A modular TSM solver for aeroelastic analysis and optimization

C. Liauzun, C. Blondeau

International Forum on Aeroelasticity and Structural Dynamics
IFASD 2024
17-21 June 2024, The Hague, The Netherlands

A MODULAR TSM SOLVER FOR AEROELASTIC ANALYSIS AND OPTIMIZATION

C. Liauzun[1] **and C. Blondeau**[1]

[1]DAAA, ONERA, Institut Polytechnique de Paris, 92320 Chatillon, France
cedric.liauzun@onera.fr
christophe.blondeau@onera.fr

Keywords: Harmonic Balance, Time Spectral Method, unsteady aerodynamics, gust response, adjoint TSM

Abstract: Several typical aeroelastic phenomena and instabilities, like flutter, induce periodic oscillations of the structure and of the aerodynamic forces. Numerical methods based on the harmonic balance technique or the Time Spectral Method (TSM) with a projection on the Fourier space has proven to be very efficient to predict the oscillatory phenomena by resolving the established regime without solving for the transient one. Such formulations lead however to critical numerical difficulties especially with a fine time sampling. A modular parallel TSM solver is implemented in a high level abstraction Python layer in order to perform aeroelastic analyses and optimizations of wings. This solver is in charge of performing all the operations regarding the temporal discretization and the time resolution. An interface with the CFD code elsA extracts all the needed information related to the spatial discretization. An Approximate Newton algorithm is used to solve the TSM problem. Both a Block-Jacobi method and a preconditioned GMRES are implemented to solve the implicit linear system. An adjoint solver is also developed in order to perform aeroelastic optimizations with dynamic objective functions and constraints. This TSM solver is evaluated for the NACA64A010 airfoil in transonic inviscid flows. Responses to harmonic pitching motions and to gusts are first computed and compared to unsteady simulations. The computation of gradients of the unsteady pressure drag with respect to shape parameters is then validated against finite differences.

1 INTRODUCTION

Several typical aeroelastic phenomena and instabilities, like flutter, induce periodic oscillations of the structure and of the aerodynamic forces. Numerous numerical methods have then been developed in order to compute such unsteady forces. The first ones such as the Doublet Lattice Method (DLM [1]) were based on a linearized potential equations with restrictive assumptions about the represented physics. Other methods based on the time consistent CFD (Euler or URANS) are the most accurate but at a very high computational cost. One of the most used time resolution implicit technique for such unsteady CFD is the Dual Time Stepping (DTS) Backward Difference Formula (BDF) proposed by Jameson [2]. Alternative techniques have also been developed such as the LFD (Linearized Frequency Domain) which consists in expressing the Euler or URANS equations in the frequency domain and in linearizing them about a steady state with respect to small perturbations of the flow field due to the harmonic excitation (forced motion or gust encounter) [3–9]. The increasing interest in nonlinear aeroelastic features led to the development of efficient nonlinear approaches dedicated to time periodic phenomena. Typically the Harmonic Balance Technique (HBT) considers a Fourier decomposition

in time of the flow. They have proven to be very efficient to predict the oscillatory phenomena by resolving the established periodic regime without solving for the transient one. The fluid equations to which is applied a Fourier Transform can be either solved in the frequency [10–12] or in the time domain (Time Spectral Method, TSM) [13–16]. TSM has become popular in the last decade due to the simplicity of its implementation in existing CFD codes. It has been applied to many aeroelastic problems [17, 18]. However, TSM can suffer from critical numerical difficulties, particularly when fine time sampling is considered. The time derivative operator indeed reinforces the off-diagonal terms of the TSM system matrix. Sicot et al. [19] proposed a pseudo-time marching strategy of the TSM problem with an implicit Block-Jacobi iterative algorithm to compute the solution increment. But this approach is restrained to CFL numbers inversely proportional to the number of harmonics. Work has been performed to circumvent this difficulty by using GMRES algorithms applied to the Jacobian matrix gathering all the instants [20]. Efficient preconditioners decoupling the space and time discretizations have also been proposed [21, 22].

Furthermore, since the TSM solves an unsteady problem using a steady process, an adjoint formulation aiming at high-fidelity optimization with unsteady objective functions and constraints can be derived more easily than a pure unsteady adjoint [23–25].

Blondeau et al. [26] showed the advantages of a modular implementation of a TSM solver and its adjoint counterpart using the elsA [1] CFD code [27]. This study proposes to evaluate the modular TSM implementation on the one hand to compute the aerodynamic response of an airfoil to forced motions and to gust profiles, and on the other hand to compute the unsteady aerodynamic coefficient derivatives with respect to shape parameters. This paper is organized as follows. The first part recalls theoretical aspects about the direct and adjoint TSM formulation. The second part is dedicated to the numerical experiments. More specifically, we will demonstrate the capability of the current implementation to predict the responses to a rigid body harmonic excitation and to a gust profile associated to the ALE formulation. Finally the third part is devoted to the assessment of the numerical efficiency of the GMRES solver applied to the TSM adjoint linear system and to the numerical validation of the assembled gradient compared to finite differences approximations.

2 THEORETICAL BACKGROUND

2.1 Direct flow solver

The semi-discrete governing fluid equations resulting from a Finite Volume discretization process can be written as

$$\frac{\partial \boldsymbol{\mathcal{V}} \mathbf{W}}{\partial t} + \mathbf{R}(\mathbf{W}, \mathbf{X}, t) = \mathbf{0} \tag{1}$$

where $\mathbf{W}$ is the vector of the fluid conservative variables, $\mathbf{X}$ is the vector of the mesh coordinates, t is time, $\boldsymbol{\mathcal{V}}$ is a diagonal matrix containing the cell volumes and $\mathbf{R}(\mathbf{W}, \mathbf{X}, t)$ is the discrete residual computed from the convective and diffusive fluxes and from the eventual source terms. In case of a deforming mesh, the Arbitrary Lagrangian Eulerian (ALE) formulation of the Euler/RANS equations is used, and the corresponding additional flux term is also included in $\mathbf{R}$.

[1] elsA is the joint property of Safran and ONERA

The TSM method is used to seek for time periodic phenomena of pulsation ω. It consists in first projecting the conservative variables and residual vector in the Fourier space and in truncating the Fourier Series to keep the first N harmonics. The application of the Inverse Fourier Transform to equation (1) written in the Fourier domain allows then the expression of the latter equation at the $M = 2N + 1$ instants of the time sampling of the reference period. Such a process yields the time-spectral approximation of the time derivative term at the n^{th} instant t_n.

$$\left.\frac{\partial \mathbf{W}}{\partial t}\right|_{t_n} = D_t(\mathbf{W}_n) = \sum_{j=0}^{M-1} d_{nj}\mathbf{W}_j \tag{2}$$

$$\text{with} \begin{cases} d_{nn} &= 0 \\ d_{nj} &= \frac{\omega}{2}(-1)^{(n-j)} \csc\left(\frac{\pi(n-j)}{M}\right) \end{cases}$$

The fluid equation at the n^{th} instant can then be written as a steady equation

$$|\boldsymbol{\mathcal{V}}_n|\frac{\partial \mathbf{W}_n}{\partial \tau_n} + \mathbf{R}_n + D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n) = \mathbf{0}, \quad 0 \leq n < M \tag{3}$$

where $\mathbf{W}_n = \mathbf{W}(t_n)$ is the vector of the fluid conservative variables at the n^{th} instant and $|\boldsymbol{\mathcal{V}}_n|$ is a diagonal matrix containing the cell volumes at instant t_n. In case of a prescribed motion, the cell volumes are assumed to be known at each instant. The pseudo-time term $|\boldsymbol{\mathcal{V}}_n|\frac{\partial \mathbf{W}_n}{\partial \tau_n}$ is added in order to stabilize the time integration scheme. It is discretized by a first-order approximation

$$|\boldsymbol{\mathcal{V}}_n|\frac{\Delta \mathbf{W}_n}{\Delta \tau_n} + \mathbf{R}_n + D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n) = \mathbf{0}, \quad 0 \leq n < M \tag{4}$$

where $\Delta\mathbf{W}_n = \mathbf{W}_n^{q+1} - \mathbf{W}_n^q$ is the solution increment between iterations q and $q+1$, and $\Delta\tau_n$ is the pseudo-time increment computed using a local time-stepping strategy based on a CFL number. The backward-Euler implicit scheme is derived by differentiating the residual and the time-spectral source term at iteration $q+1$. Let $\mathbf{J}_n = \partial\mathbf{R}_n/\partial\mathbf{W}_n$ be the Jacobian matrix of the residual vector, we have

$$\mathbf{R}_n^{q+1} \approx \mathbf{R}_n^q + \mathbf{J}_n\Delta\mathbf{W}_n \tag{5}$$

The time-spectral derivative operator D_t being linear it follows directly that

$$D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n^{q+1}) = D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n^q) + D_t(|\boldsymbol{\mathcal{V}}_n|\Delta\mathbf{W}_n) \tag{6}$$

Substituting (5) and (6) in (4) we obtain the fully implicit TSM formulation

$$\left(\frac{|\boldsymbol{\mathcal{V}}_n|}{\Delta\tau_n} + \mathbf{J}_n + D_t(|\boldsymbol{\mathcal{V}}_n|\,\cdot)\right)\Delta\mathbf{W}_n = -\mathbf{R}_n^q - D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n^q) = -\mathbf{R}_{TSM}(\mathbf{W}_n^q) \tag{7}$$

Gathering all flow instances in the vector $\mathbf{W} = (\mathbf{W}_0^T, \mathbf{W}_1^T, \cdots, \mathbf{W}_{M-1}^T)^T$, we can write the full TSM system as

$$[\mathbf{A}]\Delta\mathbf{W} = -\mathbf{R}_{TSM}(\mathbf{W}^q) \tag{8}$$

with the resulting Jacobian matrix given by

$$[\mathbf{A}] = \begin{pmatrix} \frac{|\boldsymbol{\mathcal{V}}_0|}{\Delta\tau_0} + \mathbf{J}_0 & |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_0^1 & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_0^{M-1} \\ |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_1^0 & \frac{|\boldsymbol{\mathcal{V}}_1|}{\Delta\tau_1} + \mathbf{J}_1 & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_1^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^0 & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^1 & \dots & \frac{|\boldsymbol{\mathcal{V}}_{M-1}|}{\Delta\tau_{M-1}} + \mathbf{J}_{M-1} \end{pmatrix} \tag{9}$$

where $\mathbf{d}_n^j = diag(d_{nj}) = d_{nj}\mathbf{I}$. In the expression above we have emphasized that the volume matrix $|\boldsymbol{\mathcal{V}}_n|$ may depend on the time instance for the case of an ALE formulation with a deforming mesh.

