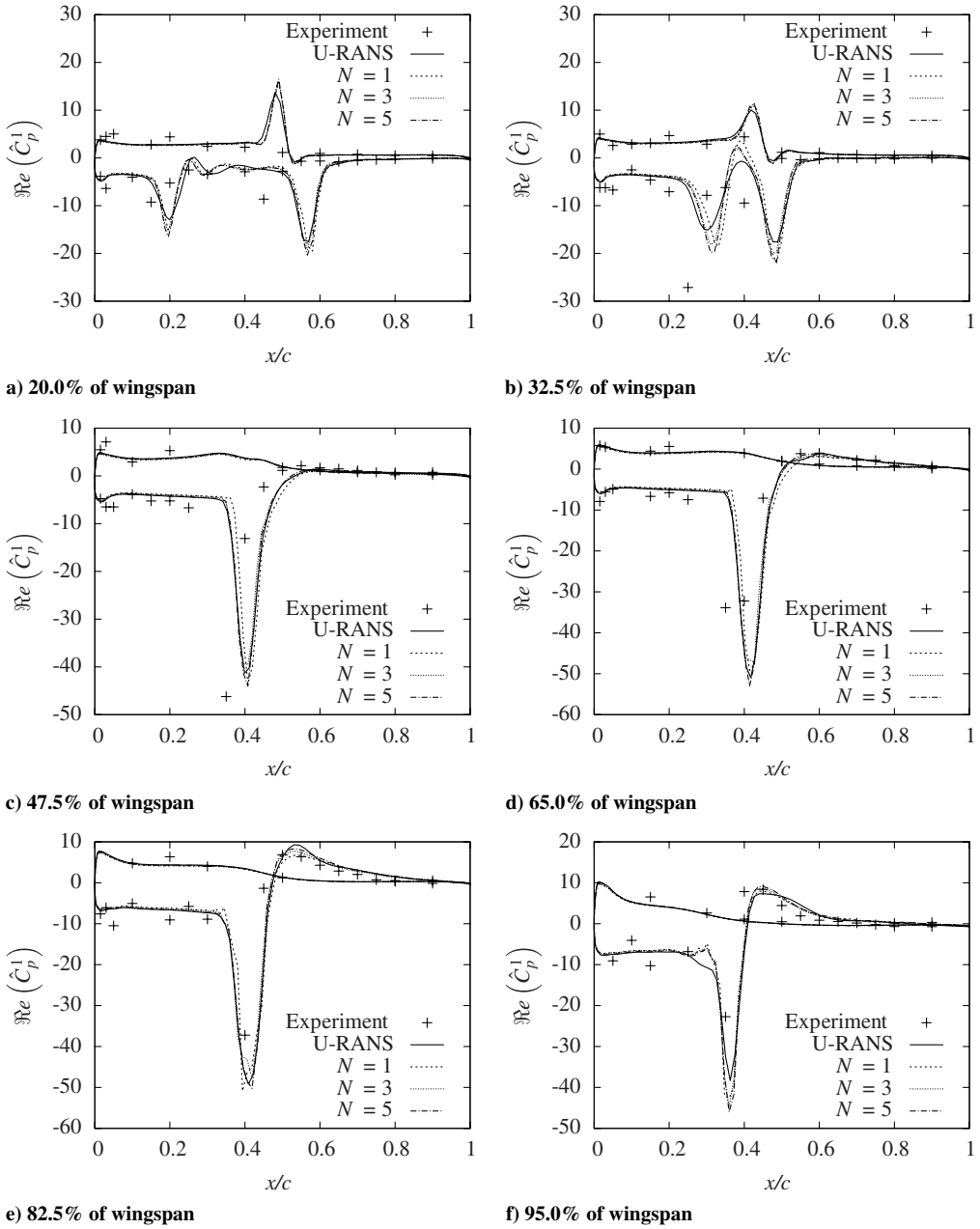
$$V_n \frac{\partial W_n}{\partial \tau_n} + R(W_n) + D_t(V_n W_n) = 0, \qquad 0 \le n < 2N + 1$$

Both methods lead to very close results and cannot be discriminated.

