

# Time Spectral Method for Periodic and Quasi-Periodic Unsteady Computations on Unstructured Meshes

Dimitri J. Mavriplis<sup>\*</sup> Zhi Yang<sup>†</sup>

*Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071*

For flows with strong periodic content, time-spectral methods can be used to obtain time-accurate solutions at substantially reduced cost compared to traditional time-implicit methods which operate directly in the time domain. However, these methods are only applicable in the presence of fully periodic flows, which represents a severe restriction for many aerospace engineering problems. This paper presents an extension of the time-spectral approach for problems that include a slow transient in addition to strong periodic behavior, suitable for applications such as transient turbofan simulation or maneuvering rotorcraft calculations. The formulation is based on a collocation method which makes use of a combination of spectral and polynomial basis functions and results in the requirement of solving coupled time instances within a period, similar to the time spectral approach, although multiple successive periods must be solved to capture the transient behavior.

The implementation allows for two levels of parallelism, one in the spatial dimension, and another in the time-spectral dimension, and is implemented in a modular fashion which minimizes the modifications required to an existing steady-state solver. For dynamically deforming mesh cases, a formulation which preserves discrete conservation as determined by the Geometric Conservation Law is derived and implemented. A fully implicit approach which takes into account the coupling between the various time instances is implemented and shown to preserve the baseline steady-state multigrid convergence rate as the number of time instances is increased. Results demonstrate the accuracy, efficiency, and parallel scalability of the proposed approach for three-dimensional calculations involving a combination of strong periodic behavior and slower transient behavior.

## I. Introduction

The use of frequency domain methods for solving periodic flow problems is now a relatively well established technique in computational aerodynamics and aeroelasticity. The basic idea is to reformulate a time-dependent periodic problem into a steady-state frequency domain problem through the use of Fourier transforms. Considerable success has been demonstrated with the Harmonic balance method<sup>6, 7, 14, 15</sup> for important applications such as turbomachinery flows. More recently, a time-spectral approach has been formulated which operates directly on solution variables in the time domain.<sup>4, 18</sup> These methods have shown the potential for large increases in computational efficiency for periodic problems since they replace a time-dependent problem with a steady-state problem which can be solved efficiently with a variety of non-linear solution techniques. Additionally, the requirement of time-stepping through initial transient behavior to establish the final periodic solution is removed in these approaches, since they target directly the final periodic behavior of the flow problem.

The time spectral approach is essentially a pseudo-spectral or spectral collocation method (in time) where the time-dependent flow variables are represented in the space spanned by a set of harmonic functions, and the collocation points represent discrete instances in time at which the unknowns are to be solved. By operating directly in the time-domain, the time-spectral approach obviates the use of Fourier transforms and can be implemented into existing CFD solvers with relative ease.

However, a principal disadvantage of all these methods is that they are strictly only valid for purely periodic flows. A large class of difficult simulation problems in aerospace engineering involves quasi-periodic

---

<sup>\*</sup>Professor, email: mavriplis@uwyo.edu

<sup>†</sup>Research Scientist, email: zyang@uwyo.edu

flows, i.e. flows with strong periodic content, but with the presence of a slow transient behavior which makes these problems not truly periodic and thus not amenable to frequency domain methods. These may include turbomachinery transient behavior, maneuvering rotorcraft problems, or the simulation of a wind turbine with varying inlet conditions. For example, in the rotorcraft example, the flow is strongly periodic due to the rotor motion, although subject to a slow transient produced by the maneuvering aircraft. Such problems are notoriously difficult to simulate because the time step is restricted due to accuracy considerations by the fast periodic motion, although a long time history must be simulated to model the transient non-periodic behavior.

In this paper, we present a novel approach for simulating quasi-periodic flows through the formulation of a hybrid backwards-difference time-spectral (BDFTS) discretization of the time derivative of the governing equations. The idea is rooted in the concept of polynomial subtraction for spectral methods, discussed by Gottlieb and Orzag<sup>5</sup> and originally credited to Lanczos.<sup>9</sup> In this approach, the non-periodic (transient) portion of a quasi-periodic function is subtracted from the function and represented with a polynomial basis set. The remaining function is periodic and thus can be approximated efficiently with spectral basis functions. Our implementation corresponds to a collocation method, where the time variation of the flow variables is represented by a linear combination of spectral/harmonic basis functions and polynomial basis functions. In practice, the spectral basis functions are chosen such that the method reverts to the time-spectral approach described in reference<sup>4,18</sup> in the presence of purely periodic flows, while the polynomial basis is chosen to reproduce a first- or second-order backwards difference scheme in the limit of vanishing periodic content. This approach offers the potential for large improvements in computational efficiency for quasi-periodic problems by resolving the periodic flow effects with spectral accuracy, while enabling a BDF-type time step suited to the time scale of the slower transient effects. Furthermore, the simulation of start-up non-periodic transient effects still can be avoided by specifying a vanishing initial slow transient, in which case the first period of simulation corresponds to a traditional time-spectral method.

In order to be competitive with traditional implicit time-domain approaches, the proposed method must be able to deal with dynamically deforming meshes in a consistent manner, and the computational cost associated with the solution of the large implicit systems generated by the time discretizations must be kept manageable.

To this end, the proposed BDFTS approach is formulated for the general case of dynamically deforming meshes, and the geometric conservation law which guarantees discrete conservation for this temporal discretization is derived and implemented. A commonly employed strategy for solving the coupled time-spectral equations consists of adding an explicit pseudo-time term and time-stepping these equations in pseudo-time to the steady-state. However, it is well known that this approach suffers stability restrictions and results in convergence degradation as the number of harmonics increases.<sup>7</sup> In this work, we employ a fully (time and space) implicit solution strategy originally described by Sicot<sup>16</sup> which in principle is capable of delivering convergence rates which are independent of the number of harmonics or time instances used in the time-spectral discretization. This approach is further accelerated through the application of a line-implicit agglomeration multigrid algorithm applied to the spatially implicit system. Efficient convergence of the implicit system is even more critical for the BDFTS method, since the solution of a large number of implicit problems may be required in the context of a long time-history simulation, as opposed to the original time-spectral approach, where only a single (pseudo) steady-state problem must be solved.

In the following section, we first outline the baseline time-spectral approach, and then describe the full BDFTS method. The geometric conservation law for these methods is then described and formulated in subsection II D. Next, the implicit solution strategy is described in subsection II E. The parallel implementation of the method is also described taking into consideration optimization on multi-core architectures and through a strategy that requires minimal modifications to existing CFD solvers, as described in subsection II F. Validation of the method and representative results are given in Section III and conclusions are given in Section IV.

## II. Formulation

### A. Base Solver

The Navier-Stokes equations in conservative form can be written as:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F}(\mathbf{U}) + \mathbf{G}(\mathbf{U})) = 0 \quad (1)$$

where  $\mathbf{U}$  represents the vector of conserved quantities (mass, momentum, and energy),  $\mathbf{F}(\mathbf{U})$  represents the convective fluxes and  $\mathbf{G}(\mathbf{U})$  represents the viscous fluxes. Integrating over a (moving) control volume  $\Omega(t)$ , we obtain:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) \cdot \bar{\mathbf{n}}) dS + \int_{\partial\Omega(t)} (\mathbf{G}(\mathbf{U}) \cdot \bar{\mathbf{n}}) dS = 0 \quad (2)$$

Using the differential identity

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV = \int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} \mathbf{U} (\dot{\mathbf{x}} \cdot \bar{\mathbf{n}}) dS \quad (3)$$

where  $\dot{\mathbf{x}}$  and  $\bar{\mathbf{n}}$  are the velocity and normal of the interface  $\partial\Omega(t)$ , respectively, equation (2) becomes:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U} \dot{\mathbf{x}}) \cdot \bar{\mathbf{n}} dS + \int_{\partial\Omega(t)} \mathbf{G}(\mathbf{U}) \cdot \bar{\mathbf{n}} dS = 0 \quad (4)$$

Considering  $\mathbf{U}$  as cell averaged quantities, these equations are discretized in space as:

$$\frac{\partial}{\partial t} (V\mathbf{U}) + \mathbf{R}(\mathbf{U}, \bar{\mathbf{x}}(\mathbf{t}), \bar{\mathbf{n}}(\mathbf{t})) + \mathbf{S}(\mathbf{U}, \bar{\mathbf{n}}(\mathbf{t})) = \mathbf{0} \quad (5)$$

where  $\mathbf{R}(\mathbf{U}, \bar{\mathbf{x}}, \bar{\mathbf{n}}) = \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U} \dot{\mathbf{x}}) \cdot \bar{\mathbf{n}} dS$  represents the discrete convective fluxes in ALE form,  $\mathbf{S}(\mathbf{U}, \bar{\mathbf{n}})$  represents the discrete viscous fluxes, and  $V$  denotes the control volume. In the discrete form,  $[\bar{\mathbf{x}}(\mathbf{t}) = \dot{\mathbf{x}}(\mathbf{t}) \cdot \bar{\mathbf{n}}(\mathbf{t})]$  and  $\bar{\mathbf{n}}(\mathbf{t})$  now represent the time varying face-integrated velocities and surface normals of the control-volume boundary faces. Note that only the product  $\dot{\mathbf{x}}(\mathbf{t}) \cdot \bar{\mathbf{n}}(\mathbf{t})$  appears in the residual, thus the functional dependence on  $\bar{\mathbf{x}}(\mathbf{t})$  in the place of  $\dot{\mathbf{x}}(\mathbf{t})$ .