2.2 Implementation

A TSM solver requires to carry out several operations that can be divided into those relevant to a pure CFD simulation, i.e. all the operations due to the space discretization, and those related to the TSM approach itself, i.e. those related to the time discretization. A solver was then implemented in a high level Python abstraction layer with a low level interface to the kernel objects of the elsA CFD code. The Python program is in charge of performing the time integration scheme, of computing the temporal source terms by achieving the coupling between the different instants and of assembling the TSM residual ($\mathbf{R}_{TSM}$ in equation (7)). The CFD code should be able to provide the residual vector from a flow field ($\mathbf{R}_n$ in equation (7)), the Jacobian matrix or the matrix-vector product and the associated local time step vector. elsA was then modified to make these latter operators available from an external Python script. One of the main advantages of such a modular implementation is the ability to quickly and rather easily investigate different solving strategies, such as the Approximate Newton Krylov which has proved effective for solving steady problems [28, 29].

Furthermore, both time and space parallelizations were implemented, taking advantage of the native spatial parallelization of the CFD code. Because of the large number of degrees of freedom in the space-time problem (the number of time instances M times the number of degrees of freedom of the spatial problem), an efficient specific parallel implementation is indeed needed. In order to keep a fixed number of unknowns per process, the classical space discretization managed by the CFD code is augmented with a time parallelization as illustrated in figure 1. The distribution of the spatial unknowns over K processes is duplicated the number of instant times. The vector of the unknows of the whole space-time problem $\mathbf{W}$ is then distributed over a grid of $M \times K$ processes indexed by the couple (n, k) where $n \in [0, M-1]$ and $k \in [0, K-1]$. Parallel communications may occur either in the time direction via one of the K time communicators (yellow) or in the space direction via one of the M space communicators (blue). In order to minimize the communications, the time parallelization is carried out according to a "round robin" strategy. Each process receives data from only one CPU and sends data to only one other CPU.

Two numerical strategies were implemented to solve (8). The first one consists in solving equation (7) using an implicit iterative Block-Jacobi as described by Sicot et-al [19]. The solution increment at the q^{th} non-linear iteration for the n^{th} instant is computed iteratively (over k) from

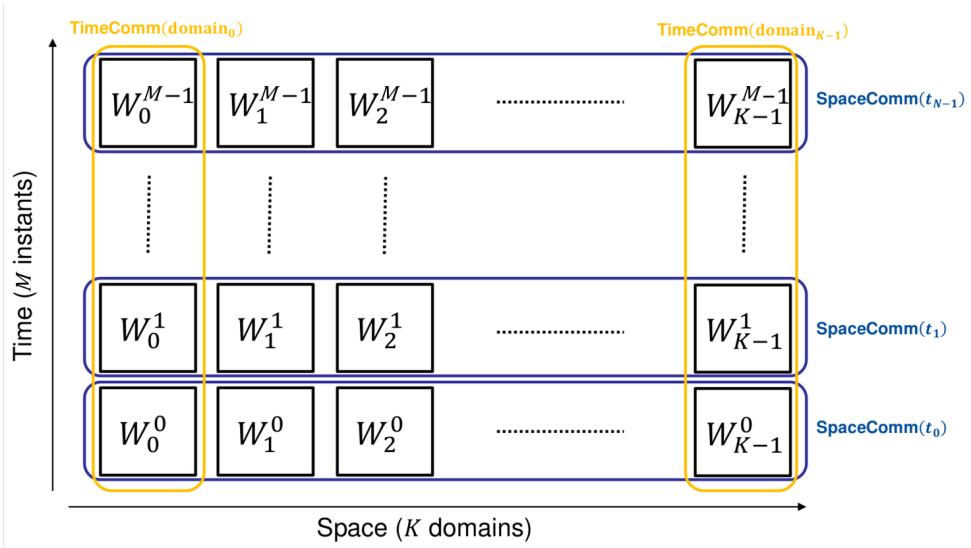


Figure 1: Distribution of the space-time vector $\mathbf{W}$ over $M \times K$ processes where $\mathbf{W}_k^m$ refers to the unknowns belonging to spatial domain k and the instant m. Parallel communications occurs only within one of the K temporal communicator (yellow) or one of the M space communicators (blue).

$$\left(\frac{|\boldsymbol{\mathcal{V}}|}{\Delta\tau_n} + \mathbf{J}_n\right) \Delta\mathbf{W}_n^{k+1} = -\mathbf{R}_n^q - |\boldsymbol{\mathcal{V}}| D_t(\mathbf{W}_n^q) - |\boldsymbol{\mathcal{V}}| D_t(\Delta\mathbf{W}_n^k) \tag{10}$$

matching the approximate global Jacobian matrix for all the instants

$$[\mathbf{A}_{BJ}] = \begin{pmatrix} \frac{|\boldsymbol{\mathcal{V}}_0|}{\Delta\tau_0} + \mathbf{J}_0 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \frac{|\boldsymbol{\mathcal{V}}_1|}{\Delta\tau_1} + \mathbf{J}_1 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \frac{|\boldsymbol{\mathcal{V}}_{M-1}|}{\Delta\tau_{M-1}} + \mathbf{J}_{M-1} \end{pmatrix} \tag{11}$$

According to this approach, the solution increment of one time instance can be computed independently of the other instants, the coupling between the instants being accounted for only in the right-hand side. System (10) can then be solved using a relaxation technique already implemented in the CFD code (LU-SSOR). Such an approach allows a quick resolution of each non-linear iteration. Nevertheless, the CFL number must remain low to converge towards a solution, especially for a high number of instants. The second implemented approach consists in handling straightforwardly the whole space-time vector $\mathbf{W}$ and in solving system (8) using a preconditioned F-GMRES (Flexible Generalized Minimal RESidual) algorithm. Such an approach improves the efficiency of the TSM problem resolution. The CFL number can indeed be inversely proportional to the residual norm, and make the non-linear resolution process tend towards a Newton algorithm with its quadratic convergence. However previous work showed that the convergence degrades rapidly when the number of instants and/or the excitation frequency increases, requiring thus many more preconditioning iterations to reach a converged solution. To alleviate this convergence degradation, authors proposed specific preconditioning techniques based on the inversion of the spatial-temporal diagonal blocks [30] associated with an approximate-factorization that decouples spatial and temporal systems [21], or based on a block-circulant matrix and a preconditioner-solving in the frequency domain [22]. The

latter was assessed in this study. The idea of the block-circulant preconditioner is based on the observation that the TSM Jacobian matrix (9) is "close to" block-circulant.

$$[\mathbf{A}] = \begin{pmatrix} \frac{|\boldsymbol{\mathcal{V}}_0|}{\Delta\tau_0} + \mathbf{J}_0 & |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_0^1 & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_0^{M-1} \\ |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_1^0 & \frac{|\boldsymbol{\mathcal{V}}_1|}{\Delta\tau_1} + \mathbf{J}_1 & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_1^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^0 & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^1 & \dots & \frac{|\boldsymbol{\mathcal{V}}_{M-1}|}{\Delta\tau_{M-1}} + \mathbf{J}_{M-1} \end{pmatrix}$$

"close to" (12)

$$\begin{pmatrix} \frac{|\boldsymbol{\mathcal{V}}_0|}{\Delta\tau_0} + \widetilde{\mathbf{J}} & |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_0^1 & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_0^{M-1} \\ |\boldsymbol{\mathcal{V}}_1|\mathbf{d}_1^0 & \frac{|\boldsymbol{\mathcal{V}}_1|}{\Delta\tau_1} + \widetilde{\mathbf{J}} & \dots & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_1^{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^0 & |\boldsymbol{\mathcal{V}}_{M-1}|\mathbf{d}_{M-1}^1 & \dots & \frac{|\boldsymbol{\mathcal{V}}_{M-1}|}{\Delta\tau_{M-1}} + \widetilde{\mathbf{J}} \end{pmatrix} = [\mathbf{A}_{BC}]$$

where the operator $\widetilde{\mathbf{J}}$ is introduced to enforce the circulant pattern. In order to be "as close as possible to" the instantaneous space Jacobian matrices $\mathbf{J}_i$, the matrix $\widetilde{\mathbf{J}}$ is chosen equal to the mean Jacobian matrix. One interesting property of the block-circulant matrices is that they are diagonalized by the Discrete Fourier Transform operator ($\mathbf{F}$).

$$[\mathbf{A}_{BC}] = \mathbf{F}^{-1}\widehat{[\mathbf{A}_{BC}]}\mathbf{F} \tag{13}$$

with

$$\begin{aligned} \widehat{[\mathbf{A}_{BC}]} = \quad \text{diag}\Bigg[& \frac{|\boldsymbol{\mathcal{V}}_0|}{\Delta\tau_0} + 0i\omega\mathbf{Id} + \widetilde{\mathbf{J}}, ..., \frac{|\boldsymbol{\mathcal{V}}_N|}{\Delta\tau_N} + Ni\omega\mathbf{Id} + \widetilde{\mathbf{J}}, \\ & \frac{|\boldsymbol{\mathcal{V}}_{N+1}|}{\Delta\tau_{N+1}} - Ni\omega\mathbf{Id} + \widetilde{\mathbf{J}}, ..., \frac{|\boldsymbol{\mathcal{V}}_{2N+1}|}{\Delta\tau_{2N+1}} - 1i\omega\mathbf{Id} + \widetilde{\mathbf{J}} \Bigg] \end{aligned}$$

with N being the number of harmonics ($M = 2N+1$), $\mathbf{Id}$ the identity matrix and $\mathbf{F}$ the Discrete Fourier Transform operator defined as

$$\begin{pmatrix} \widehat{\mathbf{W}}_0 \\ \vdots \\ \widehat{\mathbf{W}}_N \\ \widehat{\mathbf{W}}_{-N} \\ \vdots \\ \widehat{\mathbf{W}}_{-1} \end{pmatrix} = \mathbf{F} \begin{pmatrix} \mathbf{W}(t_0) \\ \vdots \\ \mathbf{W}(t_N) \\ \mathbf{W}(t_{N+1}) \\ \vdots \\ \mathbf{W}(t_{2N}) \end{pmatrix}$$

where the $\widehat{\mathbf{W}}_k$ are the discrete Fourier coefficients of the $\mathbf{W}$ variable. Applying $[\mathbf{A}]_{BC}$ as preconditioner can then be performed efficiently in the frequency domain, since it is a diagonal matrix in the Fourier space, before retrieving the time solution by applying the Inverse Fourier Transform.