An instantaneous snapshot of the pressure coefficient C_p is presented for U-RANS and TSM in Fig. 6 at $\alpha = 0.6$ deg for increasing angle of attack. A λ shock is clearly visible near the wing root.

The Fourier analysis is conducted on six sections at 20.0, 32.5, 47.5, 65.0, 82.5, and 95.0% of the wingspan. The time-averaged part is presented in Fig. 7. A one-harmonic TSM computation is sufficient to match the U-RANS computation almost everywhere but at the shock location, where the solution slightly fluctuates. With higher harmonics ($N = 3$ and $N = 5$), TSM solutions match well the reference U-RANS simulation. Both kinds of simulation give solutions that match the experimental data quite well, although the shock on the upper surface is predicted downstream of the experimental location.

The real and imaginary parts of the first harmonic of the pressure coefficient are presented in Figs. 8 and 9. The differences are more pronounced, and it appears that one harmonic is not sufficient to match the U-RANS computation, as it shows a small phase lag and some over- and undershoots around the shock area. These drawbacks are removed with a three-harmonic TSM computation,



a) 20.0% of wingspan

b) 32.5% of wingspan

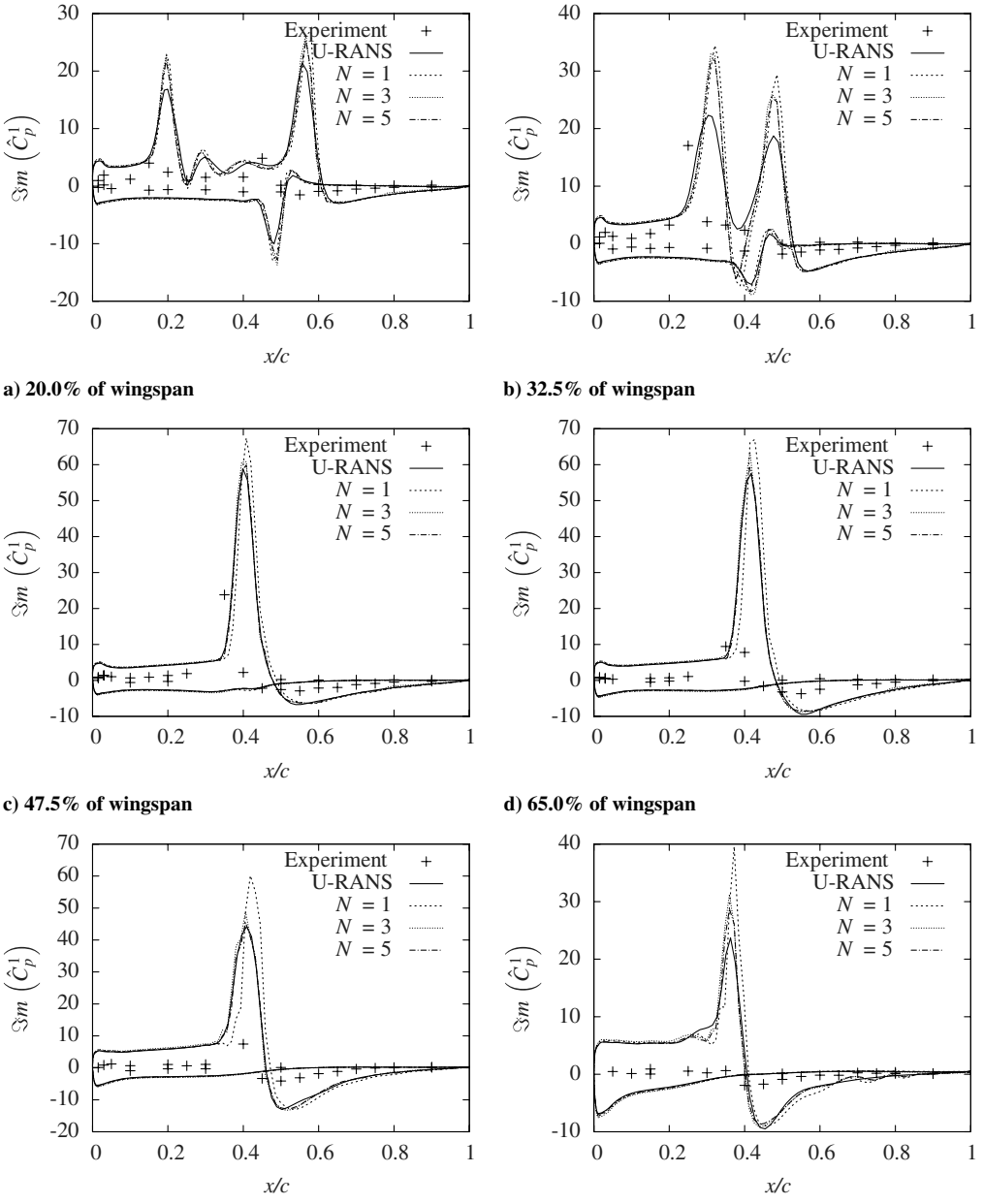
c) 47.5% of wingspan

d) 65.0% of wingspan

e) 82.5% of wingspan

f) 95.0% of wingspan

Fig. 8 Real part of the first harmonic of C_p.



a) 20.0% of wingspan **b) 32.5% of wingspan**

c) 47.5% of wingspan **d) 65.0% of wingspan**

e) 82.5% of wingspan **f) 95.0% of wingspan**

Fig. 9 Imaginary part of the first harmonic of C_p.

although some peaks are still sharp. A five-harmonic computation works better to lessen the irregularities at the peaks.

This point emphasizes the need for an accurate and efficient implicit formulation because all authors using an explicit time-step algorithm mention the difficulty of getting convergence with a higher number of harmonics. Overall, it can be concluded that a three-harmonic TSM computation is sufficient to match the U-RANS computation with engineering accuracy. In this case, the TSM is about 2.5 times faster than the reference U-RANS computation (see Fig. 10).