The Navier-Stokes equations are discretized by a central difference finite-volume node-based scheme with additional matrix-based artificial dissipation on hybrid meshes which may include triangular and quadrilateral elements in two dimensions, or tetrahedra, pyramids, prisms and hexahedra in three dimensions. Second-order accuracy is achieved using a two-pass construction of the artificial dissipation operator, which corresponds to an undivided biharmonic operator. A single unifying edge-based data-structure is used in the flow solver for all types of elements. For multigrid calculations, a first-order accurate discretization is employed for the convective terms on coarse grid levels.<sup>11,12</sup>

## B. Time Spectral Method

If the flow is periodic in time, the variables  $\mathbf{U}$  can be represented by a discrete Fourier series. The discrete Fourier transform of  $\mathbf{U}$  in a period of  $T$  is given by<sup>4</sup>

$$\hat{\mathbf{U}}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{U}^n e^{-ik \frac{2\pi}{T} n \Delta t} \quad (6)$$

where  $N$  is the number of time intervals and  $\Delta t = T/N$ . The Fourier inverse transform is then given as

$$\mathbf{U}^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{\mathbf{U}}_k e^{ik \frac{2\pi}{T} n \Delta t} \quad (7)$$

Note that this corresponds to a collocation approximation, i.e. the function  $\mathbf{U}(t)$  is projected into the space spanned by the truncated set of complex exponential (spectral) functions, and the expansion coefficients (in this case the  $\hat{\mathbf{U}}_k$ ) are determined by requiring  $\mathbf{U}(t)$  to be equal to its projection at  $N$  discrete locations in time, as given by equations (6) and (7).

Differentiating equation (7) in time, we obtain:

$$\frac{\partial}{\partial t} (\mathbf{U}^n) = \frac{2\pi}{T} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} ik \hat{\mathbf{U}}_k e^{ik \frac{2\pi}{T} n \Delta t} \quad (8)$$

Substituting equation (6) into equation (8), we get<sup>1,8</sup>

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=0}^{N-1} d_n^j \mathbf{U}^j \quad (9)$$

where

$$d_n^j = \begin{cases} \frac{2\pi}{T} \frac{1}{2} (-1)^{n-j} \cot\left(\frac{\pi(n-j)}{N}\right) & n \neq j \\ 0 & n = j \end{cases}$$

for an even number of time instances and

$$d_n^j = \begin{cases} \frac{2\pi}{T} \frac{1}{2} (-1)^{n-j} \operatorname{cosec}\left(\frac{\pi(n-j)}{N}\right) & n \neq j \\ 0 & n = j \end{cases}$$

for an odd number of time instances. The collocation approach for solving equation (5) consists of substituting the collocation approximation for the continuous function  $\mathbf{U}(t)$  given by equation (7) into equation (5), and requiring equation (5) to hold exactly at the same  $N$  discrete locations in time (i.e. multiplying (5) by the dirac delta test function  $\delta(t - t^n)$  and integrating over all time), yielding:

$$\sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j + \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) + \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) = 0 \quad n = 0, 1, 2, \dots, N-1 \quad (10)$$

This results in a system of  $N$  equations for the  $N$  time instances  $\mathbf{U}^n$  which are all coupled through the summation over the time instances in the time derivative term. The spatial discretization operators remain unchanged in the time-spectral approach, with only the requirement that they be evaluated at the appropriate location in time. Thus, the time-spectral method may be implemented without any modifications to an existing spatial discretization, requiring only the addition of the temporal discretization coupling term, although the multiple time instances must be solved simultaneously due to this coupling.

### C. Hybrid BDF/Time Spectral Method

The idea of polynomial subtraction for quasi-periodic functions is to subtract out the non-periodic transient, which can be modeled using a polynomial basis set, and to approximate the remaining purely periodic component with a spectral basis set.<sup>5</sup> From the point of view of a collocation method, this corresponds to using a mixed spectral/polynomial basis set for the projection of the continuous solution (in the time dimension).

We proceed by splitting the quasi-periodic temporal variation of the solution into a periodic and slowly varying mean flow as:

$$\mathbf{U}(t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{\mathbf{U}}_k e^{ik \frac{2\pi}{T} n \Delta t} + \bar{\mathbf{U}}(t) \quad (11)$$

where the slowly varying mean flow is approximated by a collocation method using a polynomial basis set as:

$$\bar{\mathbf{U}}(t) = \phi_{12}(t) \mathbf{U}^{m+1} + \phi_{11}(t) \mathbf{U}^m \quad (12)$$

for a linear variation and

$$\bar{\mathbf{U}}(t) = \phi_{23}(t) \mathbf{U}^{m+1} + \phi_{22}(t) \mathbf{U}^m + \phi_{21}(t) \mathbf{U}^{m-1} \quad (13)$$

for a quadratic variation in time. Here  $\mathbf{U}^m$  and  $\mathbf{U}^{m+1}$  represent discrete solution instances in time usually taken as the beginning and ending points of the considered period in the quasi-periodic motion (and  $\mathbf{U}^{m-1}$  corresponds to the beginning point of the previous period). In the first case,  $\phi_{12}(t)$  and  $\phi_{11}(t)$  correspond to the linear interpolation functions given by:

$$\phi_{11}(t) = \frac{t^{m+1} - t}{T} \quad (14)$$

$$\phi_{12}(t) = \frac{t - t^m}{T} \quad (15)$$

with the period given as  $T = t^{m+1} - t^m$ . Similarly, the  $\phi_{23}(t), \phi_{22}(t), \phi_{21}(t)$  are given by the corresponding quadratic interpolation functions. Note that in this case, the collocation approximation leads to the determination of the Fourier coefficients as:

$$\hat{\mathbf{U}}_k = \frac{1}{N} \sum_{n=0}^{N-1} \tilde{\mathbf{U}}^n e^{-ik \frac{2\pi}{T} n \Delta t} \quad (16)$$

with  $\tilde{\mathbf{U}}^n = \mathbf{U}^n - \bar{\mathbf{U}}^n$  defined as the remaining periodic component of the function after polynomial subtraction. Differentiating equation (11) and making use of equations (9) and (16) we obtain the following expression for the time derivative:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=0}^{N-1} d_n^j \tilde{\mathbf{U}}^j + \phi'_{12}(t_n) \mathbf{U}^{m+1} + \phi'_{11}(t_n) \mathbf{U}^m \quad (17)$$

for the case of a linear polynomial functions in time. The  $\phi'_{12}(t_n)$  and  $\phi'_{11}(t_n)$  represent the time derivatives of the polynomial basis functions (resulting in the constant values  $\frac{-1}{T}$  and  $\frac{1}{T}$  in this case), and the various time instances are given by:

$$t_j = t_m + \frac{j}{N}(t_{m+1} - t_m), \quad j = 0, \dots, N-1$$

We also note that  $\bar{\mathbf{U}}(t_m) = \mathbf{U}^m = \mathbf{U}(t_m)$  and thus we have  $\tilde{\mathbf{U}}^0 = 0$ . In other words, the constant mode in the spectral representation must be taken as zero, since it is contained in the polynomial component of the function representation. Therefore, the  $j = 0$  component in the summation can be dropped, and rewriting equation (17) in terms of the original time instances  $\mathbf{U}^n$  we obtain:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=1}^{N-1} d_n^j \mathbf{U}^j - \left( \sum_{j=1}^{N-1} d_n^j \phi_{12}(t_j) - \phi'_{12}(t_n) \right) \mathbf{U}^{m+1} - \left( \sum_{j=1}^{N-1} d_n^j \phi_{11}(t_j) - \phi'_{11}(t_n) \right) \mathbf{U}^m \quad (18)$$