At last, a parallel implementation of the F-GMRES algorithm is needed for a large number of instants or of mesh cells. The flexible GMRES from the PETSc library [31] was then used via its Python interface.

2.3 Adjoint solver

Let $\mathbf{p}$ be a set of n_p shape design parameters. We introduce the generic form of an objective function for a periodic response as the weighted sum $\mathcal{J} = \sum_{n=0}^{M-1} c_n \mathcal{J}_n(\mathbf{W}_n(\mathbf{p}), \mathbf{X}_n(\mathbf{p}))$. $\mathcal{J}$ is any scalar aerodynamic function of interest like drag, lift or moment coefficient.

One can note that the vectors of flow variables and grid coordinates at the $2N+1$ instants in the period appear in function $\mathcal{J}$. The sequences of flow fields $\mathbf{W}_n$ and of volume meshes $|\boldsymbol{\mathcal{V}}_n|$ directly dependent on the meshes coordinates $\mathbf{X}_n$ are linked by the discrete residual equations

$$\mathbf{R}_{TSM}(\mathbf{W}_n, \mathbf{X}_n) = \mathbf{R}_n(\mathbf{W}_n, \mathbf{X}_n) + D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n) = \mathbf{0} \tag{14}$$

where $D_t(|\boldsymbol{\mathcal{V}}_n|\mathbf{W}_n)$ is given by (2). In this section, applications of the ALE formulation for only a rigid entrainment velocity field are discussed. The mesh geometry $\mathbf{X}_n$ and the associated volume matrix $|\boldsymbol{\mathcal{V}}_n|$ are therefore taken constant in the following. Anyway, the extension to a general ALE formulation is straightforward. Direct differentiation of (14) with respect to a design parameter p gives

$$\frac{d\mathbf{R}_{TSM}}{dp} = \sum_{j=0}^{M-1} \frac{\partial \mathbf{R}_{TSM}}{\partial \mathbf{W}_j} \frac{d\mathbf{W}_j}{dp} + \frac{\partial \mathbf{R}_{TSM}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp} \tag{15}$$

A strong requirement here is that the total variation of the residual must vanish for any design parameter change, i.e. $d\mathbf{R}_{TSM}/dp = \mathbf{0}$. Inserting the definition of $\mathbf{R}_{TSM}$ in (15) leads to the following tangent linear system:

$$\mathbf{J}_n \frac{d\mathbf{W}_n}{dp} + |\boldsymbol{\mathcal{V}}| \sum_{j=0}^{M-1} d_{nj} \frac{d\mathbf{W}_j}{dp} = -\frac{\partial \mathbf{R}_n}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp} - \delta_p |\boldsymbol{\mathcal{V}}| \sum_{j=0}^{M-1} d_{nj} \mathbf{W}_j \tag{16}$$

where $\delta_p|\boldsymbol{\mathcal{V}}| = \frac{\partial |\boldsymbol{\mathcal{V}}|}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp}$ represents the matrix of analytic sensitivities of cell volumes to a change in p. One can recognize on the left-hand side the full Jacobian matrix $[\mathbf{A}]$ in (9).

The total derivative of the objective function is then written as

$$\frac{d\mathcal{J}}{dp} = \sum_n c_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp} + \sum_n c_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{W}_n} \frac{d\mathbf{W}_n}{dp} \tag{17}$$

The following equality also holds $\forall\, \boldsymbol{\lambda}_n \in \mathbb{R}^{n_a}$

$$c_n \boldsymbol{\lambda}_n^T \frac{d\mathbf{R}_{TSM}}{dp} = 0, \quad 0 \leq n < M \tag{18}$$

and combining with (15) and (17) leads to

$$\begin{aligned} \frac{d\mathcal{J}}{dp} = & \sum_n c_n \frac{\partial \mathcal{J}_n}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp} + \sum_n c_n \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{R}_{TSM}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{dp} \\ & + \sum_n c_n \left(\frac{\partial \mathcal{J}_n}{\partial \mathbf{W}_n} + \boldsymbol{\lambda}_n^T \mathbf{J}_n \right) \frac{d\mathbf{W}_n}{dp} + \sum_n c_n \boldsymbol{\lambda}_n^T \left(|\boldsymbol{\mathcal{V}}| \sum_j d_{nj} \frac{d\mathbf{W}_j}{dp} \right) \end{aligned} \tag{19}$$

The adjoint system is obtained by canceling the factor of $d\mathbf{W}_n/dp$ which defines the adjoint vectors $\boldsymbol{\lambda}_n$ as the solutions of the following coupled linear systems

$$\mathbf{J}_n^T\boldsymbol{\lambda}_n + |\boldsymbol{\mathcal{V}}| \sum_{j=0}^{M-1} d_{jn}\boldsymbol{\lambda}_j = -\left(\frac{\partial \mathcal{J}_n}{\partial \mathbf{W}_n}\right)^T, \quad 0 \leq n < M \tag{20}$$

The transpose of the system matrix (9) of the TSM solver naturally appears in the adjoint linear system. Using the skew-symmetric property of the time-spectral derivative matrix $(d_{nj})_{0\leq n,j<M}$ we have $d_{jn} = -d_{nj}$. Finally, the total derivative of the objective function turns out to be

$$\frac{d\mathcal{J}}{dp} = \sum_n c_n \left(\frac{\partial \mathcal{J}_n}{\partial \mathbf{X}} + \boldsymbol{\lambda}_n^T \frac{\partial \mathbf{R}_n}{\partial \mathbf{X}}\right)\frac{d\mathbf{X}}{dp} + \sum_n c_n \boldsymbol{\lambda}_n^T \left(\delta_p |\boldsymbol{\mathcal{V}}|\, D_t(\mathbf{W}_n)\right) \tag{21}$$

3 NUMERICAL EXPERIMENTS

Simulations using the TSM solver were performed with the 2D NACA64A010 symmetric airfoil. The flight conditions were defined according to the experimental results which are reported by the AGARD group in [32]. The *dynamic index 55* case was selected and corresponds to a transonic Mach number of 0.796. The flow conditions are reported in Table 1.

Table 1: Flow conditions for dynamic index 55.

M	0.796		Mach number
p_∞	133912	Pa	free stream pressure
p_t	203321	Pa	total pressure
q	59395	Pa	dynamic pressure
α_0	-0.21	$^\circ$	mean incidence

The fluid domain is discretized with a structured 2D C-mesh of 257x33 cells (fig 2). The inviscid Euler equations are considered for the fluid model. An upwind second order Roe spatial discretization scheme associated to a MUSCL reconstruction and a Van Albada limiter for the convective fluxes is used.

3.1 Forced motion

The TSM solver was first applied to compute the airfoil response to a prescribed harmonic pitching motion whose angle of attack is defined by the function

$$\alpha(t) = \alpha_0 + \widehat{\alpha}\sin(\omega t) \tag{22}$$

where α_0 is the mean angle of attack and $\widehat{\alpha}$ is the amplitude of the pitching motion. The motion amplitude and frequency were respectively 1.02° and 34.4Hz. These simulations were compared with time consistent ones using the Dual Time Stepping method. The phase portrait of the aerodynamic coefficients as functions of the angle of attack considering the whole time history of the reference unsteady analysis are plotted in Figure 3 below. The plain bold curves correspond to the established periodic response in the 8^{th} time period, meaning that the computational time for the 7 preceding transient periods has been spent for nothing. The colored markers correspond to the instantaneous values obtained for a varying number of harmonics.

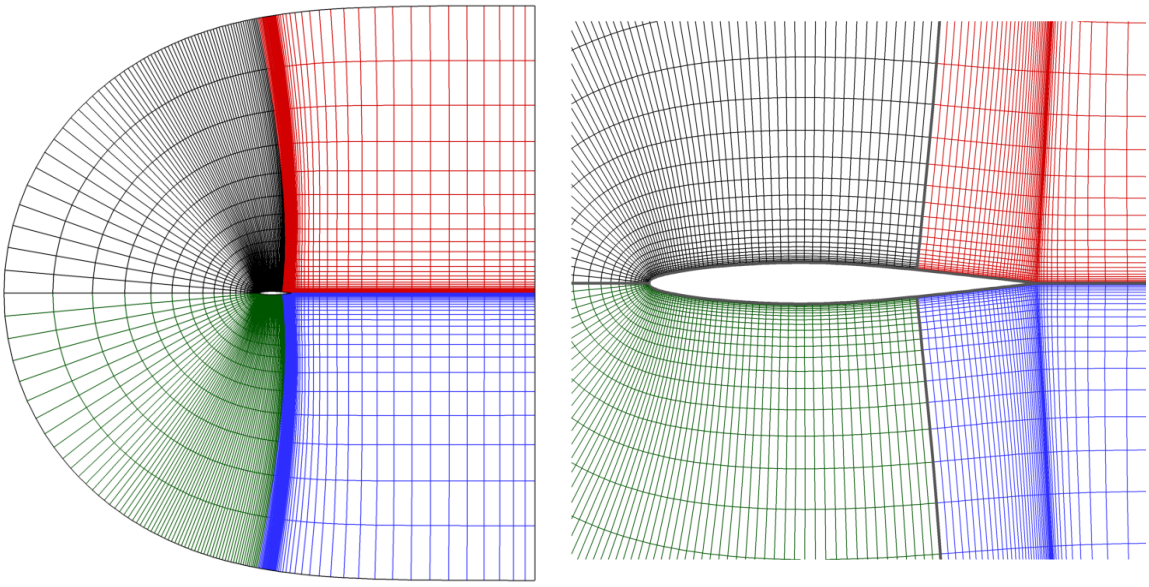


Figure 2: Structured 4-block 2D C-mesh of the *NACA64A010* airfoil.

