To the University of Wyoming:

The members of the Committee approve the dissertation of Nathan L. Mundis presented on April 30, 2014.

Dr. Dimitri J. Mavriplis, Chairperson

Dr. Craig C. Douglas, External Department Member

Dr. Jonathan W. Naughton

Dr. Jay Sitaraman

Dr. James D. Baeder, University of Maryland – College Park

APPROVED:

Dr. Paul A. Dellenback, Head, Department of Mechanical Engineering

Dr. Khaled A.M. Gasem, Interim Dean, College of Engineering and Applied Science

Mundis, Nathan L., The development of a robust, efficient solver for spectral and spectral-element time discretizations, Ph.D., Department of Mechanical Engineering, May, 2014.

This work examines alternative time discretizations for the Euler equations and methods for the robust and efficient solution of these discretizations. Specifically, the time-spectral method (TS), quasi-periodic time-spectral method (BDFTS), and spectral-element method in time (SEMT) are derived and examined in detail. For the two time-spectral based methods, focus is given to expanding these methods for more complicated problems than have been typically solved by other authors, including problems with spectral content in a large number of harmonics, gust response problems, and aeroelastic problems. To solve these more complicated problems, it was necessary to implement the flexible variant of the Generalized Minimal Residual method (FGMRES), utilizing the full second-order accurate spatial Jacobian, complete temporal coupling of the chosen time discretization, and fully-implicit coupling of the aeroelastic equations in the cases where they are needed. The FGMRES solver developed utilizes a block-colored Gauss-Seidel (BCGS) preconditioner augmented by a defect-correction process to increase its effectiveness. Exploration of more efficient preconditioners for the FGMRES solver is an anticipated topic for future work in this field.

It was a logical extension to apply this already developed FGMRES solver to the spectral-element method in time, which has some advantages over the spectral methods already discussed. Unlike purely-spectral methods, SEMT allows for both h- and p-refinement. This property could allow for element clustering around areas of sharp gradients and discontinuities, which in turn could make SEMT more efficient than TS for periodic problems that contain these sharp gradients and would require many time instances to produce a precise solution using the TS method. As such, a preliminary investigation of the SEMT method applied to the Euler equations is conducted and some areas for needed improvement in future work are identified.

In this work, it is shown that many if not most periodic problems can be solved more quickly and more precisely (utilizing the current FGMRES solver) using the time-spectral method than the currently state-of-the-art second-order accurate time-implicit methods. Additionally, the potential efficiency gains of the quasi-periodic time-spectral method for

strongly-periodic problems, over time-implicit methods for these same types of problems, is demonstrated. Problems with strong, moving shock waves, high reduced frequencies, and/or content in high harmonics (higher than the 20th harmonic) are particularly difficult to solve efficiently using TS, BDFTS, and SEMT, and future work should focus on solver improvements to address these types of problems in particular.

THE DEVELOPMENT OF A ROBUST, EFFICIENT SOLVER FOR SPECTRAL AND SPECTRAL-ELEMENT TIME DISCRETIZATIONS

by

Nathan L. Mundis,
M.S. Aerospace Eng., University of Missouri – Rolla (2007)
B.S. Aerospace Eng., University of Missouri – Rolla (2004)

A dissertation submitted to the
Department of Mechanical Engineering
and the
University of Wyoming
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
MECHANICAL ENGINEERING

Laramie, Wyoming
May 2014

This work is dedicated to my partner, Matthew S. Nichols. How he managed to survive my completing this work will forever remain a mystery.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to acknowledge all of the people instrumental in the preparation of this dissertation. Of course the members of my committee must be thanked first and foremost: Dr. Dimitri Mavripls, Dr. Jay Sitaraman, Dr. James Baeder, Dr. Jon Naughton, and Dr. Craig Douglas. Of my committee members, I must thank my advisor, Dimitri, the most. He has spent countless hours discussing the theory that has gone into this work and suggesting improvements, both to my CFD code and this dissertation. Thank you, Dimitri.

I'd also like to thank all of my co-workers in our CFD lab. They were always there to bounce ideas off of or even receive complaints about the latest, greatest bug that I managed to program into my code. These lab members include: Dr. Markus Rumpfkeil, Dr. Wataru Yamazaki, Dr. Cristian Nastase, Dr. Karthik Mani, Dr. Zhi Yang, Dr. Brian Lockwood, Dr. Nicholas Burgess, Dr. Taisuke Nambu, Dr. Asitav Mishra, Dr. Michael Brazell, Andrew Kirby, Enrico Fabiano, Kevin Brown, and Bryan Flynt.

Further, I would like to thank the sources that have funded this research. First, I thank the Alfred Gessow Vertical Lift Research Center of Excellence at the University of Maryland. Next, I am thankful for ONR Contract N0014-09-1-1060, under the program management of Dr. Judah Milgram. Finally, I give many thanks for AFOSR FA9550-13-C-0015, under the program management of Dr. Michael Kendra.

I'd like to thank the friends and family who have supported through this work: my partner Matthew, my parents Jim and Bonnie Mundis, my sister Michelle and my brothers Michael and Matthew, my faith family at Saint Matthew's Episcopal Cathedral, and all my friends all over the world (you know who you are).

To everyone listed here, and everyone who is not, thank you for your love, patience,

kindness, and encouragement through this long and daunting process. Without the support of everyone in my life, I would never have been able to finish this degree. Thank you, thank you, thank you.

Nathan L. Mundis

University of Wyoming
May 2014

Chapter 1

Introduction

As steady-state computational fluid dynamics (CFD) becomes mature as a field, the need for unsteady CFD solutions and solvers becomes ever more apparent. State-of-the-art unsteady methods typically employ time-implicit backward-difference formulae, most often with second-order accuracy in time. This thesis explores the use of alternative methods to discretize the unsteady time-derivative term in the Euler equations. Though the effects of fluid viscosity are ignored because of the use of the Euler equations, the methods developed in this work should prove readily extensible to unsteady Reynolds-averaged Navier-Stokes discretizations (RANS) as well as more complicated viscous frameworks.

Two distinct frameworks for discretizing the time derivative are explored. First, the time-spectral and quasi-periodic time-spectral methods, which make use of Fourier transforms and spectral bases and are applicable to periodic and strongly periodic problems, are discussed. Then, the spectral-element method in time, whereby time is divided into chunks with each chunk containing a continuous polynomial basis, is explored. The goals in exploring these methods is to attain faster solutions with better accuracy than current state-of-the-art methods. For the time-spectral methods, achievement of these goals has generally been demonstrated only for simpler problems with a low number of harmonics (i.e. time instances) [4–6]. This work looks much more closely at the accuracy and efficiency of these methods, especially when used to solve more difficult problems such as those with high amounts of harmonic content, non-smooth regions, and aeroelasticity. To solve these more

difficult problems efficiently, the need for a far better solver paradigm became apparent. As such, much of this work is focused on the development of such a solver.

Ultimately, this work should be seen as an attempt to seek a much more complete assessment of these methods for a wide range of problems, both flow-alone and aeroelastic, while at the same time improving the solver to make these methods competitive no matter what problem they are used to solve. In the following sections, for both of these time-discretization frameworks, these methods, the class of problems to which they are applicable, and why they are important for the field will be explored in detail.

1.1 Spectral Methods in Time (TS and BDFTS)

The idea of using Fourier series in the solution of time-periodic fluid flow problems has evolved over the last three decades. The first such methods split the flow into a steady and unsteady parts. The steady flow equations would be solved first and then used in the solution of the unsteady contributions. The unsteady equations were written in such a way that the harmonic modes were decoupled and could be solved independently [7, 8]. Adamczyk [9] developed a method that no longer assumed the unsteady contributions would be small. The time variation of the flow was split into a time-averaged part and an unsteady part. Because of the non-linear nature of the Euler equations, time-averaging generated *deterministic stress* terms to provide closure. This method included some, but not all, non-linear unsteady effects.

He [10] developed the nonlinear harmonic method which includes all the nonlinear effects. This method was then further developed for aeroelastic applications [11,12] and blade-row interactions [13]. This discretization requires two interdependent sets of equations which are solved in an alternating, iterative fashion until convergence. Hall et al. has developed the *Harmonic Balance* method which has evolved over the years. The first *Harmonic Balance* method was a frequency domain method for the linearized unsteady Euler equations [14, 15] which were used for oscillating cascades and gust response problems [16]. The next iteration of the *Harmonic Balance* method evolved to pursue full nonlinear effects and time domain

simulations [17, 18]. The *High-dimensional Harmonic Balance* method recently developed is practically indistinguishable from the time-spectral method [19].

McMullen et al. [20, 21] developed the *Non-linear Frequency Domain* method (NLFD), which solves the full non-linear Euler or RANS equations in the frequency domain. This method has high computational costs as at least two fast-Fourier transforms are required at every iteration. The time-spectral method is a natural extension of the NLFD method which recasts the frequency content into the time domain and eliminates the need for Fourier transformation in the iterations [4, 22].

In this work, both the time-spectral and high-dimensional harmonic balance methods are derived using discrete Fourier analysis. These methods, developed by Hall [19], McMullen [20, 21], and Gopinath [22, 23], as discussed above, transform the unsteady equations in the physical domain to a set of steady equations in the frequency domain and then use the time-discretization operator to transform the frequency content back into a discrete number of time instances that reside in the time domain. Each of these time instances is coupled to all other time instances through the time-discretization operator, and the entire system is solved as a single, large, steady-state problem.

For problems with strong periodic content, such as turbomachinery flows or rotorcraft aerodynamics, time-spectral methods can be used to substantially reduce the cost of computing the full, time-dependent solution for a given level of accuracy. In many cases, time-spectral methods using only a small number of time instances per period can provide equivalent or superior accuracy, at substantially reduced cost, compared to traditional time-implicit solutions using hundreds of time steps per period. In other cases, many time instances per period may be needed to resolve both high and low frequency periodic content simultaneously, making time-spectral methods less competitive. In the literature, the time-spectral method has been shown to be faster than the dual-time stepping implicit methods using backwards-difference time formulae for time-periodic computations, such as turbomachinery flows [20, 23], oscillatory pitching airfoil/wing cases [22, 24], flapping wing [25], helicopter rotor [26, 27] and vortex shedding problems [21, 28], which all use small numbers of time instances. However, for problems with large numbers of time instances, it is not yet clear if the

theoretical efficiency gains of TS methods over time-implicit methods can be realized. One of the main goals of this work is to examine the difficulties of using the time-spectral method to solve problems that require large numbers of time instances, and to seek to overcome those difficulties through the implementation of better solution techniques.

Furthermore, the solution of aeroelastic problems introduces yet another layer of complexity and often results in problems with much higher frequency content. In the recent past, the time-spectral method has been proven capable of solving the coupled fluid/structure equations for the purely periodic problem of a helicopter rotor in constant speed forward flight [26, 27]. The simulation of aeroelastic flutter phenomena is computationally expensive and could benefit tremendously from the successful application of a more efficient time-integration scheme. However, time-spectral methods are not directly applicable to most fixed-wing flutter problems since these problems are usually not purely periodic. On the other hand, the recently developed hybrid BDF/time-spectral approach (BDFTS) aims to simulate quasi-periodic flows that contain slow mean flow transients combined with relatively fast periodic content using global BDF time step sizes of the periodic content period length. The BDFTS method makes use of the properties of the time-spectral approach to capture accurate details of the periodic flow components [5, 6, 29]. The aeroelastic flutter problem is a coupled fluid/structural problem with strong periodic content, i.e. the pitching and plunging of the airfoil or wing, and a slow transient, i.e. increasing or decreasing amplitude of the periodic motion. These factors, coupled with the abundance of published solutions to aeroelastic flutter problems [1–3] make it the ideal motivation for a coupled fluid-structure, quasi-periodic time-spectral method.