Finally, the above expression for the time derivative is substituted into equation (5) which is then required to hold exactly at time instances  $j = 1, 2, \dots, N-1$  and  $j = N$  (which corresponds to the  $m+1$  time instance):

$$\begin{aligned} \sum_{j=1}^{N-1} d_n^j V^j \mathbf{U}^j - \left( \sum_{j=1}^{N-1} d_n^j \phi_{12}(t_j) - \phi'_{12}(t_n) \right) V^{m+1} \mathbf{U}^{m+1} - \left( \sum_{j=1}^{N-1} d_n^j \phi_{11}(t_j) - \phi'_{11}(t_n) \right) V^m \mathbf{U}^m \\ + \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) + \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) = 0 \quad n = 1, 2, \dots, N \end{aligned} \quad (19)$$

As previously, we have  $N$  coupled equations with  $N$  unknown time instances, although in this case the  $j = 0$  time instance which corresponds to the  $\mathbf{U}^m$  values are known from the solution of the previous period, while the  $j = N$  or  $\mathbf{U}^{m+1}$  values are not known, since these are not equal to the  $j = 0$  values as they would be in a purely periodic flow. In the case of vanishing periodic content, summation terms involving the  $d_n^j$  coefficients vanish by virtue of equation (17) with  $\tilde{\mathbf{U}}^j = 0$  and it is easily verified that the above formulation reduces to a first-order backwards difference scheme with a time step equal to the period  $T$ . On the other hand, for purely periodic motion, we have  $\mathbf{U}^{m+1} = \mathbf{U}^m$  which results in cancellation of the polynomial derivative terms  $\phi'_{12}(t_n)$  and  $\phi'_{11}(t_n)$ . Furthermore, using the identities  $\phi_{12}(t_j) + \phi_{11}(t_j) = 1$ , and  $\sum_{j=0}^{N-1} d_n^j = 0$ , it can be seen that the remaining polynomial terms reduce to the missing  $j = 0$  instance in the summation. Given the equality  $\mathbf{U}^{m+1} = \mathbf{U}^m$ , the last equation at  $j = N$  becomes identical to the  $j = 0$  equation and the time-spectral method given by equation (10) is recovered.

In this description we have used linear polynomials corresponding to a BDF1 time-stepping scheme for clarity. In practice, BDF2 time-stepping schemes are required for accuracy purposes, and the equivalent scheme based on quadratic polynomials is given as:

$$\begin{aligned} \sum_{j=1}^{N-1} d_n^j V^j \mathbf{U}^j - \left( \sum_{j=1}^{N-1} d_n^j \phi_{23}(t_j) - \phi'_{23}(t_n) \right) V^{m+1} \mathbf{U}^{m+1} \\ - \left( \sum_{j=1}^{N-1} d_n^j \phi_{22}(t_j) - \phi'_{22}(t_n) \right) V^m \mathbf{U}^m - \left( \sum_{j=1}^{N-1} d_n^j \phi_{21}(t_j) - \phi'_{21}(t_n) \right) V^{m-1} \mathbf{U}^{m-1} \\ + \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) + \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) = 0 \quad n = 1, 2, \dots, N \end{aligned} \quad (20)$$

where the values  $\mathbf{U}^{m-1}$  and  $\mathbf{U}^m$ , which correspond to the time instances at the beginning and end of the previous period are known from the solution of earlier periods, and  $\mathbf{U}^{m+1} = \mathbf{U}^N$  as previously.

#### D. Geometric Conservation Law

For dynamically deforming meshes, the individual cell volumes or control volumes  $V(t)$  are functions of time, and strictly speaking, it is the product  $V(t)\mathbf{U}(t)$  which must be represented by the Fourier series

$$V^n \mathbf{U}^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \widehat{V\mathbf{U}}_k e^{ik \frac{2\pi}{T} n \Delta t} \quad (21)$$

in order to obtain the first term as written in equations (10), (19) or (20). However, since the Fourier coefficients  $\widehat{V\mathbf{U}}_k$  are never required, and the values of the volumes at the specific time instances are known or computable, this does not lead to any complications in the solution of equations (10) and (20). On the other hand, the geometric conservation law states that in order to maintain discrete conservation, equations (10) and/or (20) must preserve uniform flow as an exact (discrete) solution.<sup>3,13</sup> Substituting  $\mathbf{U}^n = \text{constant}$  into equation (10) the  $\mathbf{R}$  operator reduces to the ALE term (i.e. second term in second integral in equation (4)), and the  $\mathbf{S}$  operator vanishes due to its dependence on gradients of  $\mathbf{U}$ , yielding the following constraint for the face-integrated grid speeds in the case of the time-spectral method:

$$\sum_{CVFaces} \bar{\mathbf{x}}^n = \sum_{j=0}^{N-1} d_n^j V^j \quad (22)$$

where the sum on the left hand side is over the faces which delimit the cell or control volume  $V^j$ . The right hand side of this equation represents a known quantity, since the cell volumes at each time instance are computable given the known grid point locations. However, equation (21) does not in itself enable the prescription of the face integrated grid speeds, since there are more grid faces than volumes. Following references<sup>13,19</sup> the right hand side can be further decomposed into the time derivative of the volume swept by each constituent face, providing a set of equations determining the face integrated grid speed terms as:

$$\bar{\mathbf{x}}^n = \dot{\mathbf{x}}^n \cdot \mathbf{n}^n = \sum_{j=0}^{N-1} d_n^j F^j \quad (23)$$

where  $F^j$  now represents the volume swept by the face corresponding to  $\bar{\mathbf{x}}$ . For example, in the case of three-dimensional tetrahedral faces, each term in the right hand side summation represents the volume swept by the face  $F^j$  between the current  $n^{th}$  and the  $j^{th}$  time instance. These swept volumes are computed by discretizing the trajectories of the faces between the two time levels into a large number of discrete steps (usually 360 per period) and computing the volume swept between each step using a two-point integration rule in time, as discussed in references.<sup>3,13</sup>

For the BDFTS method, the corresponding GCL condition becomes:

$$\begin{aligned} \sum_{CVFaces} \bar{\mathbf{x}}^n = & \sum_{j=1}^{N-1} d_n^j V^j - \left( \sum_{j=1}^{N-1} d_n^j \phi_{23}(t_j) - \phi'_{23}(t_n) \right) V^{m+1} \\ & - \left( \sum_{j=1}^{N-1} d_n^j \phi_{22}(t_j) - \phi'_{22}(t_n) \right) V^m - \left( \sum_{j=1}^{N-1} d_n^j \phi_{21}(t_j) - \phi'_{21}(t_n) \right) V^{m-1} \end{aligned} \quad (24)$$

and the volumes are decomposed into constituent faces, and the volume swept by these faces are computed in an analogous manner to that described above for the time-spectral method, in order to determine the GCL compliant face integrated grid speeds.

## E. Fully Implicit Method

A common approach for solving the system of equations resulting from the time spectral method (c.f. equation (10)) consists of adding a pseudo-time term as:

$$\frac{\partial}{\partial \tau}(V^n \mathbf{U}^n) + \sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j + \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) + \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) = 0 \quad (25)$$

and time-stepping these equations until a pseudo-time steady state is achieved. However, for explicit pseudo-time stepping approaches, it has been shown that the pseudo-time step is limited by stability considerations as:<sup>18</sup>

$$\Delta \tau_n = CFL \frac{V^n}{\|\lambda\| + \frac{N}{2} V^n} \quad (26)$$

where  $\lambda$  is the spectral radius of the spatial discretization operator  $\mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) + \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n)$  and  $N$  represents the number of time instances that must be solved together. This restriction results in convergence degradation as the number of time instances or harmonics is increased. This restriction can be removed by resorting to an implicit approach in pseudo-time. Such an approach has been derived in reference<sup>16</sup> using a first-order backwards difference scheme in pseudo time.