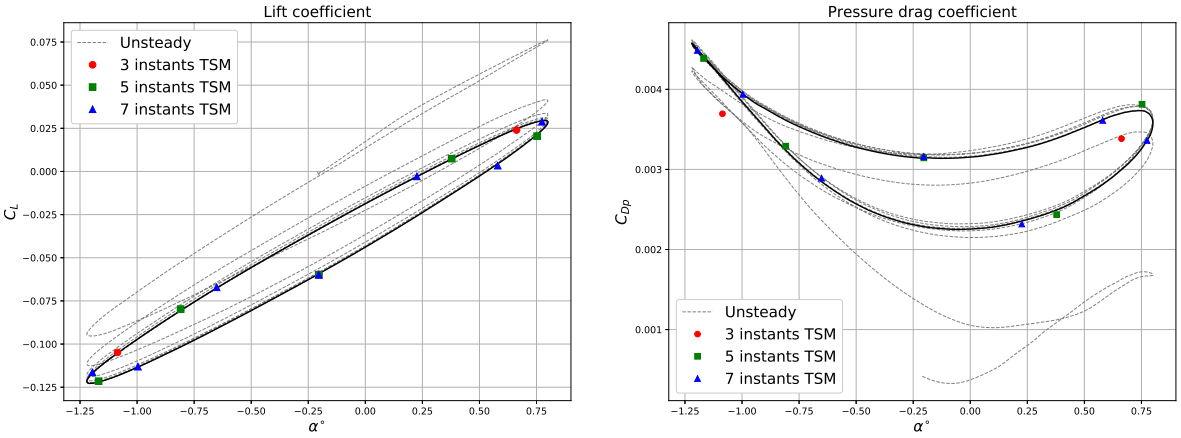


Figure 3: Aerodynamic coefficients versus angle of attack plots. The instantaneous values for a varying number of harmonics are superimposed to the reference unsteady time history.

One can notice that the lift evolution over one period is accurately predicted using the TSM solver with only one harmonic, whereas the prediction of the drag evolution requires at least 3 harmonics. These simulations were performed considering the excitation as a rigid rotation motion of the whole mesh. When considering the deforming mesh, a new mesh was computed for each instant applying the rotation only to the airfoil and using a grid deformation tool based on Trans-Finite Interpolation (TFI) techniques. The grid deformation speed is determined by applying the time-spectral derivative operator. The histories of the lift and drag coefficients over one period resulting from unsteady simulations and from TSM with 9 instants are plotted in Figure 5. The latter figure shows a good agreement between unsteady and TSM simulations for both rigid motions and deforming meshes.

Some investigations about the numerical algorithms were carried out. The TSM problem with 3 and 9 instants (1 and 4 harmonics) were solved using a Newton like non-linear algorithm with an additional pseudo time term (CFL). The resulting linear system to compute the solution

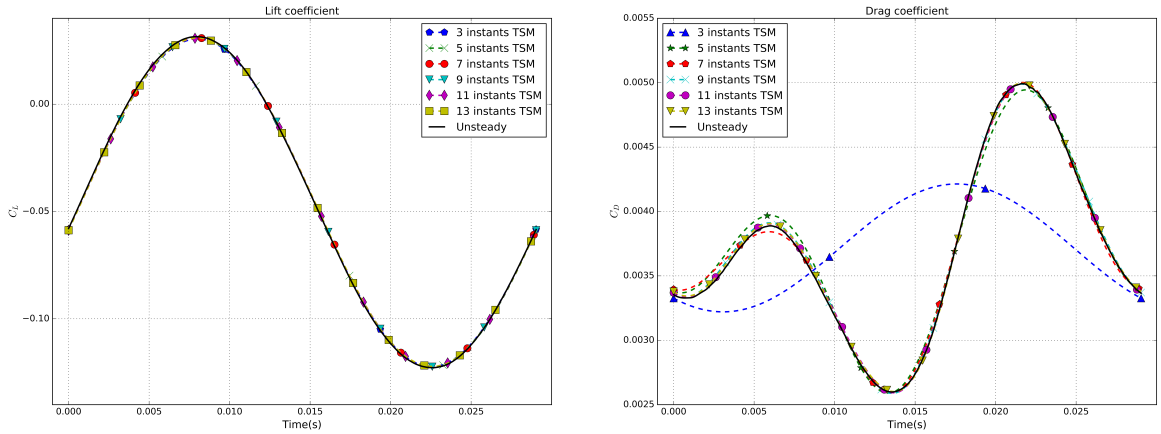


Figure 4: Accuracy of the time-spectral reconstruction for lift and drag coefficients for a spectral approximation from 1 to 6 harmonics (rigid motions).

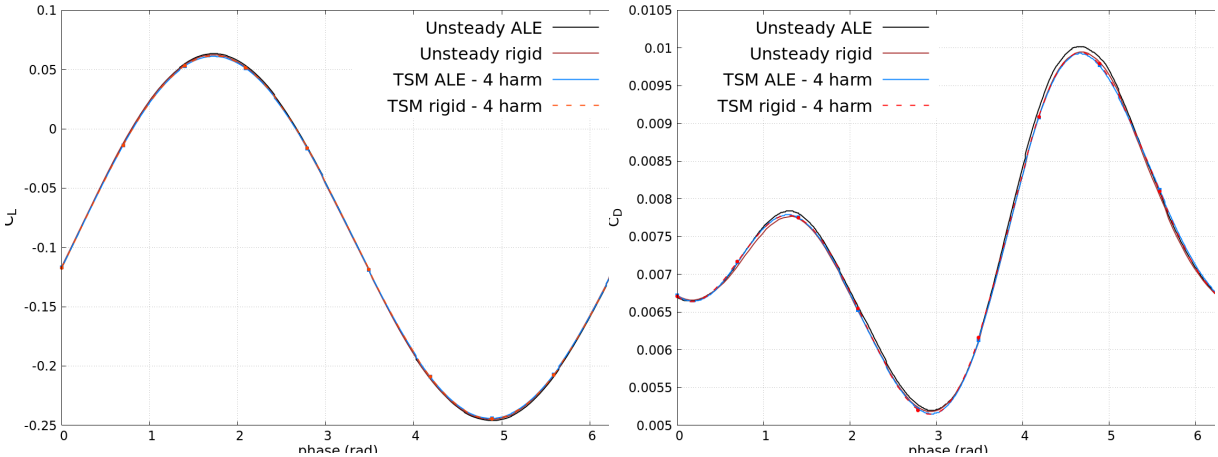


Figure 5: Lift and drag histories resulting from the unsteady and TSM with 4 harmonics (9 instants) simulations for both the motion applied rigidly and to a deforming mesh.

increment at each non-linear iteration was solved using either a Block-Jacobi algorithm or a preconditioned F-GMRES algorithm. In the first case, when the Block-Jacobi algorithm was used, the CFL number had to remain constant at a rather low value, i.e. no greater than 10 and 3 for 3 and 9 instants respectively, to reach a converged solution. The linear system which arises at each Jacobi sub-iteration was then partially solved using either a LU-SGS relaxation method or a preconditioned F-GMRES, yielding similar results. In the second case, when a preconditioned F-GMRES algorithm was applied to the whole system gathering the degrees of freedom of all instants, the CFL number was applied either constant or according to the residual norm-dependent function

$$CFL = CFL_0 * |\mathbf{R}|^{-r} \tag{23}$$

with $\mathbf{R}$ being the residual of the TSM problem and r the exponent equal to 1.2 for 3 instants and 1.1 for 9 instants. The F-GMRES algorithm was used with a Krylov basis of size 100 and a stopping criterion when the linear relative residual reached 0.05. The convergence profiles towards the solution are comparable for both rigid and deformable meshes and for an identical constant CFL number, whatever the resolution algorithm used. The CFL number has the strongest influence on the convergence speed even if the preconditioned F-GMRES is used to

solve the implicit linear system. Its evolution proportional to a power of the inverse of the residual norm allows indeed a rapid increase of itself and reducing thus drastically the number of needed non-linear iterations as can be seen in figure 6 plotting the residual history for 3 instants (left plot) and 9 instants (right plot) for both resolution algorithms.