V. Conclusions

The time spectral method is dedicated to simulate time-periodic flows with a better efficiency than classical time-marching methods, that is, a quality of physics close to good U-RANS computations with a faster convergence rate. Up until now, the coupled steady computations have only been solved with explicit time marching, yielding in small time steps, further decreased by the stability criteria, which restricts the time step for high frequencies and for a large number of harmonics. In an industrial context, it was therefore

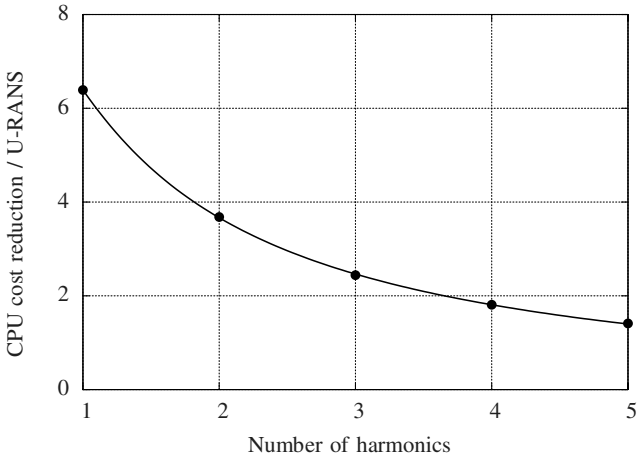


Fig. 10 CPU cost reduction of TSM compared with the reference U-RANS computation.

necessary to propose an alternative to the explicit time marching. Based on the LU-SSOR decomposition, new block-Jacobi implicit approaches have been derived to reduce the sensitivity of the method to high-frequency issues. Several solving processes have been tested to retain the most efficient approach. Finally, the BJ-SSOR approach enables better performances and faster convergence.

Our current effort concerns the extension of the time spectral method to turbomachinery. The frequencies met in these applications are much higher than in pitching wing flows and the present implicit treatments should ensure convergence. Even though the BJ-SSOR approach does not offer the best CPU performances, it will still be considered for turbomachine applications, as it ensures the strongest coupling.

Acknowledgments

This work has benefited from the generous support of the Direction des Programmes Aéronautiques Civils (French Civil Aviation Agency) as part of the Analyse Instationnaire des Turbomachines en aÉrodynamique et aCoustique program. The authors would also like to thank Societe Nationale d'Etude et de Construction de Moteurs d'Aviation for its active sponsoring. And finally, ONERA, the French aerospace laboratory and owner of the elsA solver, is greatly acknowledged for its scientific support.