Further, extending time-spectral methods to address gust response problems pushes the limits of applicability of these methods since the timescale of the gust can be any length, from a small fraction of the period of airfoil or blade (in the case of rotorcraft) motion to several periods of blade motion. Previously, the BDFTS method has only been demonstrated on problems where the transient is on the order of several periods of airfoil or blade motion, and these methods are not necessarily well suited to problems where the transient is less than the period of airfoil or blade motion. The problems presented in the current work are

such that gusts only affect the airfoil section for a fraction of its period of motion and can be solved using the purely-periodic time-spectral method so long as enough spectral harmonics are present to accurately resolve the gust profile. Note that both aeroelastic flutter and gust response are critical problems in the context of both airplanes and rotorcraft and have been traditionally modeled using lower order lifting-line based aerodynamics [30]. Herein, a computationally efficient aeroelastic methodology that can utilize the Euler or Navier-Stokes equations to model the aerodynamics is proposed.

In order to converge the coupled aeroelastic time-spectral system with large numbers of time instances efficiently and robustly, the flexible variant of the Krylov subspace, Generalized Minimal Residual method (FGMRES) [31] is utilized. By using FGMRES, the disparate time instances are much more strongly coupled than they would be if a stationary iterative method, such as the Jacobi method, were used. Additionally, use of FGMRES allows the fluid equations to be coupled to the structural equations and the structural equations to be coupled to the fluid equations implicitly. This implicit fluid/structure coupling has proven key in the development of a robust aeroelastic, time-spectral solver [32, 33].

The TS and BDFTS methods should only be applied to problems with strong periodic content. While the TS method is only applicable to purely periodic problems, the BDFTS method can, technically, be applied to any problem type; however, when the BDFTS method is applied to aperiodic or weakly periodic problems, no efficiency gains are realized over time-implicit methods, so this application is not recommended. Fortunately, many engineering problems of practical importance are strongly periodic. Thus, TS and BDFTS methods fill an important niche in the advancement of the field of CFD.

1.2 Spectral-element Method in Time (SEMT)

The spectral-element method in time (SEMT) discretizes the time dimension into a number of time elements. Within each of these elements, the variation in time is described with polynomial functions. Although this method is referred to as a spectral-element method within the literature [34, 35], it can also be viewed as a continuous-Galerkin finite-element method.

This terminology arises because so-called spectral-element methods exhibit spectral-like error convergence, i.e. as the polynomial degree increases within the elements, the magnitude of the error decreases at an increasing rate. This exponential error convergence is a hallmark of spectral methods, such as the time-spectral method, mentioned above. Since continuous-Galerkin methods exhibit similar error convergence, they are referred to as spectral-element methods.

The SEMT can be used to describe both hyperbolic and periodic problems in time, so it is applicable to all of the same problems as time-implicit methods. This makes SEMT of broader application than TS and BDFTS, which are efficient for purely periodic and quasi-periodic problems, respectively. Unsteady aerodynamic problems of particular interest include analysis of flutter and self-excited oscillation (limit-cycles) [36, 37].

The idea of using space-time finite elements to discretize the time domain was first proposed by Oden [38], Desai et al. [39], Fried [40], and Argyris and Scharpf [41]. These finite elements can be coupled in one of three ways depending on the type of problem to be solved: time-marching backward-only coupling, monolithic-time coupling, and periodic coupling. The first two of these can be applied to all types of problems, while the last is only appropriate for purely periodic problems, just like the time-spectral method.

In backward-only coupling, each element is coupled only to its immediately previous element with subsequent elements being solved sequentially. In the monolithic-time approach, the entire time domain for which a solution needs to be found is solved at once. Elements are coupled both to the immediately previous element and to the next element, meaning that all elements must be solved simultaneously. Kurdi and Beran [34] believe the monolithic time approach is better because "the global-time projection of the response, transforms the time dependent differential equation into an algebraic form, which in contrast to time-marching, presents a strong connection between the unsteady (transient) solution and the system parameters." Apparently, these properties are particularly beneficial in the optimal control of limit-cycle oscillations (LCOs) in various aircraft [42, 43] because global-time allows for the sensitivity of the LCO amplitude to be computed [34].

Time-marching finite elements can employ Hamilton's law of varying action [44],

weighted residual methods [45], discontinuous-Galerkin methods [46], and spectral-element methods [34]. Space-time time-marching finite-elements include formulations based on least squares [47–50] and discontinuous-Galerkin methods [51–56]. The time-marching schemes use the same backward-only coupling as the results presented herein.

Conversely, finite-difference time-marching schemes are much more commonplace and include Runge-Kutta methods [57] and the implicit Newmark scheme [58,59]. Much attention is given to these schemes for the unsteady solution of the Euler and Navier-Stokes equations [60–63].

Like time-spectral methods, SEMT can take advantage of additional parallelism by solving all points in time within a given element simultaneously. This parallelism is extended even further when the monolithic-time or periodic couplings are used, as more points in time can be solved at once. The periodic SEMT might have an advantage over TS for problems with a broad range of harmonic content. In the TS discretization, higher harmonics must be captured by adding time-instances, i.e. through p-refinement alone. In contrast, SEMT might be able to capture these areas of large gradients by clustering spectral elements in time around them, i.e. through h-refinement. This potential advantage of periodic SEMT over TS is one main reason to begin the exploration of the spectral-element method in time.

1.3 Dissertation Overview

This work explores alternative, parallelizable discretizations of the time-derivative term in partial differential equations with a focus on the efficient and robust solution of these methods both when applied to the Euler equations alone and the coupled fluid/structure equations of aeroelasticity. The chapters of the dissertation break down as follows:

- Chapter 2 covers all of the mathematical theory involved in the spatial and temporal discretization of the Euler equations including time-implicit, time-spectral, quasi-periodic time-spectral, and spectral-element in time temporal discretizations
- Chapter 3 covers the structural equations, how they are put into first-order form and how their temporal discretization varies among the methods explored herein

- Chapter 4 examines how all of the different time-discretizations are solved implicitly and how the structural equations are included to provide robust and efficient aeroelastic solutions
- Chapter 5 covers all of the remaining theory and methodology that is used herein but did not seem to fit in the three previous chapters
- Chapter 6 begins the examination of results by looking at flow alone results which are computed using the time-spectral method
- Chapter 7 examines results of aeroelastic problems first by looking at an aeroelastic test problem and then by examining other results including gust response problems, all using the time-spectral method
- Chapter 8 presents results for aeroelastic fixed-wing flutter problems using the quasi-periodic time-spectral (BDFTS) method
- Chapter 9 presents a preliminary look at the spectral-element method in time by examining this method applied to three ordinary differential equations and two solutions of the SEMT Euler equations
- Chapter 10 wraps things up by summarizing this entire document, explaining how this work advances the field of CFD, and looking forward to what improvements can still be made in the future

The methods explored in this dissertation will be shown to solve precisely a wide variety of flow and aeroelastic problems. While it will be shown that great improvements have been made in the stability and efficiency of the solution of these time discretizations, it will also be shown that many improvements remain to be made.

Chapter 2

The Discretization of the Euler Equations

2.1 Governing Equations

The Euler equations in conservative form can be written as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \tag{2.1}$$

where $\mathbf{U}$ represents the vector of conserved quantities (mass, momentum, and energy) which in two dimensions is given as follows:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}$$

and $\mathbf{F}(\mathbf{U})$ represents the convective fluxes, which have x and y components as follows:

$$\mathbf{F_x}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \qquad \mathbf{F_y}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}.$$

Integrating over a (moving) control volume $\Omega(t)$, the following can be obtained:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\Omega(t)} (\nabla \cdot \mathbf{F}(\mathbf{U})) dV = 0. \tag{2.2}$$

Applying the divergence theorem to the second term in the equation above produces the following form for the Euler equations:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) \cdot \tilde{\mathbf{n}}) dS = 0. \tag{2.3}$$

Using the differential identity

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV = \int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} \mathbf{U}(\dot{\mathbf{x}} \cdot \tilde{\mathbf{n}}) dS \tag{2.4}$$

where $\dot{\mathbf{x}}$ and $\tilde{\mathbf{n}}$ are the velocity and normal of the interface $\partial\Omega(t)$, respectively, equation (2.3) becomes:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U}\dot{\mathbf{x}}) \cdot \tilde{\mathbf{n}} dS = 0. \tag{2.5}$$

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

$$\frac{\partial}{\partial t}(V\mathbf{U}) + \mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = 0 \tag{2.6}$$

where $\mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}, \tilde{\mathbf{n}}) = \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}) \cdot \tilde{\mathbf{n}} dS$ is the spatial residual, representing the discrete convective fluxes in arbitrary-Lagrangian-Eulerian (ALE) form and V denotes the control volume. In the discrete form, $\dot{\mathbf{x}}(t)$ and $\tilde{\mathbf{n}}(t)$ now represent the time varying velocities and surface normals of the control-volume boundary faces.

2.2 Spatial Discretization

The Euler equations are discretized by a cell-centered central difference finite-volume scheme on hybrid meshes containing triangles and quadrilaterals in two dimensions. Conservative fluxes are calculated for each face of the control volume "i" with respect to any of its nearest neighbors "k" as shown in Figure (2.1). In general, the conservative flux at a cell face has the following form:

$$\mathbf{F}_{ik}(\mathbf{U}) = \frac{1}{2} [\mathbf{F}_i(\mathbf{U}_i) + \mathbf{F}_k(\mathbf{U}_k)] + \kappa \underline{\mathbf{T}} |\underline{\mathbf{\Lambda}}| \underline{\mathbf{T}}^{-1} (\mathbf{U}_i - \mathbf{U}_k) \tag{2.7}$$