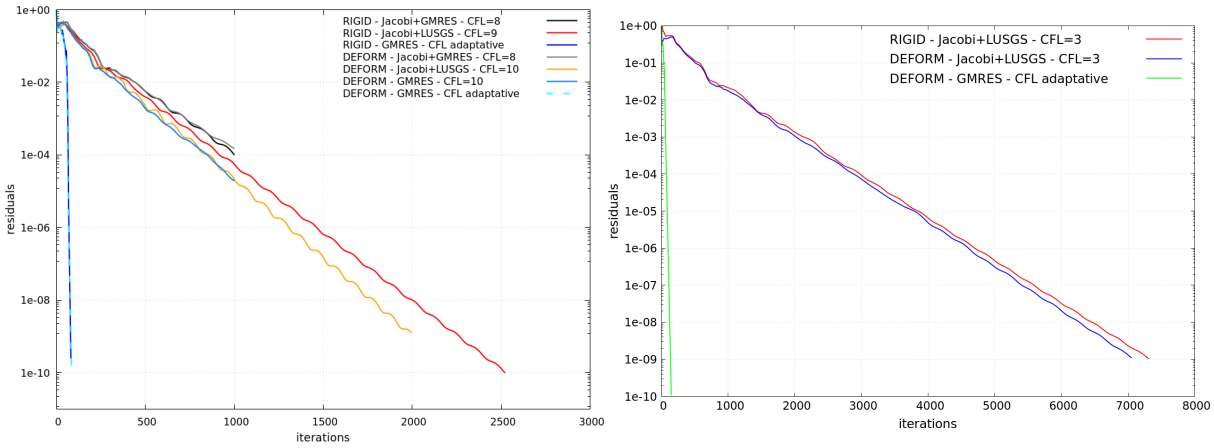


Figure 6: Residual evolutions for the TSM simulations with 3 (left) and 9 instants (right)

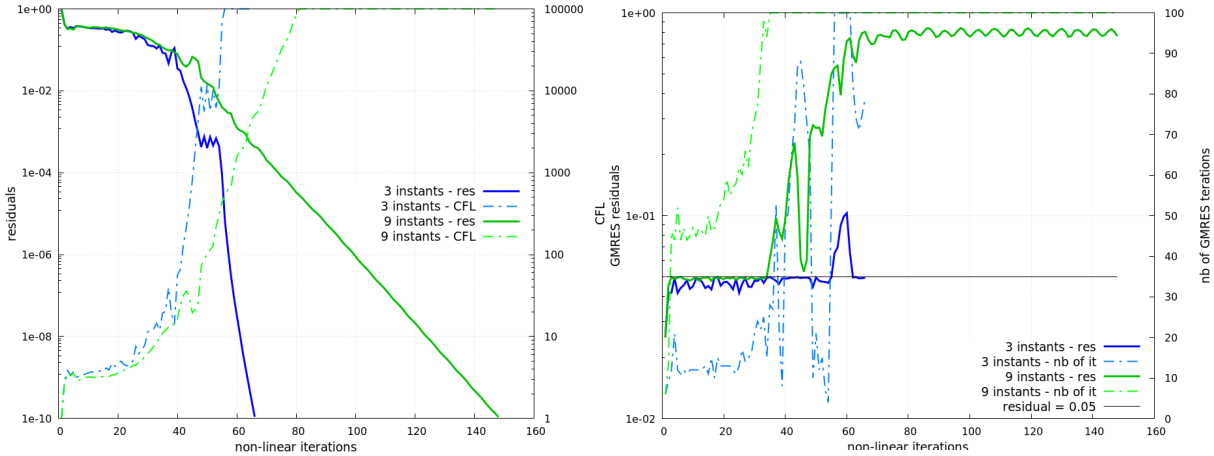


Figure 7: Evolutions of the residual, CFL (left), GMRES reached convergence and number of GMRES iterations (right) for the TSM simulations with 3 and 9 instants using the preconditioned F-GMRES

One can however notice a strong dependency of the number of instants on the convergence of the non-linear iterations. When the Block-Jacobi was used for the implicit linear system, the maximal CFL number decreased from 10 to 3 with the number of instants increasing from 3 to 9. And the number of non-linear iterations to reach a converged solution (residual norm less than 10^{-9}) increased from about 2000 to about 7000. When the preconditioned F-GMRES was applied to the implicit linear system with a CFL number varying according to 23, the number of non-linear iterations increased then from 64 to 137 as can be seen in 7 left. The latter figure plots the residuals and the CFL values with respect to the the non-linear iterations. Its right-wing counterpart plots the reached residuals and the number of linear iterations used by the GMRES algorithm function of the non-linear iterations. The different convergence behaviors shown by the TSM simulations with 3 (blue curves) and 9 instants (green curves) can be explained by the GMRES behavior. For simulations with 3 instants, the GMRES algorithm reached the requested residual reduction after a moderate number of GMRES iterations at each non-linear iteration. But with 9 instants, the GMRES used the maximum allowed number of iterations as soon as the CFL increased rapidly, and the GMRES residual remained close to

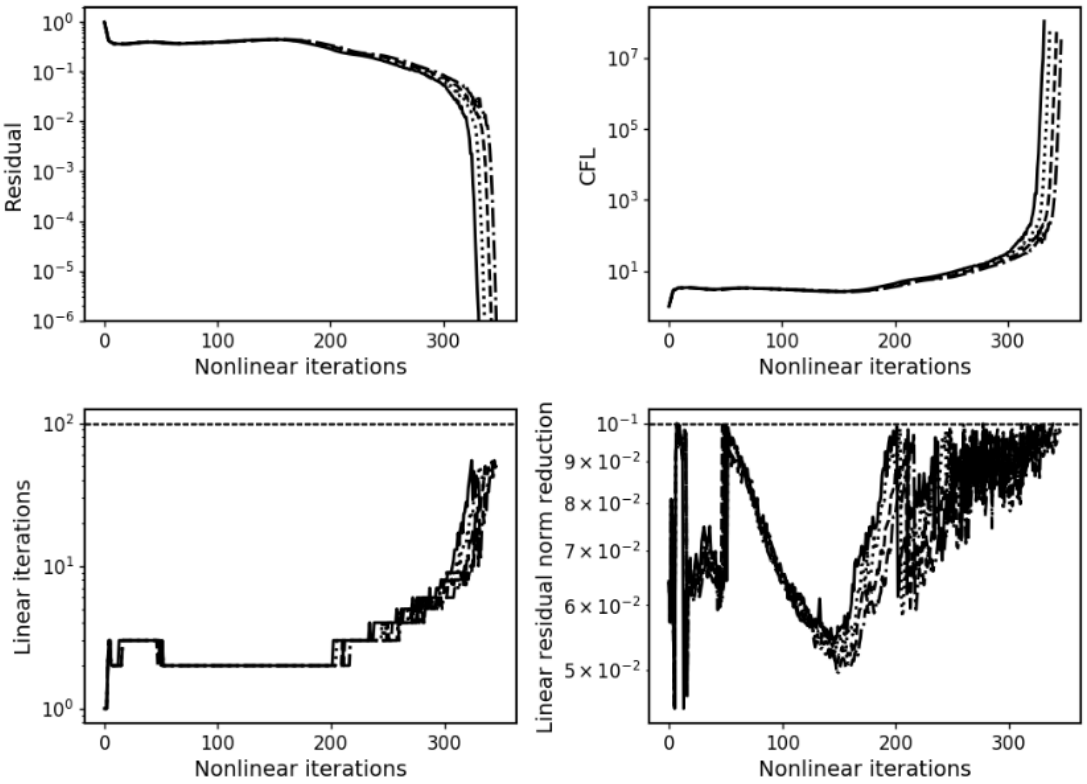


Figure 8: Convergence of the block-circulant preconditioned F-GMRES solution method. The norm of the nonlinear residual, the CFL number, the number of the GMRES iterations and the linear residual reduction are monitored for 3 (solid), 5 (dotted), 7 (dashed) and 9 (dash-dotted) instants.

1, yielding thus a poor solution increment. The Block-Jacobi method used as preconditioner seems therefore not efficient enough for high values of the CFL and the number of instants, conditions that lessen the diagonal dominance of the TSM system matrix and are thus unfavorable to the Block-Jacobi algorithm. To circumvent this dependency on the number of instants, the block-circulant preconditioner was assessed for 3, 5, 7 and 9 instants with a rigid motion of the airfoil. The simulations were performed with a requested linear residual drop of 10^{-1} and a CFL verifying 23 with a 10^{15} maximal CFL value. The evolution of the non-linear residual, the CFL number, the number of linear iterations used to reach the requested residual convergence, and the linear residual reduction are monitored in figure 8.As seen from the top left plot, the nonlinear residual decreases according to a typical pattern of pseudo-Newton method with first, a rather extended plateau with low CFL's, and second a rapid residual drop associated to high CFL's. The convergence curve is noticeably robust to the number of instants. Contrary to figure 7, the linear solver performs robustly for all number of instants (see bottom plots), with moderate numbers of GMRES iterations (at maximum about 50 for CFL of about 10^7). Clearly, this robustness of the linear solver to the number of instants is what enables the robust nonlinear convergence and the use of large CFL's yielding fast convergence rates.

3.2 Gust response

The TSM solver was also applied to compute the aerodynamic response of the airfoil to a gust. The latter excitation was implemented as an additional grid deformation speed according to the field velocity method proposed by Sitaraman et-al [33]. The first applied gust was a harmonic vertical gust whose velocity V_g is expressed as

$$V_{gust}(\mathbf{x}, t) = \frac{V_{gust}}{2} \left(1. - \cos(\mathbf{k} \cdot \mathbf{x} + \omega t)\right) \tag{24}$$

with V_{gust} being the gust amplitude, $\mathbf{k}$ the wave vector whose norm is $|\mathbf{k}| = \frac{2\pi}{\lambda}$ and λ the wave length, $\mathbf{x}$ the position vector and ω the gust pulsation. The applied gust here was characterized by a 10 ms^{-1} amplitude, and a wavelength equal to 25 times the chord matching a frequency of 10.73 Hz. Figure 9 plots the time evolution of the lift and drag coefficients computed using an unsteady simulation and TSM with 3 and 9 instants simulations. As for the previous responses to a forced pitching motion, the lift is accurately predicted using 3 instants whereas the drag prediction requires a greater number of instants. A good agreement can indeed be noticed between the drag evolution computed using both the unsteady and the TSM with 9 instants simulations. A one peak gust was also applied to the airfoil. Such a gust is characterized by a null speed excepted in the phase $(\mathbf{k} \cdot \mathbf{x} + \omega t)$ range $[-\lambda, 0]$ where the vertical velocity is determined by (24). Since the TSM solver requires a periodic excitation, the applied gust signal had a total length equal to 2 or 3 times the peak length (λ). The total signal length should indeed be selected such as the peak influence vanishes at the beginning and end of the time period, but also such as the response is captured with a reasonable number of instants while they are uniformly distributed along the input signal. Figure 10 shows the lift and drag evolution for both unsteady and TSM with 9 instants simulations. A rather good agreement can be noticed on both lift and drag, but the TSM computations are as expected strongly influenced by the length of the input signal and by the number of instants. The lift resulting from the TSM simulation with the longest input signal is closer to that resulting from the unsteady simulation than that resulting from the TSM simulation with the shortest signal. But the closest drag evolution is obtained for the shortest input signal.