An alternate strategy consists of devising a Newton approach for solving the fully coupled non-linear equations at all time instances given by equation (10) or (20). The Newton scheme takes the form:

$$\mathbf{A} \Delta \mathbf{U} = - \sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j - \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) - \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) \quad (27)$$

with the resulting Jacobian matrix given by:<sup>16</sup>

$$\mathbf{A} = \begin{bmatrix} \frac{V^0}{\Delta \tau_0} \mathbf{I} + \mathbf{J}_0 & V^1 d_0^1 \mathbf{I} & \dots & V^{N-1} d_0^{N-1} \mathbf{I} \\ V^0 d_1^0 \mathbf{I} & \frac{V^1}{\Delta \tau_1} \mathbf{I} + \mathbf{J}_1 & \dots & V^{N-1} d_1^{N-1} \mathbf{I} \\ \vdots & \vdots & \dots & \vdots \\ V^0 d_{N-1}^0 \mathbf{I} & V^1 d_{N-1}^1 \mathbf{I} & \dots & \frac{V^{N-1}}{\Delta \tau_{N-1}} \mathbf{I} + \mathbf{J}_{N-1} \end{bmatrix} \quad (28)$$

where a diagonal pseudo-time term can be included as shown for enhanced diagonal dominance of the Jacobian matrix. In the above matrix,  $\mathbf{J}_j$  correspond to the Jacobian of the spatial discretization operator evaluated at time instance  $j$ . Keeping only the diagonal blocks in this matrix and discarding all other terms corresponds to a spatially implicit method with explicit treatment of the different time instances. On the other hand, if all terms are retained, and a block Jacobi strategy is used to solve equation (27), a fully implicit solution strategy which takes into account the coupling between all time instances is obtained. An iteration of the block Jacobi strategy can be written as:

$$\left[ \frac{V^n}{\Delta \tau_n} \mathbf{I} + \mathbf{J}_n \right] \Delta \mathbf{U}_n^{l+1} = - \sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j - \mathbf{R}(\mathbf{U}^n, \bar{\mathbf{x}}^n, \bar{\mathbf{n}}^n) - \mathbf{S}(\mathbf{U}^n, \bar{\mathbf{n}}^n) - \sum_{j \neq n} [V^j d_n^j \mathbf{I}] \Delta \mathbf{U}_j^l \quad (29)$$

with  $n = 0, 1, 2, \dots, N-1$ , and where  $l$  represents the linear block Jacobi cycle index. Each cycle of this solution strategy requires the inversion of the matrix block on the left-hand side of equation (29), which corresponds to a spatially implicit solution. This is achieved using a line-implicit agglomeration linear multigrid solver previously developed for steady-state and time implicit problems in the baseline solver.<sup>10, 12</sup> Note that the off-diagonal blocks of the matrix  $\mathbf{A}$  which also appear on the right-hand side of equation (29) are rather simple, as they involve a scalar multiplication of the identity matrix. However, the size of each block corresponds to a complete spatial field for one time instance. The fully implicit method can be implemented with little additional memory and requires only a small amount of additional floating point operations. However, for parallel implementations, each time instance solution must broadcast its entire solution field to all other solution instances. In order to reduce overall communication, this coupling is performed at the end of every multigrid cycle, rather than after each iterative smoothing cycle. Although the implicit method has been described in the context of the TS method, it extends readily to the BDFTS method by considering the equivalent Newton scheme and associated linearization applied to equation (20).

## F. Parallel Implementation

A principal advantage of the TS and BDFTS approaches is that they can be implemented with relative ease into existing steady-state codes because the spatial discretization operator is unchanged from the baseline code. However, both approaches result in multiple time instances which are coupled and must be solved simultaneously. On the one hand, this provides the opportunity for exploiting parallelism in the time direction, as compared to traditional time-stepping schemes which necessarily advance sequentially in time. This feature may prove to be particularly enabling with the advent of rapidly expanding hardware parallelism, particularly for cases where parallelism in the spatial dimension has been exhausted (perhaps due to the adequacy of moderate grid sizes). A non-intrusive approach for implementing TS/BDFTS methods in parallel can be achieved by introducing separate MPI communicators. The baseline solver operates in parallel and makes use of an MPI communicator to exchange information between neighboring spatial grid partitions. Our strategy consists of replicating instances of the entire solver on additional processors for each required time instance in the TS/BDFTS formulation. In this manner, the code remains unchanged apart from the addition of a source term which provides the coupling between time instances due to the TS time derivative term. This fully parallel implementation contains two types of inter-processor communication: communication between spatial partitions within a single time instance, and communication between all of the time instances. A simple approach is to use a separate additional MPI communicator for the latter type of communication, leaving all original spatial communication routines unchanged. One of the drawbacks of the TS/BDFTS methods is that each time instance must broadcast its entire solution field to all other time instances, which can result in a significant amount of communication. We mitigate this to some degree, as mentioned previously in the implicit solution strategy, by only updating the coupling terms from other time instances after each multigrid cycle. On machines with high-bandwidth interconnects we achieve good scalability using this approach. However, on inexpensive clusters with Gigabit ethernet, significant slowdown due to communication volume has been observed. One approach to relieve this bottleneck is through careful placement of the time instance partitions. For multicore and/or multiprocessor hardware nodes within a distributed memory parallel machine, the optimal strategy consists of placing all time instances of a particular spatial partition on the same node, with each time instance being assigned to a local core, while the individual spatial partitions are distributed across the nodes of the machine. In this manner, all time-instance communication generated by the TS/BDFTS methods (which is spatially local) becomes node local and benefits from the shared memory and/or faster local communication bandwidth within a node. The current popularity of 4,8 or 16 core nodes makes this approach well suited for the typical number of time instances used in the TS/BDFTS methods.

## III. Results and Discussion

### A. Validation of the Time Spectral Method

A two-dimensional inviscid flow test case is constructed using a forced oscillation NACA0012 airfoil, at a Mach number of 0.755, and a mean incidence  $\alpha_0$  of 0.016 degrees. A periodic pitching motion is prescribed at the quarter chord point of the airfoil:

$$\alpha(t) = \alpha_0 + \alpha_A \sin(\omega t) \quad (30)$$

where the reduced frequency  $\omega$  is equal to 0.1628 and the amplitude  $\alpha_A$  is equal to 2.51 degrees. The unstructured mesh consists of 4379 nodes and 8543 triangles. Figure (1a) shows the near field mesh. For each time interval  $n$ , a three level multigrid sequence is used in the solver. Figure (1b) shows the comparison of the convergence histories between the fully coupled implicit method and the spatially implicit method where the off-diagonal blocks in the  $\mathbf{A}$  matrix are discarded. For the local implicit method, the CFL number decreases as the harmonics increase, resulting in a degradation of the convergence rate. For the fully coupled implicit method, the CFL number and the convergence rate remain constant as the number of harmonics is increased. Figure (2a) shows the comparison of the lift coefficient versus the angle of attack between the time accurate method (using the second-order accurate BDF2 time discretization) with  $\Delta t = T/64$  where  $T$  denotes the period, and the time spectral method with  $N = 4, 8, 12$  time instances. For this case, the time spectral method with even 1 harmonic or 4 time intervals shows equivalent accuracy compared to the time accurate method. Figure (2b) shows the comparison of the moment coefficient versus the angle of attack between the time accurate method (BDF2) with  $\Delta t = T/64$  and the time spectral method with  $N = 4, 8, 12$ .

The moment history contains multiple harmonics and thus is not captured accurately with  $N = 4$  in the TS method. However, the results show convergence of the TS method to the time accurate BDF2 results method as the number of harmonics or time intervals increases.

In the next test case, the time spectral method is compared with the BDF2 time accurate method for a more complicated case involving a wider range of frequencies. A periodic pitching motion is prescribed at the quarter chord point of the airfoil as:

$$\alpha(t) = \frac{\alpha_A}{5 - 4\cos(\omega t)} \quad (31)$$

where the reduced frequency  $\omega$  equals 0.1628 and the amplitude  $\alpha_A$  equals 2.51 degrees. This motion is no longer purely harmonic and should prove to be more difficult to model accurately with small numbers of time instances. The mesh employed for this case is identical to that used in the previous case. Figure (3a) shows the comparison of the lift coefficient versus the angle of attack between the time accurate method (BDF2) with  $\Delta t = T/128$  and the time spectral method with  $N = 4, 8, 12, 16$ . Because the motion of this case consists of a wide range of frequencies, the result using a single harmonic or four time intervals shows relatively large error. As the time intervals  $N$  increases, the results of the time spectral method show convergence to the results of the BDF2 time accurate method. Figure (3b) shows the comparison of the moment coefficient versus the angle of attack between the time accurate method (BDF2) with  $\Delta t = T/128$  and the time spectral method with  $N = 4, 8, 12, 16$  where good agreement is seen using  $N = 16$ .