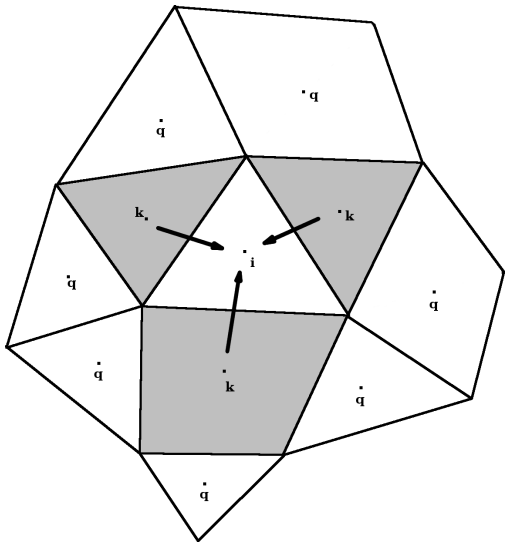


Figure 2.1: Illustration of flux evaluation at the control volume interfaces

The second term on the right hand side of the equation above is additional matrix-based artificial dissipation, which is needed to stabilize the numerical solution of the Euler equations [64]. As written in equation (2.7), the artificial dissipation added is first-order accurate and corresponds to the dissipation matrix multiplied by the Laplacian of the conserved variables. Additionally, $\underline{\mathbf{T}}$ and $\underline{\mathbf{T}}^{-1}$are the left and right eigenvectors of the linearized Euler equations normal to the control volume face between "i" and "k." The matrix $|\underline{\mathbf{\Lambda}}|$ is the diagonal matrix containing the absolute values of the four eigenvalues associated with the two-dimensional Euler equations. Since one eigenvalue is repeated, there are only three distinct eigenvalues: $u_n - \dot{x}_n$, $u_n - \dot{x}_n + c$, and $u_n - \dot{x}_n - c$, where u_n is the fluid velocity normal to the control volume face, $\dot{x}_n$ is the grid velocity normal to the boundary face in question, and c is the speed of sound at that face. When one of these eigenvalues approaches zero, the dissipation for this component also approaches zero, which can lead to numerical instability. To remedy this problem, the eigenvalues are modified as follows:

$$\begin{aligned}
|u_n - \dot{x}_n| &= max\Big[|u_n - \dot{x}_n|, \quad \delta(|u_n - \dot{x}_n| + c)\Big] \\
|u_n - \dot{x}_n + c| &= max\Big[|u_n - \dot{x}_n + c|, \quad \delta(|u_n - \dot{x}_n| + c)\Big] \\
|u_n - \dot{x}_n - c| &= max\Big[|u_n - \dot{x}_n - c|, \quad \delta(|u_n - \dot{x}_n| + c)\Big]
\end{aligned}$$

where $|u_n - \dot{x}_n| + c$ is the maximum eigenvalue and δ is an empirical factor with a value between zero and one. In all cases presented herein, the value of $\delta = 0.1$ is used, which is

also common in many production codes. The process of using a small fraction of the largest eigenvalue when one of the real eigenvalues vanishes is often referred to as an "entropy fix" because, in addition to guaranteeing numerical dissipation for all components of the Euler equations, it additionally eliminates the possibility of an isentropic compression wave, which, though not physically possible, is a numerically possible solution of the Euler equations.

As noted previously, the dissipation presented in equation (2.7) is only first-order accurate. Second-order accuracy, for smoothly varying meshes, is achieved using a two-pass construction of the artificial dissipation operator, which corresponds to an undivided biharmonic operator. The first pass calculates the undivided Laplacian as follows for every element in the spatial domain:

$$\mathbf{L}_i(\mathbf{U}) = \sum_{k=1}^{neighbors} (\mathbf{U}_k - \mathbf{U}_i). \tag{2.8}$$

The second pass calculates second-order matrix dissipation by utilizing the undivided Laplacian, as follows:

$$\mathbf{F}_{ik}(\mathbf{U}) = \frac{1}{2}\left[\mathbf{F}_i(\mathbf{U}_i) + \mathbf{F}_k(\mathbf{U}_k)\right] - \kappa \underline{\mathbf{T}}|\underline{\mathbf{\Lambda}}|\underline{\mathbf{T}}^{-1}[\mathbf{L}_i(\mathbf{U}) - \mathbf{L}_k(\mathbf{U})] \tag{2.9}$$

where all of the variables are as defined previously.

For cases with strong gradients, such as hypersonic flows with strong shock waves, the second-order accurate formulation shown here may cause instabilities. To remedy these instabilities, a blended first- and second-order scheme that uses a switch, such as a pressure switch, to determine the appropriate order can be used [65, 66]. For the transonic and lower speed flows presented herein, shock waves are of moderate to weak strength, if present at all, and such a blended scheme is unnecessary. Thus, all of the cases presented herein are second-order accurate in space throughout the entire domain. It should be noted that this spatial discretization is only formally second-order accurate when used on smoothly varying meshes [64].

2.3 Temporal Discretizations

The main focus of this work is to look at spectral discretizations of the time-derivative term, the first term on the left-hand side of equation (2.6). However, for the sake of comparison, it is also necessary to describe the time-implicit discretization which is used by most state-of-the-art time-accurate CFD codes. The following subsections describe first the time-implicit backward-difference formulae (BDF1 and BDF2) used to discretize the time derivative term, followed by the time-spectral (TS) discretization for this same term. Next, the quasi-periodic time-spectral (BDFTS) discretization is described. Finally, the discretization of the time-derivative term via spectral elements in time (SEMT) is discussed.

2.3.1 Time-implicit Method (BDF2)

The time-implicit method discretizes the time derivative term using backward-difference finite-difference formulae. The first-order accurate backward difference formula (BDF1) is as follows:

$$\frac{\partial}{\partial t}(V\mathbf{U}) = \frac{(V\mathbf{U})^n - (V\mathbf{U})^{n-1}}{\Delta t} \tag{2.10}$$

where $(V\mathbf{U})^n$ uses the values for the control volume and conserved variables from the current time step while $(V\mathbf{U})^{n-1}$ uses values for those same quantities from the previous time step. The time-discretization error given by this formula is first-order accurate $O(\Delta t)$ indicating that by making the time step size half of its previous value, the error would reduce by that same factor of a half. The backward difference formula that is second-order accurate $O(\Delta t^2)$ (BDF2) is given by the following:

$$\frac{\partial}{\partial t}(V\mathbf{U}) = \frac{3(V\mathbf{U})^n - 4(V\mathbf{U})^{n-1} + (V\mathbf{U})^{n-2}}{2\Delta t} \tag{2.11}$$

where the additional $(V\mathbf{U})^{n-2}$ term uses the values for the control volume and conserved variables from two time steps ago. For the second-order accurate formula, reducing the time-step size by a factor of two should result in a four-fold reduction in the temporal error. Higher-order backward difference formulae exist and follow a similar pattern; for instance,

the third-order backward difference is given as follows:

$$\frac{\partial}{\partial t}(V\mathbf{U}) = \frac{11(V\mathbf{U})^n - 18(V\mathbf{U})^{n-1} + 9(V\mathbf{U})^{n-2} + -2(V\mathbf{U})^{n-3}}{6\Delta t}$$

which now include the values for the control volume and conserved variables for the three previous time steps. However, third-order and higher-order backward difference formulae are not A-stable, meaning that convergence to the wrong answer, or divergence itself, may result when these higher-order difference formulae are used [67]. This limit is known as the Second Dahlquist barrier. As such, the second-order accurate BDF2 is the highest order time-implicit formulation in common use. Thus, BDF2 is is used as the baseline to which the various spectral time discretizations are compared. The BDF2 Euler equations are given as follows:

$$\frac{3(V\mathbf{U})^n - 4(V\mathbf{U})^{n-1} + (V\mathbf{U})^{n-2}}{2\Delta t} + \mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n) = 0 \tag{2.12}$$

2.3.2 Time-spectral Method (TS)

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 [22]:

$$\widehat{\mathbf{U}}_k = \frac{1}{N}\sum_{j=0}^{N-1} \mathbf{U}^j e^{-if_k t_j} \tag{2.13}$$

where N is the number of time intervals and $t_j = \frac{jT}{N}$ and $f_k = k\frac{2\pi}{T}$. The Fourier inverse transform is then given as follows:

$$\mathbf{U}^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \widehat{\mathbf{U}}_k e^{if_k t_n} \tag{2.14}$$

where, similarly, $t_n = \frac{nT}{N}$. It should be noted that $\frac{N}{2}$ is an integer division operation, so for odd numbers of time instances, equation (2.14) can be rewritten as follows:

$$\mathbf{U}^n = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \widehat{\mathbf{U}}_k e^{if_k t_n}. \tag{2.15}$$

Going forward, derivations are split into separate sections for even numbers and odd numbers of time instances.

Even Numbers of Time Instances

Differentiating equation (2.14) in time, the following it obtained:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} if_k \widehat{\mathbf{U}}_k e^{if_k t_n} \tag{2.16}$$

for an even number of time instances; odd numbers of time instances will be treated subsequently. Now, substituting equation (2.13) into equation (2.16), the following double sum is obtained:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{j=0}^{N-1} if_k \mathbf{U}^j e^{-if_k t_j} e^{if_k t_n}. \tag{2.17}$$

The $-\frac{N}{2}$ wave number should be set to zero since it has no corresponding positive counterpart. For instance, if $N = 4$, the wave numbers k over which the modes are summed in equation (2.17) are $k \in \{-2, -1, 0, 1\}$. Note that -2 is included, but $+2$ is not. This means that the $\sin(f_2 t)$ component of the signal can be resolved, but the $\cos(f_2 t)$ component cannot, implying that the second harmonic must necessarily have a phase angle of zero. Though this may be the case in some instances, it is not generally true; thus, the $-\frac{N}{2}$ wave number is eliminated from the summation. Additionally, the summations are switched thanks to the commutative property of addition and the substitutions $t_j = \frac{jT}{N}$ and $t_n = \frac{nT}{N}$ are utilized:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} if_k \mathbf{U}^j e^{-if_k \frac{Tj}{N}} e^{if_k \frac{Tn}{N}}. \tag{2.18}$$

It is desired that the final form on the time derivative term should appear as follows:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=0}^{N-1} d_n^j \mathbf{U}^j \tag{2.19}$$

where the d_n^j are the time-spectral coupling coefficients. These coefficients involve simplifying the following sum where $f_k = k\frac{2\pi}{T}$ has been substituted and simplified:

$$d_n^j = \frac{2\pi}{N \times T} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} ike^{ik\frac{2\pi}{N}(n-j)}. \tag{2.20}$$

The sum given in this expression can be evaluated as follows. It is best to begin with the following:

$$S = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} e^{ik\frac{2\pi}{N}(n-j)} = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} e^{ikx} \tag{2.21}$$

where $x = \frac{2\pi}{N}(n-j)$. Expanding this series, the following is obtained:

$$\begin{aligned} S = &\ e^{i(-\frac{N}{2}+1)x} + e^{i(-\frac{N}{2}+2)x} + ... + e^{i(\frac{N}{2}-2)x} + e^{i(\frac{N}{2}-1)x} \\ S = &\ e^{i(-\frac{N}{2}+1)x}[1 + e^{ix} + e^{2ix} + ... + e^{(N-2)ix}] \\ S = &\ e^{i(-\frac{N}{2}+1)x}\frac{1-e^{(N-1)ix}}{1-e^{ix}} = \frac{e^{(-\frac{N}{2}+1)ix}-e^{\frac{N}{2}ix}}{1-e^{ix}} \\ S = &\ \frac{e^{\frac{-N+1}{2}ix}-e^{\frac{N-1}{2}ix}}{e^{-\frac{1}{2}ix}-e^{\frac{1}{2}ix}} = \frac{\sin(\frac{N-1}{2}x)}{\sin(\frac{x}{2})}. \end{aligned} \tag{2.22}$$

This expression can then be differentiated with respect to x to yield the needed sum:

$$\frac{dS}{dx} = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} ike^{ikx} = \frac{(\frac{N-1}{2})\cos(\frac{N-1}{2}x)\sin(\frac{x}{2}) - \frac{1}{2}\cos(\frac{x}{2})\sin(\frac{N-1}{2}x)}{(\sin\frac{x}{2})^2} \tag{2.23}$$

by making use of the definition of x given previously and using the following trigonometric identities:

$$\begin{aligned} \sin\left(\frac{Nx}{2} - \frac{x}{2}\right) &= -(-1)^{n-j}\sin\left(\frac{x}{2}\right) \\ \cos\left(\frac{Nx}{2} - \frac{x}{2}\right) &= (-1)^{n-j}\cos\left(\frac{x}{2}\right) \end{aligned} \tag{2.24}$$

the expression for $^{dS}/_{dx}$ can be simplified as follows:

$$\frac{dS}{dx} = \frac{(\frac{N-1}{2})(-1)^{n-j}\cos(\frac{x}{2})sin(\frac{x}{2}) + \frac{1}{2}(-1)^{n-j}\cos(\frac{x}{2})\sin(\frac{x}{2})}{(\sin\frac{x}{2})^2}. \tag{2.25}$$