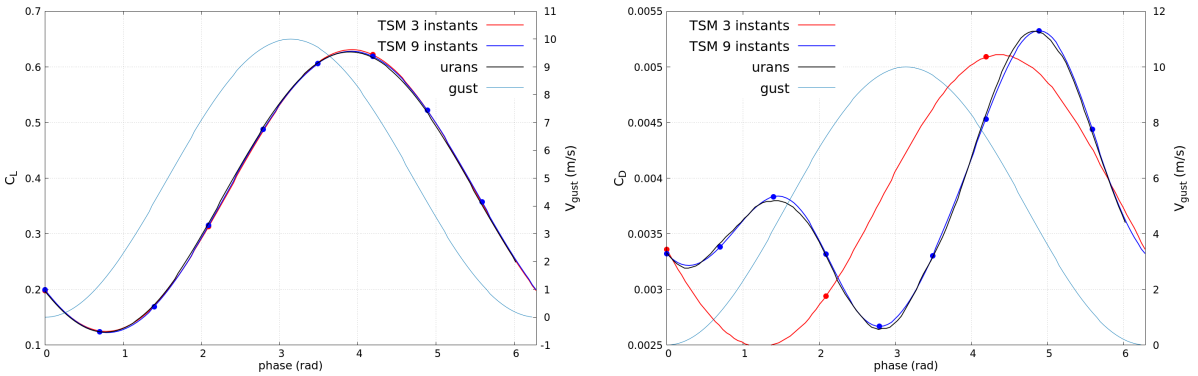


Figure 9: Lift and drag history over one period in response to a harmonic $(1 - \cos)$ gust

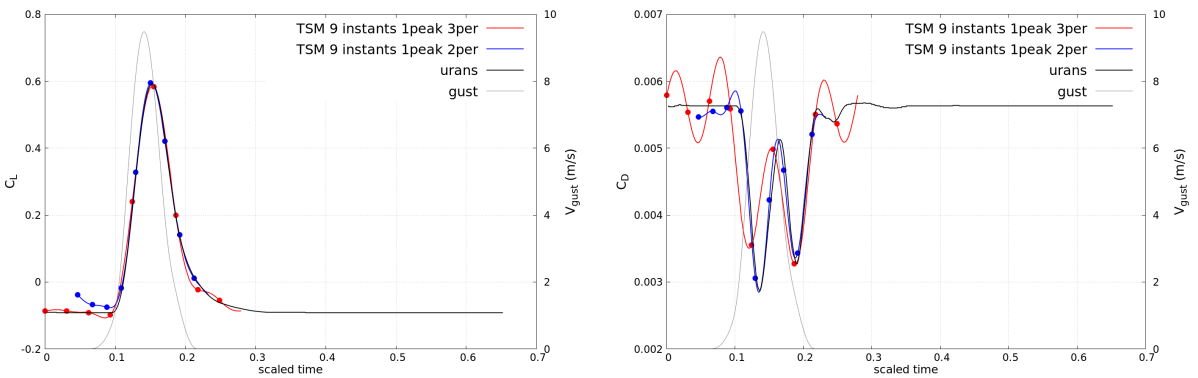


Figure 10: Lift and drag history in response to a one peak $(1 - \cos)$ gust

3.3 Gradient computation for shape optimization using an adjoint TSM solver

The TSM adjoint solver was applied to the same NACA64 2D test case (4-block structured C-type mesh) presented in the section devoted to the TSM solver. As already mentioned, the Block-Jacobi iterative solution strategy can no longer be applied because the adjoint system lacks of diagonal dominance. As pointed out by Mundis and Mavriplis in [21], with an increasing number of harmonics the full Jacobian matrix in (9) proceeds farther and farther from diagonal dominance. This results in linear systems of increasing complexity in terms of size and numerical stiffness. This undesirable property becomes problematic for relaxation-based iterative solvers whose convergence relies on diagonal dominance. To circumvent the need for diagonal dominance, Krylov subspace methods (e.g. the GMRES algorithm [34]) were considered in more recent works [35–39]. In practice, we decided to switch to more efficient Krylov solvers embedded in the *hpddm* package [40] of the linear algebra library PetSc [31]. The interested reader may refer to [41] for details about the nested Krylov strategy applied to CFD adjoint systems. Whatever, the linear solution method used (i.e. stationary methods or Krylov methods), a preconditioner is always needed to decrease the condition number of the system and allow for better convergence. In the present work, we focus on the so-called Block-Jacobi preconditioner. However, several preconditioning strategies have been proposed in the literature, among which the *STI* (Spatial-Temporal diagonal-block Inversion) preconditioner [38], the *approximate-factorization* preconditioner [36], the *block-circulant* preconditioner [[22], Chap. 3]. These alternative preconditioning strategies for solving adjoint problems will be evaluated in the near future.

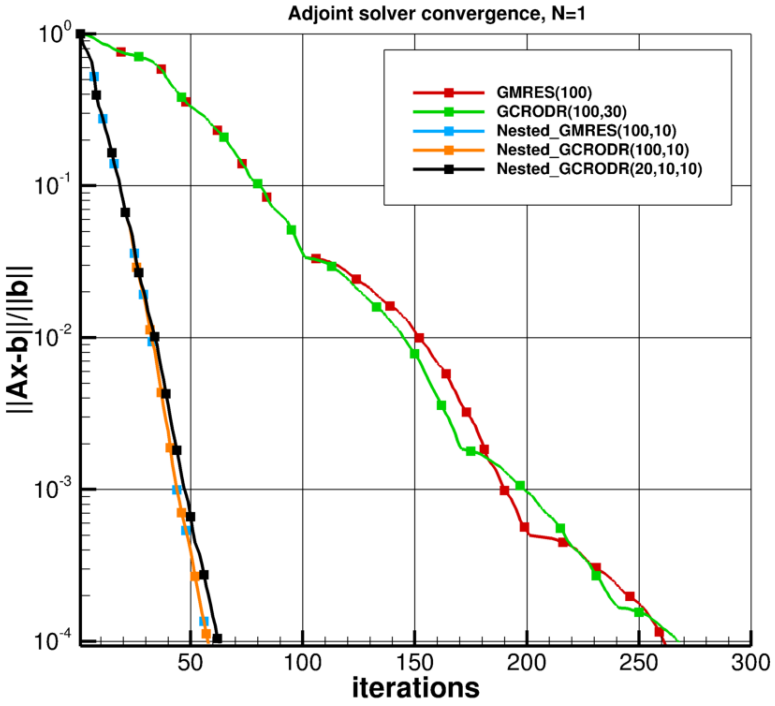


Figure 11: Convergence of various Krylov solvers applied to the TSM adjoint linear system, for a single harmonic spectral approximation. Stationary preconditioned GCRO-DR and GMRES-DR perform similarly. The same conclusion is drawn for nested Krylov solvers. For this simple problem, even a small outer Krylov subspace of 20 vectors turns out to be very efficient.

Various advanced Krylov solvers with deflation and restarting like GMRES-DR (Generalized Minimal Residual Method with Deflated Restarting) and GCRO-DR (Generalized Conjugate

Residual method with inner Orthogonalization and Deflated Restarting) combined with stationary or iterative nonlinear preconditioners have been tested. The associated convergence histories of the relative residual norm are plotted in Figure 11. GMRES-DR and GCRO-DR perform equivalently, while their flexible counterparts exhibit a speedup factor of about 5. The numerical parameters of the Krylov solvers are reported in the figure legend.

The efficiency of the Krylov solver is then evaluated for an increasing number of harmonics in the spectral approximation of the unsteady solution. In Figure 12 the associated residual norm convergence history of the GMRES-DR and FGMRES-DR adjoint solvers is plotted. Clearly the Block-Jacobi preconditioner, even if associated to a nested Krylov solver, cannot achieve wave number independence in terms of iterations. As mentioned previously, designing such a preconditioner is still an active area of research. Reusing existing building blocks was our main goal in setting up this TSM modular adjoint solver prototype and improvements will be considered in the near future regarding the preconditioning strategy.