The last time-spectral validation case corresponds to a three-dimensional inviscid flow test case which is constructed by forced oscillation of an AGARD 445.6 wing at a Mach number of 0.701. The periodic pitching motion is prescribed at the quarter chord point of the airfoil by equation (30) with the reduced frequency  $\omega$  equal to 0.1745 and amplitude  $\alpha_A$  equal to 5 degrees. Figure (4a) shows the near field mesh for the AGARD 445.6 wing. This mesh consists of 40460 nodes and 224531 tetrahedra. Figure (4b) shows the comparison of the lift coefficient versus the angle of attack between the time accurate method (BDF2) with  $\Delta t = T/72$  and the time spectral method with  $N = 4, 6, 8$ . In this case, only one harmonic (i.e.  $N = 4$ ) is sufficient to obtain good accuracy compared to the time accurate method.

## B. Validation of the BDF/Time Spectral Method

First, a two-dimensional inviscid pitching-climbing airfoil case was constructed by using the same NACA0012 mesh as the previous section. The Mach number is set to 0.555. A periodic pitching motion is prescribed at the quarter chord point of the airfoil. Simultaneously, over a sequence of ten periods, the mean angle of attack of the airfoil changes, while the airfoil translation accelerates due to prescribed vertical and horizontal velocity transients. The motion of the airfoil is illustrated in Figure (5a), while the periodic pitching motion, mean angle of attack, and horizontal and vertical velocities of the airfoil are plotted in Figure (5b).

The angle of attack is prescribed as:

$$\alpha(t) = \alpha_0 + \bar{\alpha}(t) + \alpha_1 \sin(\omega_1 t) + \alpha_2 \sin(\omega_2 t) \quad (32)$$

where the mean angle of attack is given as:

$$\bar{\alpha}(t) = \begin{cases} 0 & t < t_1 \\ \alpha_m \frac{1}{2} (1 - \cos(\omega_m (t - t_1))) & t \geq t_1 \end{cases}$$

where  $t_1$  is the prescribed time at which the transient motion begins (taken as 1 period) and with the constants taken as:

$$\begin{aligned} \alpha_0 &= 0.016^\circ, & \alpha_m &= 2^\circ, & \alpha_1 &= 2.51^\circ, & \alpha_2 &= 1.25^\circ \\ \omega_1 &= 0.1632, & \omega_2 &= 3\omega_1, & \omega_m &= 0.1\omega_1 \end{aligned}$$

The angle of attack and the forward/upward velocity are shown in Figure (5b), which clearly illustrates that the variations in the mean angle of attack and the airfoil translational motion represent slow transients compared to the periodic pitching motion, which itself contains multiple harmonics. Figure (6a) shows the comparison of the computed lift coefficient in the first period (based on the frequency  $\omega_1$ ). Because the highest mode of the pitching motion  $\omega_2$  is three times that of the base mode  $\omega_1$ , 7 time instances are need to produce an accurate solution. From the figure, the BDFTS scheme with five time instances shows poor

agreement with the reference BDF2 results obtained using 512 time steps per period. On the other hand, using 7 or more time instances with the BDFTS scheme produces very good agreement with the reference BDF2 solution over the complete time history which includes 11 periods, as shown in Figure (6b). The response of the drag coefficient shows twice as many modes as that of the lift coefficient, and at least 13 time instances are need to produce an accurate drag coefficient with the BDFTS scheme. Figure (7a) shows the comparison of the computed drag coefficients over the first period. The results of the BDFTS scheme with fewer than 13 time instances show poor agreement with the reference BDF2 solution (using 512 time steps per period), while the BDFTS results using 13 time instances show very good agreement with the BDF2 solution, not only over the first period, but over the entire 11 periods of slow transient motion, as shown in Figure (7b). Figure (8) shows the convergence history of the BDFTS scheme for this case over a selected period of motion. Using the fully implicit block-Jacobi scheme driven by a spatial agglomeration multigrid algorithm and temporal-instance communication updates after each multigrid cycle, robust convergence rates which are independent of the number of time instances are obtained.

The next case is a three-dimensional inviscid pitching-climbing wing which uses the same mesh as the AGARD 445.6 wing in the previous section. The Mach number is set to 0.511. The wing undergoes a forced to a pitching motion, while at the same time undergoing a slow change in mean angle of attack and a transient rising motion similar to that prescribed in the previous case. The time dependent angle of attack and the prescribed airfoil forward and upward velocities are shown in Figure (9a). Figure (9b) shows the comparison of the computed lift coefficient. The BDFTS scheme with  $N = 3, 5, 7$  time instance shows good agreement with the reference BDF2 solution computed using 64 time steps per period. The computed drag coefficients using the BDFTS scheme are seen to compare well with the BDF2 reference solution as well in Figure (10a) over the entire time history. Figure (10b) shows additional detail for the comparison of the computed drag coefficients, where it can be seen that the BDFTS results converge uniformly to the BDF2 reference solution as the number of time instances is increased. Figure (11) shows the convergence history of the BDFTS scheme for this case over a typical time period, illustrating convergence rates which are independent of the number of time instances in the BDFTS scheme.

The next case consists of a more realistic three-dimensional problem based on the three-bladed rotor configuration from the TRAM experiment, which is a quarter scale V-22 model. This case consists of a Reynolds-averaged Navier-Stokes simulation using the Spalart-Allmaras turbulence model<sup>17</sup> and operating on a hybrid unstructured mesh of mixed prismatic and tetrahedral elements. The mesh consists of 1.8 million grid points and rotates with the body, although no mesh deformation is required, since the configuration is assumed to be rigid. The rotational speed of the rotor is constant, while the rotor shaft undergoes a pitch and translation maneuver representative of the transition from hover to forward flight, as shown in Figure (12a). The shaft angle and the forward/upward velocity are shown in Figure (12b). This case is similar to the case described in reference.<sup>2</sup> Figures (13a and b) show the comparison of the computed forces in the x and z coordinate directions between the BDF2 approach using a 1 degree of rotation time step and the BDFTS method using 7 and 9 time instances per rotational period. Reasonable agreement between the BDF2 and BDFTS calculations is observed, considering the BDFTS approach comprises only 7 or 9 time instances for every 360 time steps (i.e. a factor of  $40 \sim 50$  less) in the BDF2 simulation. This case also represents a relatively difficult test for the BDFTS approach, since the time scale of the transient motion is relatively fast compared to the period of revolution (i.e. only 5 rotor revolutions are executed during the transient motion). The overall force variations are well predicted by the BDFTS method, although agreement near the end of the maneuver for the x-direction force degrades and some of the finer details of the force histories are not matched. However, the results using 9 time instances show better agreement with the BDF2 simulation than the result using 7 time instances, and simulations with larger number of harmonics or time instances  $N$  per period can be expected to deliver even closer agreement with the reference solution.

A comparison of total wall clock time is made between the BDFTS and BDF2 simulations in order to demonstrate the efficiency of the BDFTS approach. Approximately 24.1 seconds of wall clock time are required to complete one multigrid cycle for BDF2 using 8 processors or cores (Intel Xeon CPU). Because the BDFTS solver operates in parallel both in space and time the BDFTS simulation with 7 time instances is run on 56 cores and takes 31.2 wall clock seconds per multigrid cycle, while the BDFTS simulation with 9 time instances takes 32.9 seconds per multigrid cycle using 72 cores. For BDF2 using a time step of 1 degree (360 time steps per revolution) and pseudo time integration of 50 multigrid cycles per time step, the total computational wall clock time is  $24.1 \times 50 \times 360 \times 6 = 2604028$  seconds For BDFTS using 7(9) time instances and 300 multigrid cycles per pseudo time integration, the total computational wall clock time is

$31.2(32.9) \times 300 \times 6 = 56101(59282)$  seconds. Thus, the speed up factor between BDFTS and BDF2 for these cases varies between 43.9 and 46.4 using the same spatial partitioning (but different overall numbers of cores). In order to compare the speedup on equivalent number of cores, the number of spatial partitions for the BDF2 case must be increased to match the number of cores used in the BDFTS cases. Assuming 100% parallel efficiency, BDF2 would require  $2604028/7 = 372004$  seconds using 56 cores and  $2604028/9 = 289336$  seconds using 72 cores. Thus, using the same number of cores, the speed up factor between BDFTS with 7(or 9) time instances and BDF2 is about 6.63(or 4.88). In practice, when large numbers of cores are available, the BDFTS approach can be expected to provide more opportunity for effective parallelism compared to BDF2 or other standard time-implicit approaches.