Further simplified, this becomes as follows:

$$\frac{dS}{dx} = \frac{\frac{N}{2}(-1)^{n-j}\cos(\frac{x}{2})sin(\frac{x}{2})}{(\sin\frac{x}{2})^2} = \frac{N}{2}(-1)^{n-j}\cot\frac{x}{2}. \tag{2.26}$$

Replacing x with what it represents $\frac{2\pi}{N}(n-j)$ and substituting equation (2.26) for the series given in equation (2.20), the final form of the time-spectral coefficients for even numbers of time instances is found as follows:

$$d_n^j = \begin{cases} \frac{\pi}{T}(-1)^{n-j}\cot(\frac{\pi(n-j)}{N}) & n \neq j \\ 0 & n = j \end{cases}. \tag{2.27}$$

Odd Numbers of Time Instances

The formulation for the time-spectral coupling coefficients for odd numbers of time-spectral time instances follows very closely that for even numbers of time instances. All the steps are

identical except equation (2.15) is differentiated instead of equation (2.14), which results in the following:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} if_k \widehat{\mathbf{U}}_k e^{if_k t_n}. \tag{2.28}$$

Following the same steps as given for even numbers of time instances, the following equation for the time-spectral coupling coefficients, which is analogous to equation (2.20) for even numbers of time instances, is found.

$$d_n^j = \frac{2\pi}{N \times T} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}-1} ike^{ik\frac{2\pi}{N}(n-j)}. \tag{2.29}$$

The sum on the right hand side of this equation must be simplified. The following sum is a good starting point:

$$S = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{ik\frac{2\pi}{N}(n-j)} = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{ikx}. \tag{2.30}$$

The following steps expand and simplify this sum:

$$\begin{aligned} S = &\ e^{i(-\frac{N-1}{2})x} + e^{i(-\frac{N-1}{2}+1)x} + ... + e^{i(\frac{N-1}{2}-1)x} + e^{i(\frac{N-1}{2})x} \\ S = &\ e^{i(-\frac{N-1}{2})x}[1 + e^{ix} + e^{2ix} + ... + e^{(N-2)ix}] \\ S = &\ e^{i(-\frac{N-1}{2})x}\frac{1-e^{Nix}}{1-e^{ix}} = \frac{e^{(-\frac{N-1}{2})ix} - e^{\frac{N-1}{2}ix}}{1-e^{ix}} \\ S = &\ \frac{e^{(-\frac{N}{2})ix} - e^{(\frac{N}{2})ix}}{e^{-\frac{1}{2}ix} - e^{\frac{1}{2}ix}} = \frac{\sin(\frac{N}{2}x)}{\sin(\frac{x}{2})}. \end{aligned} \tag{2.31}$$

Taking the derivative of S, $\frac{dS}{dx}$, the following form is obtained:

$$\frac{dS}{dx} = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} ike^{ikx} = \frac{(\frac{N}{2})\cos(\frac{N}{2}x)\sin(\frac{x}{2}) - \frac{1}{2}\cos(\frac{x}{2})\sin(\frac{N}{2}x)}{(\sin\frac{x}{2})^2}. \tag{2.32}$$

Using the following trigonometric identities:

$$\begin{aligned} \sin\left(\tfrac{Nx}{2}\right) &= 0 \\ \cos\left(\tfrac{Nx}{2}\right) &= (-1)^{n-j} \end{aligned} \tag{2.33}$$

and the definition of $x = \frac{2\pi}{N}(n-j)$, the following form the derivative of this sum S is produced:

$$\frac{dS}{dx} = \frac{(\frac{N}{2})(-1)^{n-j}sin(\frac{x}{2})}{(sin\frac{x}{2})^2} = \frac{N}{2}(-1)^{n-j}\csc\left(\frac{x}{2}\right). \tag{2.34}$$

Finally, substituting this expression into equation (2.28) the following expression for the time-spectral coupling coefficients for odd numbers of time instances is found:

$$d_n^j = \begin{cases} \frac{\pi}{T}(-1)^{n-j}\csc(\frac{\pi(n-j)}{N}) & n \neq j \\ 0 & n = j \end{cases} . \tag{2.35}$$

It should be apparent for both even and odd numbers of time instances, that

$$d_n^j = -d_j^n = d_{n+1}^{j+1} = d_{n-1}^{j-1} = etc.$$

meaning that, when written in matrix form, the time-spectral coupling matrix will be anti-symmetric with constant-value diagonals. Given the above relations of the d_n^J coefficients, a small simplification to the equations for the time-spectral coupling coefficients can be made as follows:

$$d_m^{even} = \begin{cases} \frac{\pi}{T}(-1)^{m}\cot(\frac{\pi m}{N}) & m \neq 0 \\ 0 & m = 0 \end{cases} \tag{2.36}$$

and

$$d_m^{odd} = \begin{cases} \frac{\pi}{T}(-1)^{m}\csc(\frac{\pi m}{N}) & m \neq 0 \\ 0 & m = 0 \end{cases} \tag{2.37}$$

where $m = (n - j)$.

Final Form of the TS Euler Equations

Finally, the time derivative term given by equation (2.19) is substituted into the Euler equations given by equation (2.6), and it is required that equation (2.6) hold exactly at the same N discrete locations in time (i.e. multiply equation (2.6) by the Dirac delta test function $\delta(t-t^n)$ and integrate over all time). This process yields the following time-spectral governing equation:

$$\sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j + \mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n) = 0 \qquad n = 0, 1, 2, ..., N-1. \tag{2.38}$$

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 the spatial residuals be evaluated at the appropriate temporal locations.

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 together due to this coupling.

Also note that the time-spectral discretization 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 $\widehat{\mathbf{U}}_k$) are determined by requiring $\mathbf{U}(t)$ to be equal to its projection at N discrete locations in time, as given by equations (2.13-2.15).

It should also be noted that in all the time-spectral cases that appear in the remainder of this thesis, an odd number of time-spectral time instances has been used. It should be recalled that in the formulation of the time-spectral coupling coefficients for an even number of time instances, there was a negative wave number which had no corresponding positive counterpart. As such, this wave number was dropped from the formulation of the coefficients. It can be seen that, as a result, when $\frac{(n-j)}{N} = \frac{1}{2}$ in equation (2.27), $\cot(\frac{\pi(n-j)}{N}) = 0$, thus $d_n^j = 0$. In other words, opposite time instances remain uncoupled, e.g. if there are four time instances numbered 0, 1, 2, and 3, time instance 0 is coupled to 1 and 3, but is uncoupled from 2. Similarly, time instances 1 and 3 would also be uncoupled. As a result, the time-spectral coupling matrices (i.e. the coefficients written in matrix form) have the following form for even and odd numbers of time instances, respectively:

$$D^{even} = \begin{bmatrix} 0 & d_1^{even} & ... & d_{\frac{N-1}{2}}^{even} & 0 & -d_{\frac{N-1}{2}}^{even} & ... & -d_1^{even} \\ -d_1^{even} & 0 & d_1^{even} & ... & d_{\frac{N-1}{2}}^{even} & 0 & ... & -d_2^{even} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ -d_{\frac{N-1}{2}}^{even} & ... & -d_1^{even} & 0 & d_1^{even} & ... & d_{\frac{N-1}{2}}^{even} & 0 \\ 0 & -d_{\frac{N-1}{2}}^{even} & ... & -d_1^{even} & 0 & d_1^{even} & ... & d_{\frac{N-1}{2}}^{even} \\ d_{\frac{N-1}{2}}^{even} & 0 & -d_{\frac{N-1}{2}}^{even} & ... & -d_1^{even} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & d_1^{even} \\ d_1^{even} & d_2^{even} & ... & 0 & ... & -d_2^{even} & -d_1^{even} & 0 \end{bmatrix} \tag{2.39}$$

$$D^{odd} = \begin{bmatrix} 0 & d_1^{odd} & ... & d_{\frac{N}{2}-1}^{odd} & -d_{\frac{N}{2}-1}^{odd} & ... & -d_1^{odd} \\ -d_1^{odd} & 0 & d_1^{odd} & ... & d_{\frac{N}{2}-1}^{odd} & ... & -d_2^{odd} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \vdots \\ -d_{\frac{N}{2}-1}^{odd} & ... & -d_1^{odd} & 0 & d_1^{odd} & ... & d_{\frac{N}{2}-1}^{odd} \\ d_{\frac{N}{2}-1}^{odd} & -d_{\frac{N}{2}-1}^{odd} & ... & -d_1^{odd} & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & d_1^{odd} \\ d_1^{odd} & d_2^{odd} & ... & ... & -d_2^{odd} & -d_1^{odd} & 0 \end{bmatrix}. \tag{2.40}$$

As can be seen, the matrix D^{even} has two zeroes in each row and in each column, whereas D^{odd} only has one zero. As a result, D^{even} has two eigenvalues that are zero with corresponding eigenvectors $e_1 = (1, 1, ..., 1, 1)^T$ and $e_2 = (1, 0, 1, 0, ..., 1, 0, 1, 0)^T$. The first of these corresponds to a zero time derivative for a steady-state, constant-in-time solution, which is required for a consistent scheme. The second of these eigenvectors, however, has a zero time derivative, but clearly corresponds to an odd-even decoupled solution, which is not damped by D^{even}. D^{odd} has the first of these eigenvectors, but not the second. As a result, D^{even} can be unstable, especially when the values of d_n^j are large, corresponding to problems with short periods. On the other hand, D^{odd} has no such instability as a result, only odd numbers of time instances are used in this work. Additionally, less computation must be done for odd numbers of time instances since, for example, 7 and 8 time instances can resolve exactly the same number of modes of harmonic content, but using 7 obviously results in a smaller time-spectral matrix.

2.3.3 Quasi-periodic Time-spectral Method (BDFTS)

Although the time-spectral method is only applicable to purely periodic problems, a modification of the method can be used to treat problems that have strong periodic content but also mean flow transients, such as maneuvering rotorcraft or fixed wing flutter. For these types of problems the quasi-periodic time-spectral method is ideal. The quasi-periodic time-spectral form is derived through the use of polynomial subtraction for quasi-periodic

functions by subtracting out the non-periodic transient, which can be modeled using a polynomial basis, and approximating the remaining purely periodic component with a spectral basis [68]. 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).