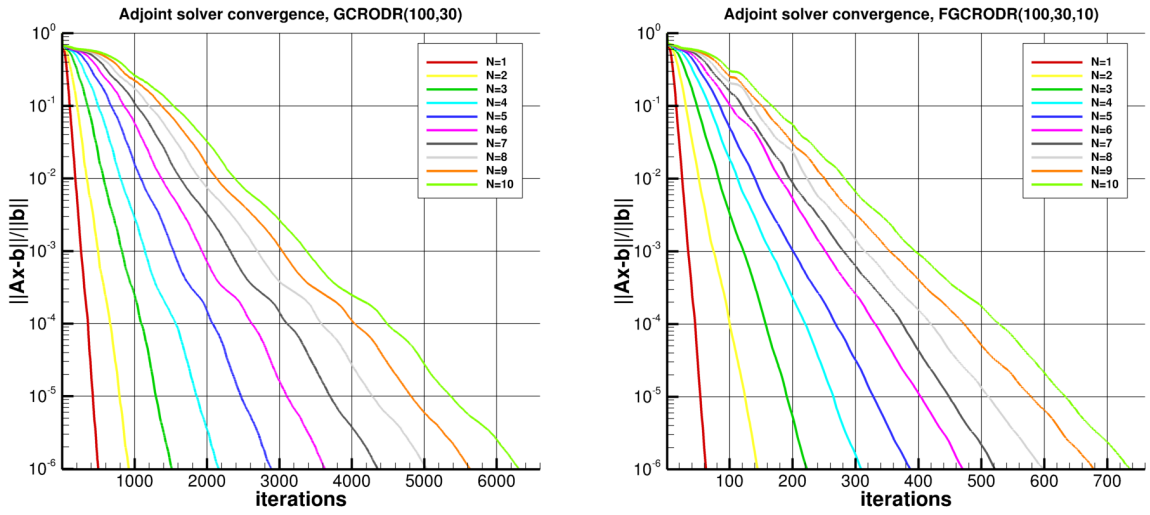


Figure 12: Convergence of the standard (left) and nested (right) GCRO-DR Krylov solvers applied to the TSM adjoint linear system for an increasing number of harmonics. The maximum size of the outer Krylov subspace is 100, the size of the recycled subspace is 30 and for the flexible solver the inner Krylov subspace is limited to 10 vectors.

As an example, Figure 13 presents the primal and adjoint density flow solutions at the three discretization instants of the single-harmonic approximation of the unsteady motion of our NACA64 test case.

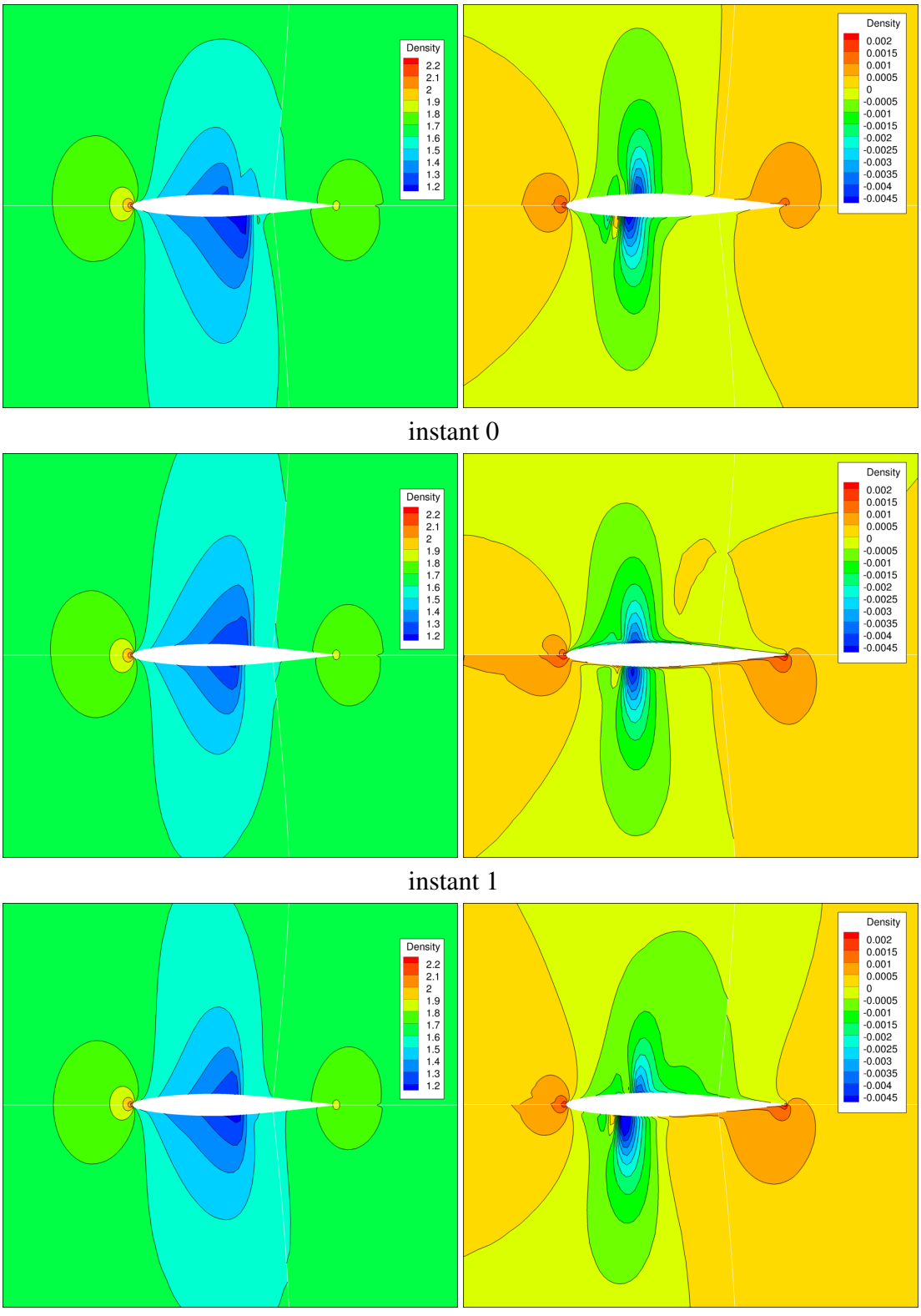


Figure 13: Primal (left) and adjoint (right) density flow solutions for a 1-harmonic spectral approximation of the forced harmonic pitching motion of the NACA64 airfoil.

The basic building blocks of the TSM adjoint solver have been validated numerically by carrying out a systematic comparison with finite difference approximations. In addition, the duality checks between the full primal TSM system matrix and its adjoint counterpart show a relative

precision of 8 significant digits. To further assess the accuracy of the assembled TSM adjoint total derivatives, we have implemented a simple parameterization of the NACA64 airfoil shape. Our objective is to compare with gradients of the aerodynamic pressure drag coefficient C_{Dp} obtained by finite differences. The parameterization is performed using the PADGE CAD modeler kindly provided by Airbus in the context of UE funded NEXTAIR project. The PADGE framework also provides the wall grid sensitivities with respect to the design variables. We can then easily apply a volume mesh deformation operator to this quantity to obtain the volume mesh derivative $d\mathbf{X}/dp$ required for the assembly of the total gradient of the objective function (see equation (21)).

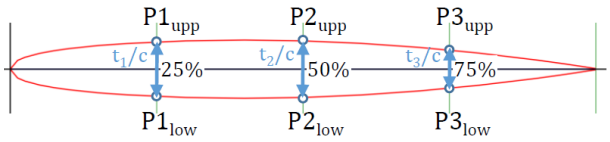
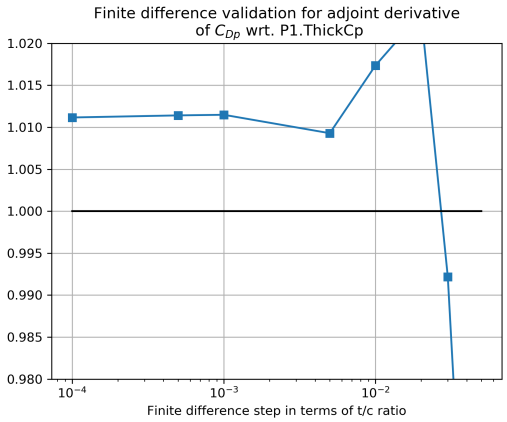
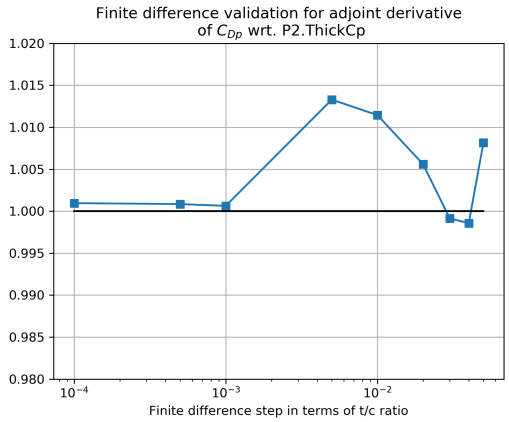


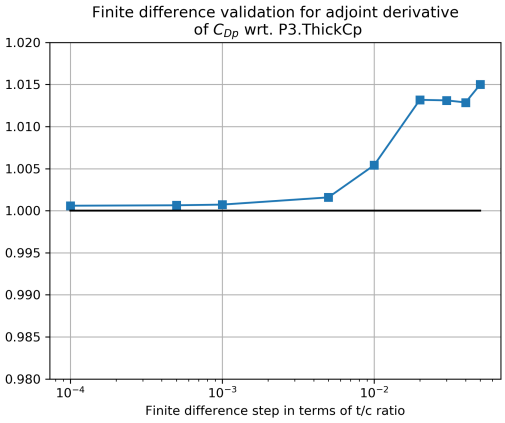
Figure 14: Simple shape parameterization of the NACA64 airfoil for validation of adjoint derivatives. Three design variables control the thickness-to-chord ratio at sections located at 25, 50 and 75% of the chord.



(a) t/c at 25%



(b) t/c at 50%



(c) t/c at 75%

Figure 15: Finite difference validation for the 3 instant time-average adjoint derivative of C_{Dp}. The three parameters control the relative thickness of the airfoil at 25, 50 and 75% of chord respectively. Blue plain curves correspond to FD derivatives scaled by adjoint ones.

We first compute the TSM solution using a single harmonic for the spectral approximation (i.e.

3 instants in the temporal period). To get the best accuracy, we ask for a relative decrease of 10 orders of magnitude for the residuals of the TSM primal and adjoint solvers. In figure 15 the time-average derivatives of the pressure drag coefficient obtained by the finite differences and the adjoint strategies are compared. Reported values are scaled by the adjoint ones. The latter then appear as horizontal lines at a constant unit value. Typically, the finite difference approximations show good convergence for increments lower than $1. \times 10^{-3}$. In this range, a very good agreement is observed with less that 1% of maximum discrepancy.