## IV. Concluding Remarks

The BDFTS method has been shown to enable the extension of time-spectral methods to quasi-periodic flows with slow transient behavior. While the BDFTS method is capable of delivering high accuracy with relatively few time instances per period for such flows, the precise order of accuracy of this temporal discretization remains to be investigated. For the TS approach, one would expect spectral accuracy in the presence of smooth time histories. For the BDFTS approach, the design accuracy may be expected to be highly dependent on the solution behavior. The method appears to be best suited for cases with two disparate time scales, a fast periodic motion and a slow transient motion. Furthermore, a controlled study of computational efficiency of the BDFTS approach versus a traditional BDF2 scheme at equivalent levels of accuracy is warranted. Extension of the approach to include higher-order polynomial functions and comparisons with other approaches such as fully implicit Runge-Kutta, spectral-element in time, and discontinuous Galerkin in time are planned in future work. Additionally, application of this methodology to practical maneuvering rotorcraft problems will also be considered.

## V. Acknowledgments

This work was partly funded under ONR Grant N00014-09-1-1060 and NSF Award 0960490.

## References

- <sup>1</sup>C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer, 1987.
- <sup>2</sup>B. D. Eric J. Nielsen and N. K. Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. AIAA 2009-3802, June 2009.
- <sup>3</sup>P. Geuzaine, C. Grandmont, and C. Farhat. Design and analysis of ale schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *Journal of Computational Physics*, 191(1):206–227, 2003.
- <sup>4</sup>A. K. Gopinath and A. Jameson. Time spectral method for periodic unsteady computations over two- and three-dimensional bodies. AIAA Paper 2005-1220, Jan. 2005.
- <sup>5</sup>D. Gottlieb and S. A. Orszag. Numerical analysis of spectral methods: Theory and applications. CBMS-26, Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1977.
- <sup>6</sup>K. C. Hall and E. F. Crawley. Calculation of unsteady flows in turbomachinery using the linearized Euler equations. *AIAA Journal*, 27(6):777–787, 1989.
- <sup>7</sup>K. C. Hall, J. P. Thomas, and W. S. Clark. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. *AIAA Journal*, 40(5):879–886, 2002.
- <sup>8</sup>J. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge Monographs on Applied and Computational Mathematics, 2007.
- <sup>9</sup>C. Lanczos. Discourse on Fourier series. Hafner, New York, 1966.
- <sup>10</sup>D. J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *Journal of Computational Physics*, 145(1):141–165, Sep. 1998.
- <sup>11</sup>D. J. Mavriplis and S. Pirzadeh. Large-scale parallel unstructured mesh computations for 3D high-lift analysis. *AIAA Journal of Aircraft*, 36(6):987–998, Dec. 1999.
- <sup>12</sup>D. J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the navier-stokes equations on mixed element meshes. Technical Report CR-198183, NASA, July 1995.
- <sup>13</sup>D. J. Mavriplis and Z. Yang. Construction of the discrete geometric conservation law for high-order time accurate simulations on dynamic meshes. *Journal of Computational Physics*, 213(2):557–573, 2006.
- <sup>14</sup>M. McMullen, A. Jameson, and J. J. Alonso. Acceleration of convergence to a periodic steady state in turbomachinery flows. AIAA Paper 2001-0152, Jan. 2001.
- <sup>15</sup>M. McMullen, A. Jameson, and J. J. Alonso. Application of a non-linear frequency domain solver to the euler and navier-stokes equations. AIAA Paper 2002-0120, Jan. 2002.

<sup>16</sup>F. Sicot, G. Puigt, and M. Montagnac. Block-jacobi implicit algorithm for the time spectral method. *AIAA Journal*, 46(12):3080–3089, 2008.

<sup>17</sup>P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA 1992-0439, Jan. 1992.

<sup>18</sup>E. van der Weide, A. K. Gopinath, and A. Jameson. Turbomachinery applications with the time spectral method. AIAA Paper 2005-4905, June 2005.

<sup>19</sup>A. van Zuijlen and H. Bijl. High order time integration for fluid-structure interaction on moving meshes. AIAA paper 2005-5247, June 2005.

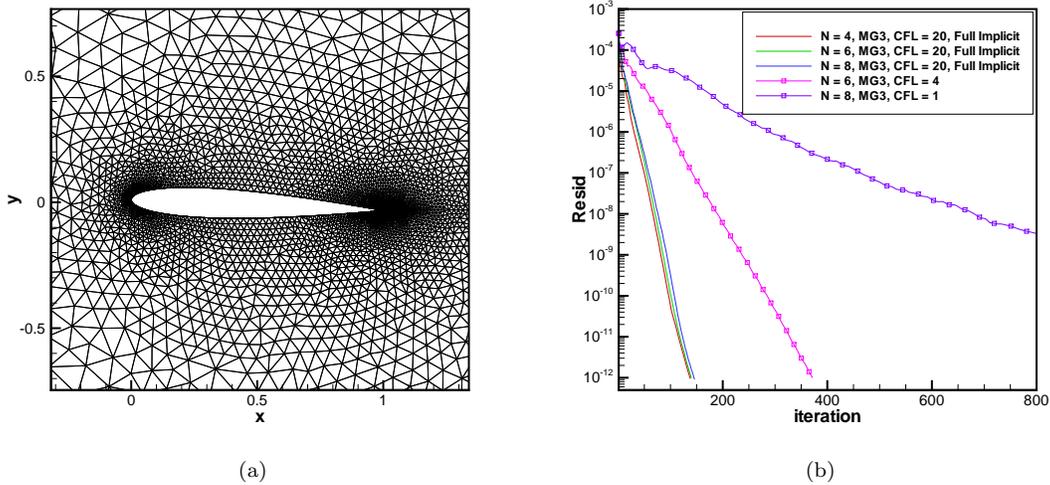


Figure 1. Near field mesh for pitching NACA0012 airfoil case (a), and comparison of convergence histories for local implicit and fully implicit method for varying numbers of time instances (b).

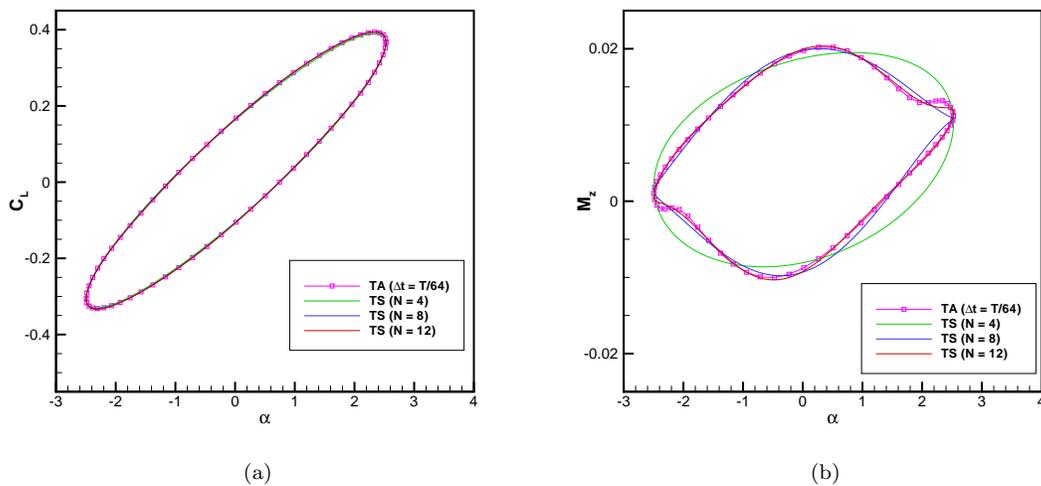


Figure 2. Comparison of computed lift coefficient (a), and moment coefficient (b) using time-accurate and time-spectral solvers for pitching NACA0012 airfoil problem.

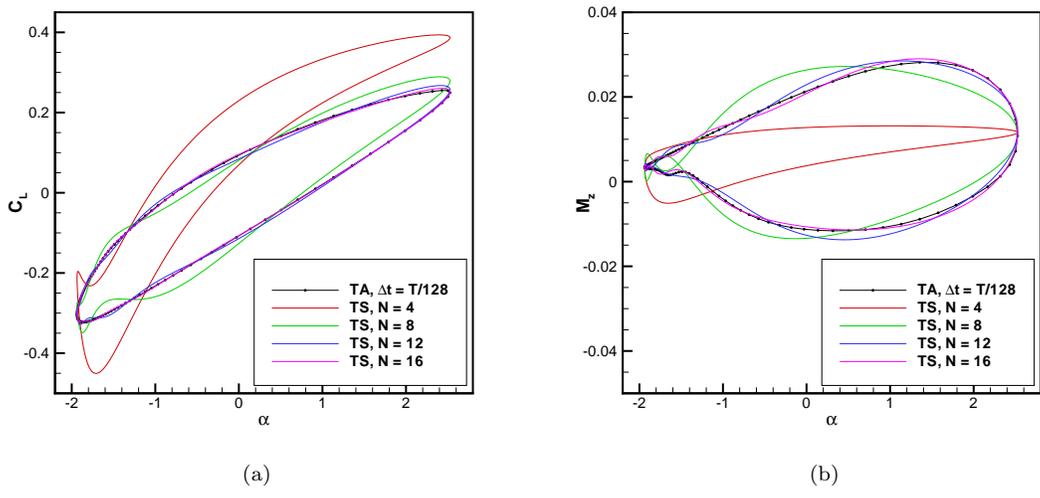


Figure 3. Comparison of computed lift coefficient (a), and moment coefficient (b) using time-accurate and time-spectral solvers for pitching NACA0012 airfoil problem with wide frequency content.