First, the quasi-periodic temporal variation of the solution is split into a periodic contribution and a slowly varying mean flow as follows:

$$\mathbf{U}(t) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \widehat{\mathbf{U}}_k e^{if_k t_j} + \bar{\mathbf{U}}(t) \tag{2.41}$$

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

$$\bar{\mathbf{U}}(t) = \phi_{12}(t)\mathbf{U}^m + \phi_{11}(t)\mathbf{U}^{m-1} \tag{2.42}$$

for a linear variation and

$$\bar{\mathbf{U}}(t) = \phi_{23}(t)\mathbf{U}^m + \phi_{22}(t)\mathbf{U}^{m-1} + \phi_{21}(t)\mathbf{U}^{m-2} \tag{2.43}$$

for a quadratic variation in time. Here $\mathbf{U}^{m-1}$ and $\mathbf{U}^m$ represent discrete solution instances in time usually taken as the beginning and ending points, respectively, of the considered period in the quasi-periodic motion (and $\mathbf{U}^{m-2}$ corresponds to the beginning point of the previous period). In the first case, $\phi_{12}(t)$ and $\phi_{11}(t)$ correspond to linear interpolation functions, which are derived as follows. First, begin with a Taylor series approximation for $\bar{\mathbf{U}}(t)$:

$$\bar{\mathbf{U}}(t) = \mathbf{U}^{m-1} + \frac{d\bar{\mathbf{U}}(t)}{dt}(t - t_{m-1}) + \frac{1}{2}\frac{d^2\bar{\mathbf{U}}(t)}{dt_2}(t - t_{m-1})^2 + H.O.T.s. \tag{2.44}$$

For a linear interpolation, keep the first two terms and use a first-order backward difference approximation for the time derivative:

$$\bar{\mathbf{U}}(t) = \mathbf{U}^{m-1} + \frac{\mathbf{U}^m - \mathbf{U}^{m-1}}{T}(t - t_{m-1}) = \frac{\mathbf{U}^m(t - t_{m-1}) + \mathbf{U}^{m-1}(T - t + t_{m-1})}{T} \tag{2.45}$$

where $T = t_m - t_{m-1}$. Separating out the $\mathbf{U}^m$ and $\mathbf{U}^{m-1}$ terms and simplifying, this becomes the following:

$$\bar{\mathbf{U}}(t) = \frac{(t - t_{m-1})}{T}\mathbf{U}^m + \frac{(t_m - t)}{T}\mathbf{U}^{m-1}. \tag{2.46}$$

By comparing this to equation (2.42) it is easy to see that the linear interpolation functions have the following form:

$$\phi_{11}(t) = \frac{t_m - t}{T} \tag{2.47}$$

$$\phi_{12}(t) = \frac{t - t_{m-1}}{T} \tag{2.48}$$

with the period given as $T = t_m - t_{m-1}$. Similarly, the $\phi_{23}(t), \phi_{22}(t), \phi_{21}(t)$ are given by the corresponding quadratic interpolation functions, which are derived by keeping the first three terms in the Taylor series given by equation (2.44), using a second-order central-difference approximation for the first and second derivative terms, as follows:

$$\bar{\mathbf{U}}(t) = \mathbf{U}^{m-1} + \frac{\mathbf{U}^m - \mathbf{U}^{m-2}}{2T}(t - t_{m-1}) + \frac{1}{2}\frac{\mathbf{U}^m - 2\mathbf{U}^{m-1} + \mathbf{U}^{m-2}}{T^2}(t - t_{m-1})^2 \tag{2.49}$$

Separating the terms for the different $\mathbf{U}$ locations generates the following:

$$\begin{aligned}\bar{\mathbf{U}}(t) = \frac{1}{2}\left[\frac{(t - t_{m-1})}{T} + \frac{(t - t_{m-1})^2}{T_2}\right]\mathbf{U}^m + \left[1 - \frac{(t - t_{m-1})^2}{T^2}\right]\mathbf{U}^{m-1} \\ + \frac{1}{2}\left[-\frac{(t - t_{m-1})}{T} + \frac{(t - t_{m-1})^2}{T^2}\right]\mathbf{U}^{m-2}.\end{aligned} \tag{2.50}$$

By comparison with equation (2.43) the following expressions for the quadratic interpolation functions should be clear:

$$\phi_{21}(t) = \frac{1}{2}\left[-\frac{(t - t_{m-1})}{T} + \frac{(t - t_{m-1})^2}{T^2}\right] \tag{2.51}$$

$$\phi_{22}(t) = 1 - \frac{(t - t_{m-1})_2}{T_2} \tag{2.52}$$

$$\phi_{23}(t) = \frac{1}{2}\left[\frac{(t - t_{m-1})}{T} + \frac{(t - t_{m-1})^2}{T^2}\right]. \tag{2.53}$$

To verify the consistency of these interpolation functions, keeping in mind that $T = t_m - t_{m-1} = t_{m-1} - t_{m-2}$, the following should be noted:

$$\begin{aligned}\phi_{21}(t_{m-2}) = 1 \quad \phi_{22}(t_{m-2}) = 0 \quad \phi_{23}(t_{m-2}) = 0 \\ \phi_{21}(t_{m-1}) = 0 \quad \phi_{22}(t_{m-1}) = 1 \quad \phi_{23}(t_{m-1}) = 0 \\ \phi_{21}(t_m) = 0 \qquad \phi_{22}(t_m) = 0 \qquad \phi_{23}(t_m) = 1\end{aligned}$$

thus the following is confirmed:

$$\bar{\mathbf{U}}(t_{m-2}) = \mathbf{U}^{m-2} \qquad \bar{\mathbf{U}}(t_{m-1}) = \mathbf{U}^{m-1} \qquad \bar{\mathbf{U}}(t_m) = \mathbf{U}^m. \tag{2.54}$$

Returning to the general formulation of the BDFTS method, the collocation approximation leads to the determination of the Fourier coefficients as:

$$\widehat{\mathbf{U}}_k = \frac{1}{N}\sum_{n=0}^{N-1} \tilde{\mathbf{U}}^j e^{-if_k t_j} \tag{2.55}$$

with $\tilde{\mathbf{U}}^j = \mathbf{U}^j - \bar{\mathbf{U}}^j$ defined as the remaining periodic component of the function after polynomial subtraction. Differentiating equation (2.41) and making use of equations (2.19) and (2.55), the following expression for the time derivative is obtained:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=0}^{N-1} d_n^j \tilde{\mathbf{U}}^j + \phi'_{12}(t_n)\mathbf{U}^m + \phi'_{11}(t_n)\mathbf{U}^{m-1} \tag{2.56}$$

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

$$t_j = t_{m-1} + \frac{j}{N}(t_m - t_{m-1}), \qquad j = 0, \ldots, N-1.$$

It should also be noted that $\bar{\mathbf{U}}(t_m) = \mathbf{U}^m = \mathbf{U}(t_m)$ and thus $\tilde{\mathbf{U}}^0 = 0$. 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 (2.56) in terms of the original time instances $\mathbf{U}^n$, the following is obtained:

$$\begin{aligned}\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=1}^{N-1} d_n^j \mathbf{U}^j - (\sum_{j=1}^{N-1} d_n^j \phi_{12}(t_j) - \phi'_{12}(t_n))\mathbf{U}^m \\ -(\sum_{j=1}^{N-1} d_n^j \phi_{11}(t_j) - \phi'_{11}(t_n))\mathbf{U}^{m-1}.\end{aligned} \tag{2.57}$$

Finally, the above expression for the time derivative is substituted into equation (2.6) which is then required to hold exactly at time instances $j = 1, 2, ..., N-1$ and $j = N$ (which corresponds to the m time instance):

$$\begin{aligned}\sum_{j=1}^{N-1} d_n^j V^j \mathbf{U}^j - (\sum_{j=1}^{N-1} d_n^j \phi_{12}(t_j) - \phi'_{12}(t_n))V^m\mathbf{U}^m \\ -(\sum_{j=1}^{N-1} d_n^j \phi_{11}(t_j) - \phi'_{11}(t_n))V^{m-1}\mathbf{U}^{m-1} \\ +\mathbf{R}(\mathbf{U^n}, \dot{\mathbf{x}}^\mathbf{n}, \tilde{\mathbf{n}}^\mathbf{n}) = 0 \qquad n = 1, 2, ..., N.\end{aligned} \tag{2.58}$$

As previously, there are N coupled equations for the N unknown time instances, although in this case the $j = 0$ time instance which corresponds to the $\mathbf{U}^{m-1}$ values are known from the solution of the previous period, while the $j = N$ or $\mathbf{U}^m$ 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 (2.56) 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, $\mathbf{U}^m = \mathbf{U}^{m-1}$ 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 $\mathbf{U}^m = \mathbf{U}^{m-1}$, the last equation at $j = N$ becomes identical to the $j = 0$ equation and the time-spectral method given by equation (2.38) is recovered.

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

$$\begin{aligned}\sum_{j=1}^{N-1} d_n^j V^j \mathbf{U}^j - (\sum_{j=1}^{N-1} d_n^j \phi_{23}(t_j) - \phi'_{23}(t_n)) V^m \mathbf{U}^m \\ -(\sum_{j=1}^{N-1} d_n^j \phi_{22}(t_j) - \phi'_{22}(t_n)) V^{m-1} \mathbf{U}^{m-1} \\ -(\sum_{j=1}^{N-1} d_n^j \phi_{21}(t_j) - \phi'_{21}(t_n)) V^{m-2} \mathbf{U}^{m-2} \\ +\mathbf{R}(\mathbf{U^n}, \dot{\mathbf{x}}^\mathbf{n}, \tilde{\mathbf{n}}^\mathbf{n}) = 0 \qquad n = 1, 2, ..., N\end{aligned} \tag{2.59}$$

where the ϕ values are as given previously and the values $\mathbf{U}^{m-2}$ and $\mathbf{U}^{m-1}$, which correspond to the time instances at the beginning and end of the previous period, respectively, are known from the solution of earlier periods, and $\mathbf{U}^m = \mathbf{U}^N$ as previously. The ϕ' terms have the

following expressions:

$$\phi'_{21}(t) = \frac{-\frac{1}{2} + \frac{(t-t_{m-1})}{T}}{T} \tag{2.60}$$

$$\phi'_{22}(t) = -2\frac{\frac{(t-t_{m-1})}{T}}{T} \tag{2.61}$$

$$\phi'_{23}(t) = \frac{\frac{1}{2} + \frac{(t-t_{m-1})}{T}}{T}. \tag{2.62}$$

An additional verification can now be made for second-order BDFTS: when $t = t_m$, $\frac{d\bar{\mathbf{U}}(t)}{dt}$ should correspond to the BDF2 formula with the time step size equal to the period length T:

$$\left.\frac{d\bar{\mathbf{U}}(t)}{dt}\right|_{t=t_m} = \frac{3}{2T}\mathbf{U}^m - \frac{2}{T}\mathbf{U}^{m-1} + \frac{1}{2T}\mathbf{U}^{m-2} \tag{2.63}$$

which has the same form as equation (2.11) with $\Delta t = T$ as expected. Thus equation (2.59) gives the second-order BDFTS equations that are used in all BDFTS computations presented in this work.