References

[1] Hall, K. C., Thomas, J. P., Ekici, K., and Voytovich, D. M., "Frequency Domain Techniques for Complex and Nonlinear Flows in Turbomachinery," *33rd AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2003-3998, June 2003.

[2] McMullen, M., and Jameson, A., "The Computational Efficiency of Non-Linear Frequency Domain Methods," *Journal of Computational Physics*, Vol. 212, No. 2, March 2006, pp. 637–661. doi:10.1016/j.jcp.2005.07.021

[3] Verdon, J. M., and Caspar, J. R., "A Linearized Unsteady Aerodynamic Analysis for Transonic Cascades," *Journal of Fluid Mechanics*, Vol. 149, 1984, pp. 403–429. doi:10.1017/S002211208400272X

[4] Ning, W., and He, L., "Computation of Unsteady Flows Around Oscillating Blades Using Linear and Nonlinear Harmonic Euler Methods," *Journal of Turbomachinery*, Vol. 120, No. 3, July 1998, pp. 508–514.

[5] Hall, K. C., Thomas, J. P., and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, No. 5, May 2002, pp. 879–886. doi:10.2514/2.1754

[6] Gopinath, A., and Jameson, A., "Time Spectral Method for Periodic Unsteady Computations over Two- and Three- Dimensional Bodies," *43rd Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-1220, Jan. 2005.

[7] Gopinath, A., and Jameson, A., "Application of the Time Spectral Method to Periodic Unsteady Vortex Shedding," *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2006-0449, Jan. 2006.

[8] Spiker, M. A., Thomas, J. P., Kielb, R. E., Hall, K. C., and Dowell, E. H., "Modeling Cylinder Flow Vortex Shedding with Enforced Motion Using a Harmonic Balance Approach," *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials (SDM) Conference*, AIAA Paper 2006-1965, May 2006.

[9] Thomas, J. P., Dowell, E. H., and Hall, K. C., "Nonlinear Inviscid Aerodynamic Effects on Transonic Divergence, Flutter, and Limit-Cycle Oscillations," *AIAA Journal*, Vol. 40, No. 4, April 2002, pp. 638–646. doi:10.2514/2.1720

[10] van der Weide, E., Gopinath, A., and Jameson, A., "Turbomachinery Applications with the Time Spectral Method," *35th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2005-4905, June 2005.

[11] Gopinath, A., van der Weide, E., Alonso, J., Jameson, A., Ekici, K., and Hall, K., "Three-Dimensional Unsteady Multi-Stage Turbomachinery Simulations Using the Harmonic Balance Technique," *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2007-0892, Jan. 2007.

[12] McMullen, M., Jameson, A., and Alonso, J., "Acceleration of Convergence to a Periodic Steady State in Turbomachinery Flows," *39th Aerospace Sciences Meeting*, AIAA Paper 2001-0152, Jan. 2001.

[13] Yoon, S., and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," *AIAA 25th Aerospace Sciences Meeting*, AIAA Paper 87-0600, Jan. 1987.

[14] Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gas-Dynamic Equations with Applications to Finite Difference Methods," *Journal of Computational Physics*, Vol. 40, April 1981, pp. 263–293. doi:10.1016/0021-9991(81)90210-2

[15] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[16] Cambier, L., and Veuillot, J., "Status of the elsA Software for Flow Simulation and Multi-Disciplinary Applications," *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2008-0664, Jan. 2008.

[17] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," *AIAA 14th Fluid and Plasma Dynamic Conference*, AIAA Paper 81-1259, June 1981.

[18] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Transport Model for Aerodynamic Flows," *30th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 92-0439, Jan. 1992.

[19] Zwaan, R. J., "LANN Wing. Pitching Oscillation. Compendium of Unsteady Aerodynamic Measurements. Addendum No. 1," AGARD TR 702, 1982, Data Set 9.

[20] Delbove, J., "Full Navier–Stokes Unsteady Simulations: Application to Flutter Prediction: Contribution aux outils de Simulation Aéroélastique des Aéronefs: Prédiction du Flottement et Déformation Statique des Voilures," Ph.D. Thesis, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, 2005.

K. Powell
Associate Editor