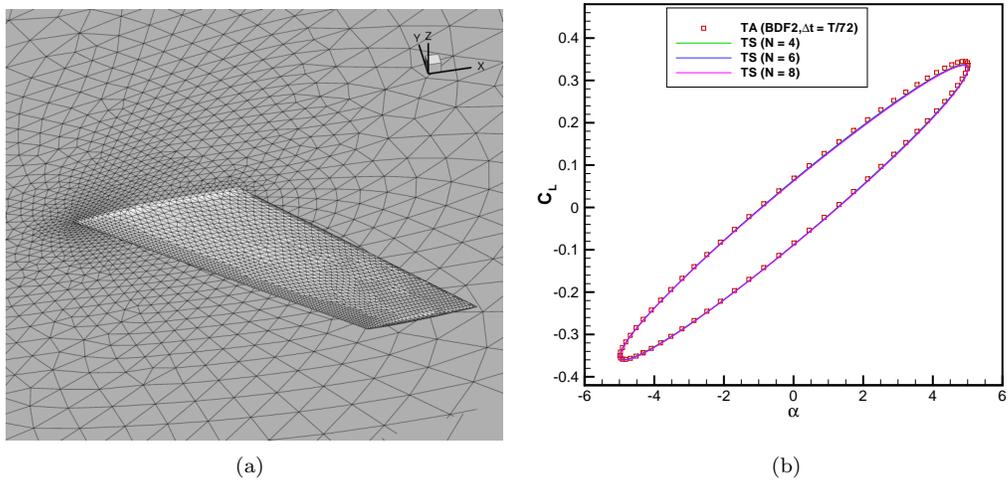


Figure 4. Three dimensional mesh for pitching AGARD 445.6 test case (a), and comparison of computed lift coefficient (b) using time-accurate and time-spectral methods.

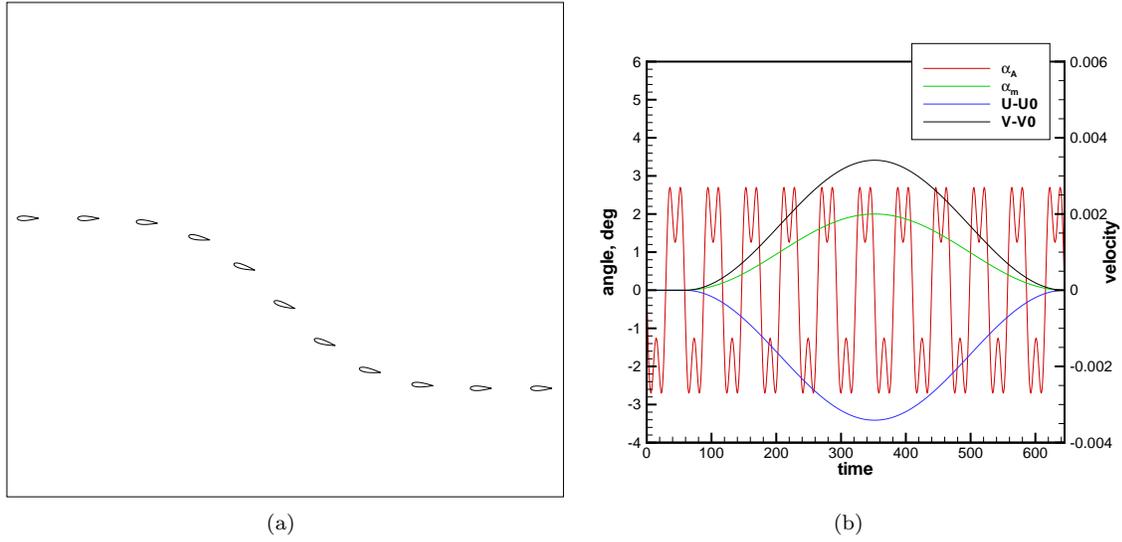


Figure 5. Illustration for the airfoil climbing (a), and forward/upward speed and angle of attack (b) for pitching-climbing airfoil.

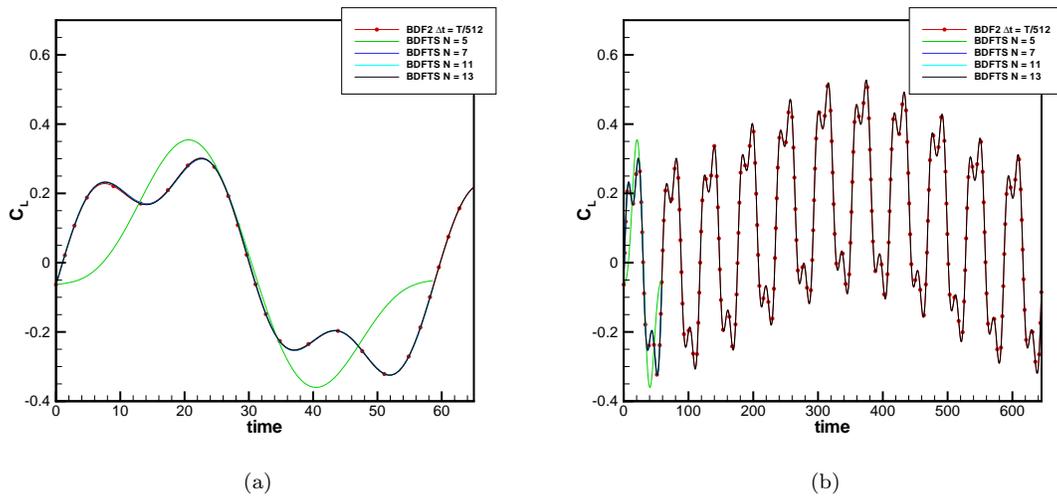


Figure 6. Comparison of computed lift coefficient in first period (a), and 11 periods (b) for pitching-climbing airfoil.

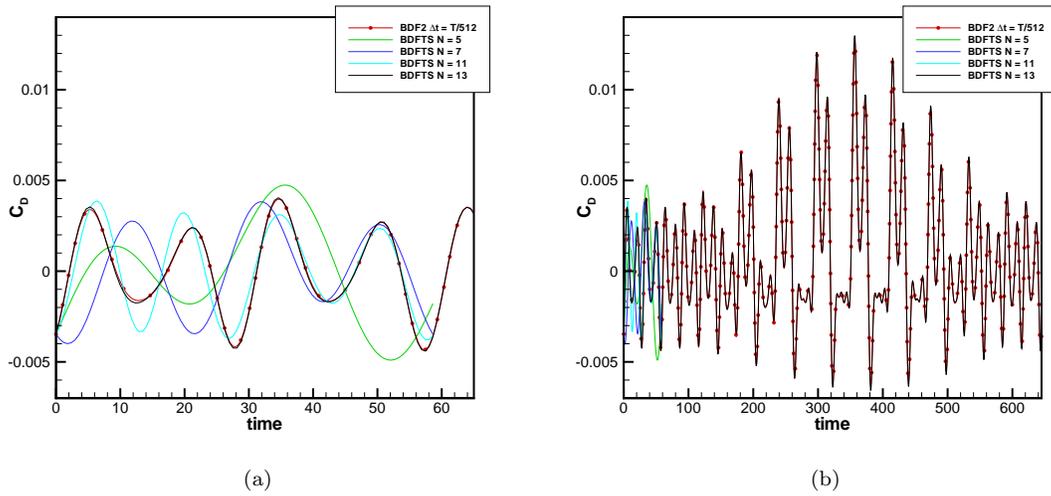


Figure 7. Comparison of computed drag coefficient in first period (a), and in 11 periods (b) for pitching-climbing airfoil.