2.3.4 Spectral-element Method in Time (SEMT)

For this method, the spectral element method is applied in the time dimension to find a periodic, quasi-periodic, or transient solution to the Euler Equations. Although this method is referred to as a "spectral method" in the literature, because if offers spectral-like p-convergence, it is in fact a finite-element method using Gauss-Lobatto quadrature points and a Lagrange polynomial basis [34]. First, the continuous time, trial solution for a single time element m is discretized using Nth degree Lagrange polynomials on $N+1$ quadrature points, as follows:

$$(V\mathbf{U})^{(m)}(\zeta) = \sum_{k=0}^{N} (V\mathbf{U})^{(m)}(\zeta_k)\psi_k^{(m)}(\zeta) \tag{2.64}$$

where $(V\mathbf{U})^{(m)}$ is the continuous time, trial function for element m, $\psi_k^{(m)}$ is the kth Lagrange polynomial of degree N in time element m given by the following:

$$\psi_k(\zeta) = \prod_{\eta=0,\ldots,N,\eta\neq k} \left(\frac{\zeta - \zeta_\eta}{\zeta_k - \zeta_\eta}\right). \tag{2.65}$$

ζ_k and ζ_η are the ζ locations of quadrature points k and η where $\zeta \in [-1, 1]$. The transformation from t to ζ and the inverse transform are $t = t_{m-1} + \frac{1}{2}(\zeta+1)T^{(m)}$ and $\zeta = \frac{2}{T^{(m)}}(t - t_{m-1}) - 1$

where, in this case, t_{m-1} is the time at which element m begins and $T^{(m)} = t_m - t_{m-1}$ is the length of element m with t_m being its rightward endpoint.

The Lobatto polynomials are the derivatives of the Legendre polynomials: $L_{o_i}(\zeta) = L'_{i+1}(\zeta)$ where the Legendre polynomials are defined as follows:

$$L_i(\zeta) = \frac{1}{2^i i!}\frac{d^i(\zeta^2-1)^i}{d\zeta^i}.$$

The Gauss-Lobatto quadrature points occur at the zeros of the completed Lobatto polynomials: $L^c_{o_{N+1}}(\zeta) = (1-\zeta^2)L_{o_{N-1}}$

To discretize the Euler equations using SEMT, one should begin with the general form of the Euler equations as give by equation (2.6). Substituting the trial solution given by equation (2.64) into (2.6) and minimizing the residuals using the Bubnov-Galerkin method [34, 69], the following is obtained for time element m:

$$\int_{-1}^{1} \nu(\zeta)\left[\frac{d(V\mathbf{U})^{(m)}}{d\zeta} + \frac{T^{(m)}}{2}\mathbf{R}(\mathbf{U}^{(m)}, \dot{\mathbf{x}}^{(m)}, \tilde{\mathbf{n}}^{(m)})\right] d\zeta = 0 \tag{2.66}$$

where $\nu(\zeta)$ is a weighting function taken as the pth Lagrange polynomial of degree N, $\psi_p^{(m)}$. Integrating by parts for $p = 0, ..., N$, the following is obtained for each element m:

$$(V\mathbf{U})^{(m)}\psi_p^{(m)}|_{-1}^{1} - \int_{-1}^{1}\left((V\mathbf{U})^{(m)}\frac{d\psi_p^{(m)}}{d\zeta} - \frac{T^{(m)}}{2}\mathbf{R}(\mathbf{U}^{(m)}, \dot{\mathbf{x}}^{(m)}, \tilde{\mathbf{n}}^{(m)})\psi_p^{(m)}\right) d\zeta = 0 \tag{2.67}$$

Exploiting the properties of the Lagrange polynomials, equation (2.67) is efficiently integrated over the Gauss-Lobatto quadrature points, including the end points, as follows:

$$\int_{-1}^{1} Q d\zeta = \sum_{p=0}^{N} Q(\zeta_p)\omega_p \tag{2.68}$$

where Q is any generic function of ζ and ω_p is the Gauss-Lobatto quadrature weight at node p. In matrix form for each element m, this becomes as follows:

$$\underline{\mathbf{\Psi}}^{(m)}\begin{bmatrix} V\mathbf{U}(\zeta_0) \\ \vdots \\ V\mathbf{U}(\zeta_N) \end{bmatrix}^{(m)} = \underline{\mathbf{I}}_{\omega}^{(m)}\begin{bmatrix} \mathbf{R}(\mathbf{U}(\zeta_0), \dot{\mathbf{x}}(\zeta_0), \tilde{\mathbf{n}}(\zeta_0)) \\ \vdots \\ \mathbf{R}(\mathbf{U}(\zeta_N), \dot{\mathbf{x}}(\zeta_N), \tilde{\mathbf{n}}(\zeta_N)) \end{bmatrix}^{(m)} \tag{2.69}$$

where

$$\underline{\boldsymbol{\Psi}}^{(m)} = \begin{bmatrix} \left(\frac{d\psi_0}{d\zeta}|_{\zeta_0}\omega_0 + 1\right) & \frac{d\psi_0}{d\zeta}|_{\zeta_1}\omega_1 & \dots & \frac{d\psi_0}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_0}{d\zeta}|_{\zeta_N}\omega_N \\ \frac{d\psi_1}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_1}{d\zeta}|_{\zeta_1}\omega_1 & \dots & \frac{d\psi_1}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_1}{d\zeta}|_{\zeta_N}\omega_N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_1}\omega_1 & \dots & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_N}\omega_N \\ \frac{d\psi_N}{d\zeta}|_{\zeta_0}\omega_0 & \frac{d\psi_N}{d\zeta}|_{\zeta_1}\omega_1 & \dots & \frac{d\psi_N}{d\zeta}|_{\zeta_{N-1}}\omega_{N-1} & \left(\frac{d\psi_N}{d\zeta}|_{\zeta_N}\omega_N - 1\right) \end{bmatrix}, \quad (2.70)$$

and

$$\underline{\mathbf{I}}_{\omega}^{(m)} = \frac{T^{(m)}}{2} \begin{bmatrix} \omega_0 & 0 & \dots & 0 & 0 \\ 0 & \omega_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \omega_{N-1} & 0 \\ 0 & 0 & \dots & 0 & \omega_N \end{bmatrix} \quad (2.71)$$

In equation (2.70) the first and last elements on the diagonal of $\underline{\boldsymbol{\Psi}}$ have an additional ± 1 which arises because of the first term in equation (2.67).

As was noted previously, the focus thus far has been on a single time element. These spectral elements in time can now be coupled in one of three different ways depending on the computational resources available and the type of the problem being solved.

Backward-only Coupling

The simplest way in which the spectral elements in time can be coupled is to treat each element as an initial value problem where the initial value for that element is the ending value of the previous element. In this way, the formulation resembles a first-order backward-difference formula, where the values in each spectral element are only dependent on the final value from the previous element in time. Past elements have no dependence on future elements.

For this type of coupling, the first row of the $\underline{\boldsymbol{\Psi}}^{(m)}$ matrix given in equation (2.70) is replaced with zeroes, except the first matrix element of the row, which is replaced with one

as follows:

$$\underline{\mathbf{\Psi}}^{(m)} = \begin{bmatrix} 1 & 0 & ... & 0 & 0 \\ \frac{d\psi_1}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_1}{d\zeta}|_{\zeta_1\omega_1} & ... & \frac{d\psi_1}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_1}{d\zeta}|_{\zeta_N\omega_N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_1\omega_1} & ... & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \frac{d\psi_{N-1}}{d\zeta}|_{\zeta_N\omega_N} \\ \frac{d\psi_N}{d\zeta}|_{\zeta_0\omega_0} & \frac{d\psi_N}{d\zeta}|_{\zeta_1\omega_1} & ... & \frac{d\psi_N}{d\zeta}|_{\zeta_{N-1}\omega_{N-1}} & \left(\frac{d\psi_N}{d\zeta}|_{\zeta_N\omega_N} - 1\right) \end{bmatrix}, \tag{2.72}$$

while the $\underline{\mathbf{I}}_{\omega}^{(m)}$ matrix remains unchanged. For each element, the value corresponding to $\zeta = -1$ is not computed, rather this value is supplied by the initial condition for the element as specified above. Thus, although the two matrices are of dimension $(N+1) \times (N+1)$, since the first row and column represent the coupling to the initial value, only N values, corresponding to all other values of ζ remain to be solved in each time element. This corresponds to one less than the number of quadrature points in the element and to the degree of the Lagrange polynomials used. Subsequent elements are then solved sequentially.

Monolithic Time

The next way in which the spectral elements in time can be coupled is the monolithic time approach. In this approach, all solution time is solved simultaneously and every time element is coupled both to its immediately previous and immediately subsequent time elements. The exception, of course, being the first and last elements which, are coupled to the initial condition and left uncoupled to the future, respectively. The monolithic time approach allows for solution in less wall-clock time by allowing the simultaneous use of more computing cores.

The inter-element connectivity or coupling is used to reduce the number of degrees of freedom for all elements except the beginning of the first and the end of the last elements as follows:

$$(V\mathbf{U})^{(m)}(\zeta_N) = (V\mathbf{U})^{(m+1)}(\zeta_0). \tag{2.73}$$

Additionally, the first element is modified into an initial value problem as shown in equation (2.72) above by replacing the first row with zeroes except the first matrix element which is replaced with 1. The end of the last element remains unmodified and uncoupled. For each element, N points in time must be solved. Given N_{ele} as the number of elements that

need to be solved in the monolithic slab of time, $N_{ele} \times N$ points in time must be solved simultaneously. In general, the assembled equation can be written in the following form:

$$\underline{\mathbf{\Psi}}_G \overrightarrow{VU} - \underline{\mathbf{\Omega}}_G \overrightarrow{R} = 0 \tag{2.74}$$

where $\overrightarrow{VU}$ is the global vector of cell-volume weighted conserved variables and $\overrightarrow{R}$ is the global vector of spatial residuals. The two matrices given in equation (2.74) above are the assembled time-coupling and quadrature-weight matrices, respectively, which have the forms as given in equations (2.75) and (2.76) on subsequent pages.

$$\underline{\mathbf{\Psi}}_G = \tag{2.75}$$

$$\begin{bmatrix}
1 & 0 & ... & 0 & 0 & ... & 0 & 0 & ... & ... & 0 & 0 & 0 & ... & 0 \\
\Psi^{(1)}_{1,0} & \Psi^{(1)}_{1,1} & ... & \Psi^{(1)}_{1,N} & 0 & ... & 0 & 0 & ... & ... & 0 & 0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\Psi^{(1)}_{N,0} & \Psi^{(1)}_{N,1} & ... & \Psi^{(1)}_{N,N} + \Psi^{(2)}_{0,0} & \Psi^{(2)}_{0,1} & ... & \Psi^{(2)}_{0,N} & 0 & ... & ... & 0 & 0 & 0 & .. & 0 \\
0 & 0 & ... & \Psi^{(2)}_{1,0} & \Psi^{(2)}_{1,1} & ... & \Psi^{(2)}_{1,N} & 0 & ... & ... & 0 & 0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & \Psi^{(2)}_{N,0} & \Psi^{(2)}_{N,1} & ... & \Psi^{(2)}_{N,N} + \Psi^{(3)}_{0,0} & \Psi^{(3)}_{0,1} & ... & ... & 0 & 0 & 0 & .. & 0 \\
0 & 0 & ... & 0 & 0 & ... & \Psi^{(3)}_{1,0} & \Psi^{(3)}_{1,1} & ... & ... & 0 & 0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & 0 & ... & 0 & 0 & ... & ... & \Psi^{(N_{ele}-1)}_{N-1,N-1} & \Psi^{(N_{ele}-1)}_{N-1,N} & 0 & ... & 0 \\
0 & 0 & ... & 0 & 0 & ... & 0 & 0 & ... & ... & \Psi^{(N_{ele}-1)}_{N,N-1} & \Psi^{(N_{ele}-1)}_{N,N} + \Psi^{(N_{ele})}_{0,0} & \Psi^{(N_{ele})}_{0,1} & ... & \Psi^{(N_{ele})}_{0,N} \\
0 & 0 & ... & 0 & 0 & ... & 0 & 0 & ... & ... & 0 & \Psi^{(N_{ele})}_{1,0} & \Psi^{(N_{ele})}_{1,1} & ... & \Psi^{(N_{ele})}_{1,N} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & 0 & ... & 0 & 0 & ... & ... & 0 & \Psi^{(N_{ele})}_{N,0} & \Psi^{(N_{ele})}_{N,1} & ... & \Psi^{(N_{ele})}_{N,N}
\end{bmatrix}$$

$$\underline{\mathbf{\Omega}}_G = \tag{2.76}$$

$$\frac{1}{2}\begin{bmatrix}
(T\omega_0)^{(1)} & 0 & ... & 0 & ... & 0 & 0 & ... & 0 \\
0 & (T\omega_1)^{(1)} & ... & 0 & ... & 0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & (T\omega_N)^{(1)} + (T\omega_0)^{(2)} & ... & 0 & 0 & .. & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & ... & (T\omega_N)^{(N_{ele}-1)} + (T\omega_0)^{(N_{ele})} & 0 & ... & 0 \\
0 & 0 & ... & 0 & ... & 0 & (T\omega_1)^{(N_{ele})} & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & ... & 0 & 0 & ... & (T\omega_N)^{(N_{ele})}
\end{bmatrix}$$

For practical purposes, the first row and column of both of these matrices can be eliminated and added to the equations as a source term representing the invariant initial condition, which is why only $N_{ele} \times N$ total points in time must be solved.