4 CONCLUSION

A modular parallel TSM solver was developed in order to perform aeroelastic analyses and wing shape optimizations. This solver is in charge of performing all the operations regarding the time discretization and the time resolution. An interface with the CFD elsA code extracts all the needed information related to the space discretization. The time resolution is carried out using an ANK method. Such a modular architecture allows the adaptation of the TSM solver to any CFD code (structured, unstructured), and most of all allows assessing and developing easily new resolution algorithms to improve the robustness and the computational efficiency. An adjoint formulation of the TSM approach was also developed in order to perform aeroelastic optimization with dynamic objective functions. This TSM solver and its adjoint formulation were evaluated to determine the response of an airfoil to forced pitching motions and to compute the gradient of the unsteady drag coefficient with respect to shape parameters. Analyses were carried out for both rigid rotations or forced motion applied to a deforming mesh. Simulations were also performed to investigate the response to gusts. This TSM approach was developed aiming at replacing the low fidelity gust load computation in an aeroelastic sizing and optimization process. The next step will then consist in extending the TSM solver capabilities to 3D wings and to coupled fluid-structure analyses and gradient computations.

5 REFERENCES

[1] Albano, E. and Rodden, W. (1969). A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. *Journal of Aircraft*, 7(2), 279–285.

[2] Jameson, A. (1991). Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *10th Computational Fluid Dynamics Conference*.

[3] Hall, K. C., Clark, W. S., and Lorence, C. B. (1994). A Linearized Euler Analysis of Unsteady Transonic Flows in Turbomachinery. *Journal of Turbomachinery*, 116(3), 477–488.

[4] Badcock, K., Timme, S., Marques, S., et al. (2011). Transonic aeroelastic simulation for instability searches and uncertainty analysis. *Progress in Aerospace Sciences*, 47(5), 392–423. ISSN 0376-0421. doi:https://doi.org/10.1016/j.paerosci.2011.05.002.

[5] Da Ronch, A., McCracken, A. J., Badcock, K. J., et al. (2013). Linear frequency domain and harmonic balance predictions of dynamic derivatives. *Journal of Aircraft*, 50(3), 694–707. doi:10.2514/1.C031674.

[6] Thormann, R. and Widhalm, M. (2013). Linear frequency domain prediction of dynamic response data for viscous transonic flows. *AIAA Journal*, 51(11), 2540–2557.

[7] Widhalm, M. and Thormann, R. *Efficient Evaluation of Dynamic Response Data with a Linearized Frequency Domain Solver at Transonic Separated Flow Condition*. doi: 10.2514/6.2017-3905.

[8] Liauzun, C., Canonne, E., and Mortchéléwicz, G. D. (2008). Flutter numerical computations using the linearized navier-stokes equations. In *NATO RTO AVT-154 Meeting Advance Methods in Aeroelasticity*.

[9] Dufour, G., Sicot, F., Puigt, G., et al. (2010). Contrasting the harmonic balance and linearized methods for oscillating-flap simulations. *AIAA journal*, 48(4), 788–797.

[10] McMullen, M. S. and Jameson, A. (2006). The computational efficiency of non-linear frequency domain methods. *Journal of Computational Physics*, 212(2), 637–661.

[11] Jameson, A., Alonso, J., and McMullen, M. (2002). Application of a non-linear frequency domain solver to the euler and navier-stokes equations. In *40th AIAA Aerospace Sciences Meeting & Exhibit, Reno, NV*. pp. AIAA 02–0120.

[12] Vilmin, S., Lorrain, E., Tartinville, B., et al. (2013). The nonlinear harmonic method: from single stage to multi-row effects. *International Journal of Computational Fluid Dynamics*, 27(2), 88–99. doi:10.1080/10618562.2012.752074.

[13] Hall, K. C., Thomas, J. P., and Clark, W. S. (2002). Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. *AIAA journal*, 40(5), 879–886.

[14] Hall, K., Thomas, J., Ekici, K., et al. (2003). Frequency domain techniques for complex and nonlinear flows in turbomachinery. In *33rd AIAA Fluid Dynamics Conference and Exhibit*. p. 3998.

[15] Gopinath, A. and Jameson, A. (2005). Time spectral method for periodic unsteady computations over two-and three-dimensional bodies. In *43rd AIAA aerospace sciences meeting and exhibit*, AIAA 2005-1220.

[16] Gopinath, A. and Jameson, A. (2006). Application of the time spectral method to periodic unsteady vortex shedding. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2006-0449.

[17] Thomas, J. and Dowell, E. (2018). *A Fixed Point Iteration Approach for Harmonic Balance Based Aeroelastic Computations*. doi:10.2514/6.2018-1446.

[18] Yao, W. and Marques, S. (2018). A harmonic balance method for nonlinear fluid structure interaction problems. *Computers & Structures*, 201, 26–36. ISSN 0045-7949. doi:https://doi.org/10.1016/j.compstruc.2018.02.003.

[19] Sicot, F., Puigt, G., and Montagnac, M. (2008). Block-jacobi implicit algorithms for the time spectral method. *AIAA Journal*, 46(12), 3080–3089.

[20] Mundis, N. L. and Mavriplis, D. J. (2014). An Efficient Flexible GMRES Solver for the Fully-coupled Time-spectral Aeroelastic System. (January), 1–25. doi:10.2514/6.2014-1427.

[21] Mundis, N. L. and Mavriplis, D. J. (2017). Toward an optimal solver for time-spectral fluid-dynamic and aeroelastic solutions on unstructured meshes. *Journal of Computational Physics*, 345, 132–161.

[22] Moulin, J. (2020). *On the flutter bifurcation in laminar flows : linear and nonlinear modal methods*. Ph.D. thesis. Thèse de doctorat dirigée par Sipp, Denis et Marquet, Olivier Mécanique des fluides et des solides, acoustique Institut polytechnique de Paris 2020.

[23] Mader, C. A. and RA Martins, J. R. (2012). Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. *AIAA journal*, 50(12), 2809–2819.

[24] He, S., Jonsson, E., Mader, C. A., et al. *Aerodynamic Shape Optimization with Time Spectral Flutter Adjoint*. doi:10.2514/6.2019-0697.

[25] Huang, H. and Ekici, K. (2014). A discrete adjoint harmonic balance method for turbomachinerty shape optimization. *Aerospace Science and Technology*, 39, 481–490.

[26] Blondeau, C. and Liauzun, C. (2019). A modular implementation of the time spectral method for aeroelastic analysis and optimization on structured meshes. In *International Forum on Aeroelasticity and Structural Dynamics, Savannah, Geogia, USA*.

[27] Cambier, L., Heib, S., and Plot, S. (2013). The onera elsa cfd software: input from research and feedback from industry. *Mechanics & Industry*, 14(3), 159–174.

[28] Yildirim, A., Kenway, G. K., Mader, C. A., et al. (2019). A jacobian-free approximate newton-krylov startup strategu for rans simulations. *Journal of Computational Physics*, 397, 108741. ISSN 0021-9991. doi:https://doi.org/10.1016/j.jcp.2019.06.018.

[29] Chisholm, T. T. and Zingg, D. W. (2009). A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows. *Journal of Computational Physics*, 228, 3490–3507.

[30] Mundis, N. L. and Mavriplis, D. J. (2015). Wave-number Independent Preconditioning for GMRES for Time-spectral Solvers. In *AIAA SciTech Forum, Kissimmee, Florida, January*.

[31] Balay, S., Abhyankar, S., Adams, M. F., et al. (2023). PETSc Web page. `https://petsc.org/`.

[32] Davis, S. S. (1982). NACA 64A010 (NASA Ames model) oscillatory pitching. *AGARD report*, 702.

[33] Sitaraman, J., Baeder, J., and Iyengar, V. (2003). *On the Field Velocity Approach and Geometric Conservation Law for Unsteady Flow Simulations*. doi:10.2514/6.2003-3835.

[34] Saad, Y. and Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3), 856–869.

[35] Woodgate, M. A. and Badcock, K. J. (2007). Fast prediction of transonic aeroelastic stability and limit cycles. *AIAA Journal*, 45(6), 1370–1381. ISSN 00011452. doi:10.2514/1.25604.

[36] Mundis, N. L. and Mavriplis, D. J. (2017). Toward an optimal solver for time-spectral fluid-dynamic and aeroelastic solutions on unstructured meshes. *Journal of Computational Physics*, 345(April), 132–161. ISSN 10902716. doi:10.1016/j.jcp.2017.04.067.

[37] Su, X. and Yuan, X. (2010). Implicit solution of time spectral method for periodic unsteady flows. *International Journal for Numerical Methods in Fluids*, 63(7), 860–876. ISSN 02712091. doi:10.1002/fld.2111.

[38] Mundis, N. L. and Mavriplis, D. J. (2015). Wave-number Independent Preconditioning for GMRES Time-spectral Solvers. *53rd AIAA Aerospace Sciences Meeting*, (January), 1–21. doi:10.2514/6.2015-0568.

[39] Leffell, J. I., Sitaraman, J., Lakshminarayan, V. K., et al. (2016). Towards Efficient Parallel-in-Time Simulation of Periodic Flows. In *54th AIAA Aerospace Sciences Meeting*, AIAA 2016-0066. doi:10.2514/6.2016-0066.

[40] Jolivet, P. and Tournier, P.-H. (2016). Block iterative methods and recycling for improved scalability of linear solvers. In *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, pp. 190–203.

[41] Jadoui, M., Blondeau, C., Martin, E., et al. (2022). Comparative study of inner–outer krylov solvers for linear systems in structured and high-order unstructured cfd problems. *Computers & Fluids*, 244, 105575.

6 ACKNOWLEDGMENTS

This study has been supported by the European project NEXTAIR which has received funding from the European Union's Horizon Europe research and innovation program under the grant agreement No 101056732. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

The authors would also like to thank their former colleague J. Moulin for his work on the bloc-circulant preconditioner development and assessment.

COPYRIGHT STATEMENT