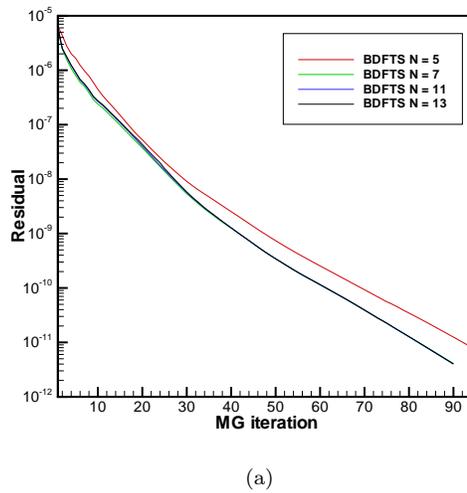


Figure 8. Comparison of convergence history for pitching-climbing airfoil.

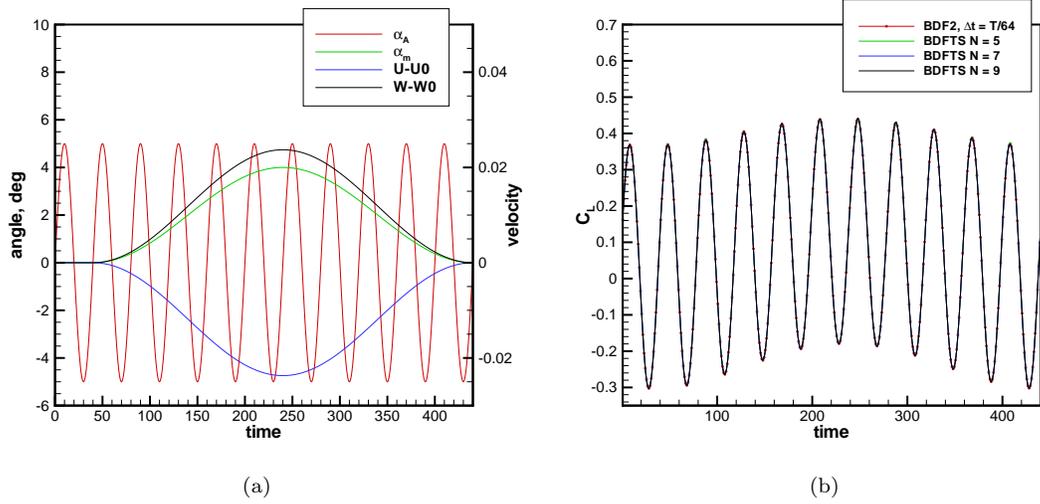


Figure 9. Forward/upward speed and angle of attack (a), and comparison of computed lift coefficient (b) for pitching-climbing wing.

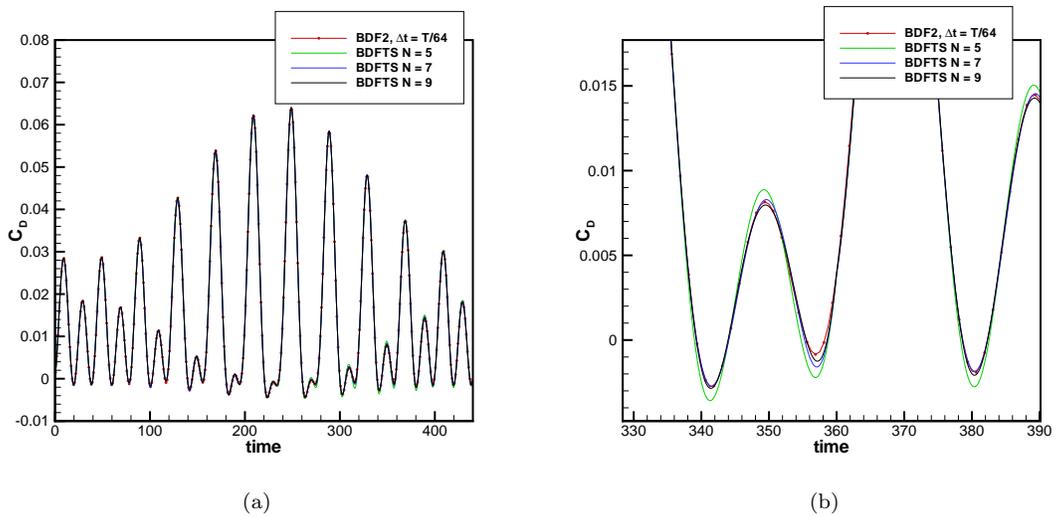
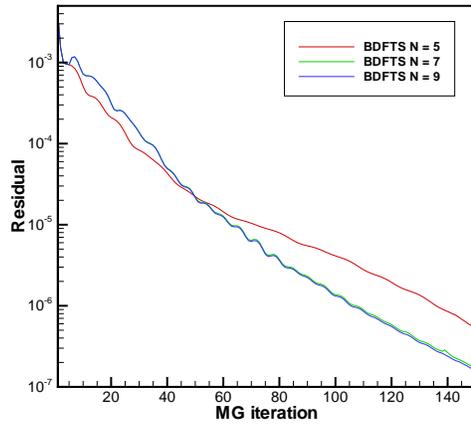
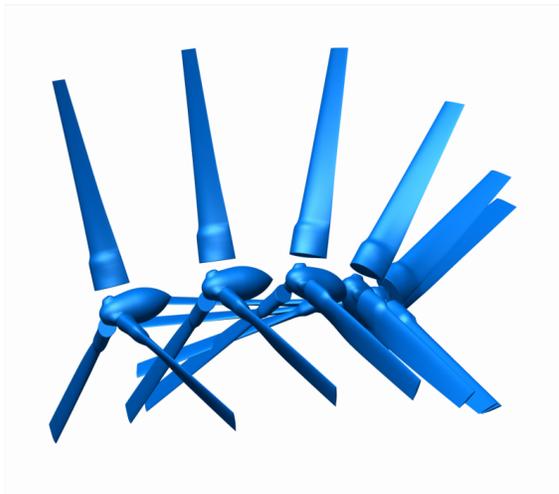


Figure 10. Comparison of computed drag coefficient (a), and zoom in (b) for pitching-climbing wing.

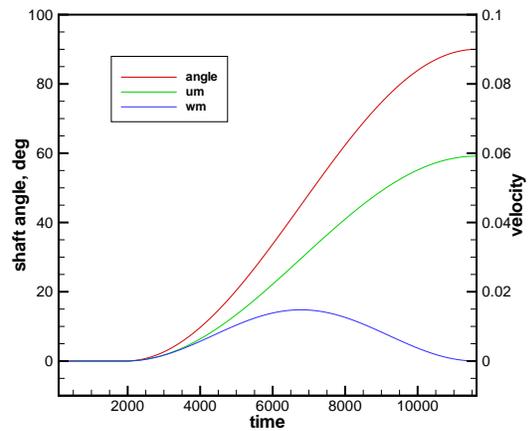


(a)

Figure 11. Comparison of convergence history for pitching-climbing wing.

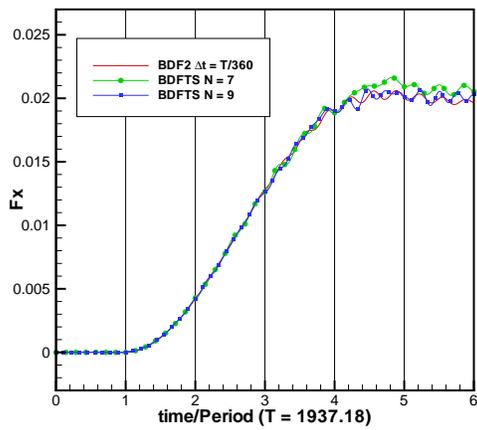


(a)

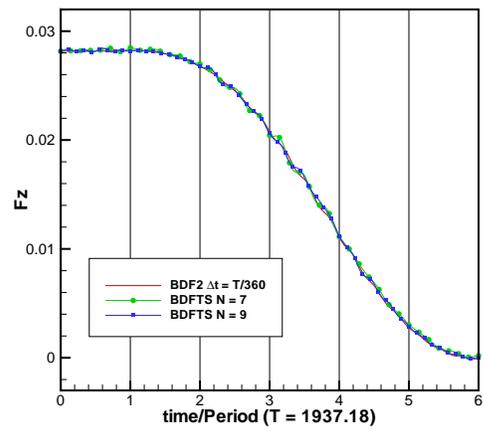


(b)

Figure 12. Illustration the maneuver of a three-bladed rotor from a hover condition to forward flight (a). Forward/upward speed and shaft angle.



(a)



(b)

Figure 13. Comparison of computed force in x (a) and z(b) directions between BDF/Time spectral method and BDF2.