Periodic Coupling

The final approach by which the spectral elements in time can be coupled is a special case applicable only to purely-periodic problems. Enough elements are used to span exactly one period of variation in the problem. These elements are connected (coupled) on their interior ends just as was done for the monolithic time approach using equation (2.73). Additionally, to assign a periodic boundary condition in time, the left end of the first element is coupled to the right end of the final element as follows:

$$(V\mathbf{U})^{(1)}(\zeta_0) = (V\mathbf{U})^{(N_{ele})}(\zeta_N). \tag{2.77}$$

By nature of the periodic boundary condition, there are no real-time transients which must be dissipated before a purely periodic solution can be found; rather, all the transients are dissipated in pseudo-time, as they are for a steady-state solution. Indeed, a periodic solution of the SEMT Euler equations can be considered, like a purely-periodic time-spectral solution, as a system of coupled steady solutions that are solved simultaneously. Similarly to the monolithic-time approach, if N_{ele} elements are used to resolve a period, $N_{ele} \times N$ points in time must be solved simultaneously to resolve the periodic variation. For instance, if $N_{ele} = 16$ and $N = 10$, a total of 160 points in time are used.

For purely-periodic coupling, $\underline{\mathbf{\Psi}}_G$ and $\underline{\mathbf{\Omega}}_G$ are given as follows on the subsequent pages.

$$\underline{\mathbf{\Psi}}_G = \tag{2.78}$$

$$\begin{bmatrix}
\Psi_{0,0}^{(1)} + \Psi_{N,N}^{(N_{ele})} & \Psi_{0,1}^{(1)} & \dots & \Psi_{0,N}^{(1)} & 0 & \dots & 0 & 0 & \dots & 0 & \Psi_{N,0}^{(N_{ele})} & \Psi_{N,1}^{(N_{ele})} & \dots & \Psi_{N,N-1}^{(N_{ele})} \\
\Psi_{1,0}^{(1)} & \Psi_{1,1}^{(1)} & \dots & \Psi_{1,N}^{(1)} & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\Psi_{N,0}^{(1)} & \Psi_{N,1}^{(1)} & \dots & \Psi_{N,N}^{(1)} + \Psi_{0,0}^{(2)} & \Psi_{0,1}^{(2)} & \dots & \Psi_{0,N}^{(2)} & 0 & \dots & 0 & 0 & 0 & .. & 0 \\
0 & 0 & \dots & \Psi_{1,0}^{(2)} & \Psi_{1,1}^{(2)} & \dots & \Psi_{1,N}^{(2)} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & \Psi_{N,0}^{(2)} & \Psi_{N,1}^{(2)} & \dots & \Psi_{N,N}^{(2)} + \Psi_{0,0}^{(3)} & \Psi_{0,1}^{(3)} & \dots & 0 & 0 & 0 & .. & 0 \\
0 & 0 & \dots & 0 & 0 & \dots & \Psi_{1,0}^{(3)} & \Psi_{1,1}^{(3)} & \dots & 0 & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & \Psi_{N-1,N-1}^{(N_{ele}-1)} & \Psi_{N-1,N}^{(N_{ele}-1)} & 0 & \dots & 0 \\
\Psi_{0,N}^{(N_{ele})} & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & \Psi_{N,N-1}^{(N_{ele}-1)} & \Psi_{N,N}^{(N_{ele}-1)} + \Psi_{0,0}^{(N_{ele})} & \Psi_{0,1}^{(N_{ele})} & \dots & \Psi_{0,N-1}^{(N_{ele})} \\
\Psi_{1,N}^{(N_{ele})} & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & \Psi_{1,0}^{(N_{ele})} & \Psi_{1,1}^{(N_{ele})} & \dots & \Psi_{1,N-1}^{(N_{ele})} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\Psi_{N-1,N}^{(N_{ele})} & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & \Psi_{N-1,0}^{(N_{ele})} & \Psi_{N-1,1}^{(N_{ele})} & \dots & \Psi_{N-1,N-1}^{(N_{ele})}
\end{bmatrix}$$

$$\underline{\mathbf{\Omega}}_G = \tag{2.79}$$

$$\frac{1}{2}\begin{bmatrix}
(T\omega_0)^{(1)} + (T\omega_N)^{(N_{ele})} & 0 & ... & 0 & ... & 0 & 0 & ... & 0 \\
0 & (T\omega_1)^{(1)} & ... & 0 & ... & 0 & 0 & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & (T\omega_N)^{(1)} + (T\omega_0)^{(2)} & ... & 0 & 0 & .. & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & ... & (T\omega_N)^{(N_{ele}-1)} + (T\omega_0)^{(N_{ele})} & 0 & ... & 0 \\
0 & 0 & ... & 0 & ... & 0 & (T\omega_1)^{(N_{ele})} & ... & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & ... & 0 & ... & 0 & 0 & ... & (T\omega_{N-1})^{(N_{ele})}
\end{bmatrix}$$

Final Form of the SEMT Euler Equations

After the global time-coupling and quadrature-weight matrices are assembled, both terms in equation (2.74) are multiplied by the inverse of the global quadrature-weight matrix to put equation (2.74) into a similar form as the TS and BDFTS equations given by equations (2.38) and (2.59), i.e. all matrices multiply the conserved variables state vector and the spatial residual vector remains unchanged. After this multiplication, the TSEM Euler equations have the following form:

$$\underline{\mathbf{E}}\overrightarrow{VU} + \overrightarrow{R} = 0 \tag{2.80}$$

where $\underline{\mathbf{E}} = -\underline{\mathbf{\Omega}}_G^{-1}\underline{\mathbf{\Psi}}_G$.

If backward-only coupling is used, this same procedure is used for each time element individually, since each time element is solved separately and sequentially as its own initial value problem. No matter the coupling system used, the degrees of freedom in each element is one less than the number of quadrature points in that element because either the first point of the current element coincides with the last point of the previous element or because the first point corresponds to a prescribed initial value. Regardless of how the spectral elements in time are coupled, each element contains only N unknowns.

Chapter 3

Structural Equations

3.1 Two Degree-of-freedom Airfoil

The aeroelastic model is based on the response of an airfoil with two degrees of freedom, namely pitch and plunge as shown in Figure (3.1). The equations of motion for such a system can be summarized as follows:

$$\begin{aligned} m\ddot{h} + S_\alpha \ddot{\alpha} + K_h h &= -L \\ S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + K_\alpha \alpha &= M_{ea} \end{aligned} \tag{3.1}$$

where the newly introduced variables in these structural equations correspond to the following quantities:

m: mass of airfoil

S_α: static imbalance

I_α: sectional moment of inertia of airfoil

K_h: plunging spring coefficient

K_α: pitching spring coefficient

h: vertical displacement (positive downward)

α: angle-of-attack

L: sectional lift of airfoil

M_{ea}: sectional moment of airfoil about elastic axis.

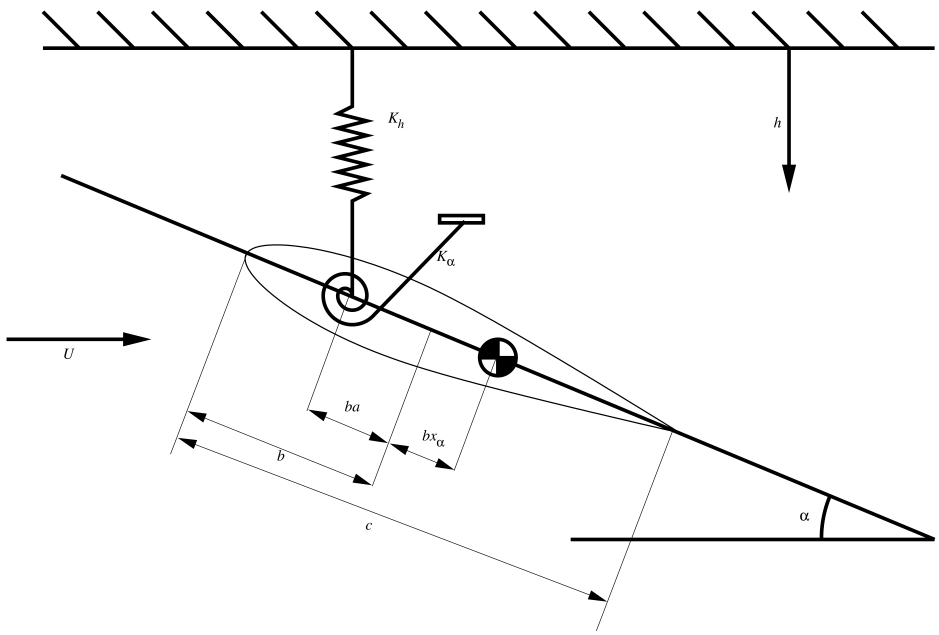


Figure 3.1: Two degree of freedom 2-D aeroelastic problem schematic

Non-dimensionalizing time in equations (3.1) with the uncoupled natural frequency of pitch yields:

$$\underline{\mathbf{M}}\ddot{\mathbf{q}} + \underline{\mathbf{K}}\mathbf{q} = \mathbf{F} \tag{3.2}$$

where the non-dimensional mass matrix $\underline{\mathbf{M}}$ and stiffness matrix $\underline{\mathbf{K}}$ are as follows:

$$\underline{\mathbf{M}} = \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix}, \quad \underline{\mathbf{K}} = \begin{bmatrix} \left(\frac{\omega_h}{\omega_\alpha}\right)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix}. \tag{3.3}$$

The corresponding non-dimensional load and displacement vectors are defined by the following relations:

$$\mathbf{F} = \frac{1}{\pi\mu k_c^2}\begin{bmatrix} -C_l \\ 2C_m \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \frac{h}{b} \\ \alpha \end{bmatrix}, \quad \ddot{\mathbf{q}} = \frac{\partial^2 \mathbf{q}}{\partial \check{t}^2}. \tag{3.4}$$

The newly introduced non-dimensionalized quantities are defined as given in the following table:

b: semichord of airfoil

k_c: reduced frequency of pitch, $k_c = \frac{\omega_\alpha c}{2U_\infty}$

μ: airfoil mass ratio, $\mu = \frac{m}{\pi\rho b^2}$

ω_h, ω_α: uncoupled natural frequencies of plunge and pitch

x_α: structural parameter defined as $\frac{S_\alpha}{mb}$

r_α^2: structural parameter defined as $\frac{I_\alpha}{mb}$

C_l, C_m: section lift coefficient and section moment coefficient about the elastic axis

$\breve{t}$: structural time, $\breve{t} = \omega_\alpha t$.

The reduced frequency k_c is typically written in terms of the flutter velocity V_f as follows:

$$k_c = \frac{\omega_\alpha c}{2U_\infty} = \frac{1}{V_f\sqrt{\mu}} \tag{3.5}$$

where c is the chord length of the airfoil, U_∞ is the freestream velocity, and the flutter velocity V_f is defined as follows:

$$V_f = \frac{U_\infty}{\omega_\alpha b\sqrt{\mu}}. \tag{3.6}$$

The natural frequency of pitch is found by solving equation (3.6) for ω_α in terms of the prescribed flutter velocity V_f.

3.1.1 Transformation of Structural Equations into First-order Form

The structural model equations as shown in Equation (3.2) are second-order partial differential equations. A transformation to first-order equations is used in order to facilitate the solution procedure. The transformation used is as follows:

$$\begin{aligned} \mathbf{r_1} &= \mathbf{q} \\ \mathbf{r_2} &= \dot{\mathbf{r}}_\mathbf{1}. \end{aligned} \tag{3.7}$$

The resulting first-order equations are then as follows:

$$\begin{aligned} \dot{\mathbf{r}}_\mathbf{1} &= \mathbf{r_2} \\ \dot{\mathbf{r}}_\mathbf{2} &= -\underline{\mathbf{M}}^{-1}\underline{\mathbf{K}}\mathbf{r_1} + \underline{\mathbf{M}}^{-1}\mathbf{F} \end{aligned} \tag{3.8}$$

which in matrix notation has the following form:

$$\begin{bmatrix} \dot{\breve{\mathbf{r}}}_1 \\ \dot{\breve{\mathbf{r}}}_2 \end{bmatrix} + \begin{bmatrix} 0 & -[I] \\ \underline{\mathbf{M}}^{-1}\underline{\mathbf{K}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \end{bmatrix} = \begin{bmatrix} 0 \\ [M]^{-1}\mathbf{F} \end{bmatrix} \tag{3.9}$$

$$\dot{\breve{\mathbf{r}}} \quad + \quad \underline{\boldsymbol{\Psi}}_S \mathbf{r} = \boldsymbol{\Phi}_S. \tag{3.10}$$

The matrix $\underline{\boldsymbol{\Psi}}_S$ is a constant for a given flutter velocity and can be precomputed and stored for use during the time-integration process. The time derivative term $\dot{\breve{\mathbf{r}}}$ can be discretized using second-order (or first-order) accurate backward difference formulae similar to the time derivative term in the flow equations, as follows:

$$\frac{3\mathbf{r}^m - 4\mathbf{r}^{m-1} + \mathbf{r}^{m-2}}{2\Delta\breve{t}} \quad + \quad \underline{\boldsymbol{\Psi}}_S \mathbf{r}^m = \boldsymbol{\Phi}_S^m \tag{3.11}$$

The periodic time-spectral method can also be used to discretize the time derivative term similar to equation (2.38):

$$\sum_{j=0}^{N-1} \breve{d}_n^j \mathbf{r}^j + \underline{\boldsymbol{\Psi}}_S \mathbf{r}^n = \boldsymbol{\Phi}_S^n \qquad n = 0, 1, 2, ..., N-1 \tag{3.12}$$

This term can also be discretized using BDFTS, similar to (2.59), as follows:

$$\begin{aligned} \sum_{j=1}^{N-1} \breve{d}_n^j \mathbf{U}^j - (\sum_{j=1}^{N-1} \breve{d}_n^j \phi_{23}(\breve{t}_j) - \phi'_{23}(\breve{t}_n))\mathbf{r}^m \\ -(\sum_{j=1}^{N-1} \breve{d}_n^j \phi_{22}(\breve{t}_j) - \phi'_{22}(\breve{t}_n))\mathbf{r}^{m-1} \\ -(\sum_{j=1}^{N-1} \breve{d}_n^j \phi_{21}(\breve{t}_j) - \phi'_{21}(\breve{t}_n))\mathbf{r}^{m-2} \\ +\underline{\boldsymbol{\Psi}}_S \mathbf{r}^n = \boldsymbol{\Phi}_S^n \qquad n = 1, 2, ..., N \end{aligned} \tag{3.13}$$

Finally, the time derivative term in the structural equations can also be discretized using SEMT similar to equation (2.69) as follows:

$$\underline{\boldsymbol{\Psi}}^{(m)} \begin{bmatrix} \mathbf{r}(\breve{\zeta}_0) \\ \vdots \\ \mathbf{r}(\breve{\zeta}_N) \end{bmatrix}^{(m)} = \underline{\mathbf{I}}_\omega^{(m)} \begin{bmatrix} \underline{\boldsymbol{\Psi}}_S \mathbf{r}(\breve{\zeta}_0) - \boldsymbol{\Phi}_S(\breve{\zeta}_0) \\ \vdots \\ \underline{\boldsymbol{\Psi}}_S \mathbf{r}(\breve{\zeta}_N) - \boldsymbol{\Phi}_S(\breve{\zeta}_N) \end{bmatrix}^{(m)} \tag{3.14}$$

where any of the three coupling methods can be used to couple the time elements as discussed in Chapter 2.

It should be noted that the time-step $\Delta\breve{t}$ appearing in the denominator of the discretized version of the structural equations is non-dimensionalized differently from the time-step in the flow equations, and their relation is $\Delta\breve{t} = \omega_\alpha \Delta t$, where Δt is the non-dimensional time-step used for the flow equations. This transformation arises because the structural equations have been non-dimensionalized by dividing through by the natural frequency of pitch, ω_α. Although the non-dimensionalization is different, the same "real" amount of time passes in a single period or time step for both the flow and structural equations. As a result, the period used to calculate the $\breve{d}_n^j$ and $\phi(\breve{t})$ in time-spectral and BDFTS is non-dimensionalized to the structural time frame. In the same way, structural time is also used in the formulation of the SEMT structural equations. A breve ($\breve{}$) is placed above all quantities that use structural time, for clarity.

Chapter 4

Solution Method

For all of the different time discretizations presented herein, the resulting system of equations is solved implicitly using a Newton method. Through the course of this research, it has been found that using the flexible variant of the Generalized Minimal Residual method (FGMRES) as the linear solver for each non-linear iteration provides efficient solution of all the time discretizations used. How efficiently FGMRES can solve the linear system depends strongly on the choice of preconditioner used therein. The present research has tended to indicate that preconditioners that utilize as complete a Jacobian matrix as possible outperform those that are more approximate in nature or neglect terms. The following sections describe the solver development.

4.1 Fully Implicit Method

A common approach for solving the system of Euler equations consists of adding a pseudo-time term as:

$$\frac{\partial}{\partial \tau}(V\mathbf{U}) + \frac{\partial}{\partial t}(V\mathbf{U}) + \mathbf{R}\left(\mathbf{U}(t), \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)\right) = 0 \tag{4.1}$$

where $\frac{\partial}{\partial \tau}(V\mathbf{U})$ is the pseudo-time derivative term which is discretized using a BDF1, and $\frac{\partial}{\partial t}(V\mathbf{U})$ is the real-time derivative term which can be discretized using any of the time approaches discussed previously. The system of equations (4.1) is pseudo-time stepped until a pseudo-time steady state is achieved for the given real-time discretization. For explicit

pseudo-time stepping approaches, it has been shown that the pseudo-time step is limited by stability considerations. In the case of the time-implicit BDF2 method, the following stability criterion applies [70, 71]:

$$\Delta\tau_n = CFL\frac{V^n}{\parallel \lambda \parallel + \frac{V^n}{\Delta t}} \tag{4.2}$$

where λ is the spectral radius of the spatial discretization operator $\mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n)$ and Δt is the physical-time step size. The second term in the denominator limits the pseudo-time step size to being smaller than the physical-time step size, i.e. $\Delta\tau_n < \Delta t$. Generally, $\parallel \lambda \parallel \gg \frac{V^n}{\Delta t}$ so the physical-time step consideration can be neglected. For time-spectral and BDFTS temporal discretizations, the pseudo-time step size is limited by the largest wave number that can be resolved, given the number of time instances used and the period length, as [23]:

$$\Delta\tau_n = CFL\frac{V^n}{\parallel \lambda \parallel + V^n k'} \tag{4.3}$$

where k' is that largest wave-number which can be resolved by the specified N time instances, as noted above and given as follows:

$$k' = \begin{cases} \frac{\pi N}{T} & \text{if } N \text{ is even} \\ \frac{\pi(N-1)}{T} & \text{if } N \text{ is odd} \end{cases} \tag{4.4}$$

where T in this case is the period length. This pseudo-time step restriction can also be considered in terms of the necessity that the pseudo-time step be smaller than the physical-time step, if, in this case, the physical-time step is considered to be the smallest spacing between time instances. Since time instances are spaced evenly throughout the period, this implies $\Delta\tau_n < \frac{T}{N}$ which is similar to the restriction above. In the case of SEMT, the pseudo-time step size is also limited, similarly to the above limits, as follows:

$$\Delta\tau_{(m)} = CFL\frac{V^{(m)}}{\parallel \lambda \parallel + \frac{V^{(m)}N^2}{T^{(m)}}} \tag{4.5}$$

where N is the degree of the Lagrange polynomial basis used and $T^{(m)}$ is the real-time length of the spectral element m, which is not necessarily the period length in the case of periodic SEMT. This restriction has not been formally derived, rather it is arrived upon by

analogy to the idea that $\Delta\tau_{(m)} < \Delta t$, where Δt is taken as the shortest physical-time distance between two quadrature points within the element (m). This restriction is summarized for $N \in \{1, ..., 12\}$ in Table (4.1), below.

Table 4.1: SEMT pseudo-time step restriction with physical-time step

$\mathbf{N}$	$\mathbf{\Delta t_{min}}$	$\frac{\mathbf{1}}{\mathbf{N}}$	$\frac{\mathbf{1}}{\mathbf{N^{3/2}}}$	$\frac{\mathbf{1}}{\mathbf{N^2}}$
1	1.0000	1.0000	1.0000	1.0000
2	0.5000	0.5000	0.3536	0.0250
3	0.2764	0.3333	0.1925	0.1111
4	0.1727	0.2500	0.1250	0.0625
5	0.1175	0.2000	0.0894	0.0400
6	0.0849	0.1667	0.0680	0.0278
7	0.0641	0.1429	0.0540	0.0204
8	0.0501	0.1250	0.0442	0.0156
9	0.0402	0.1111	0.0370	0.0123
10	0.0330	0.1000	0.0316	0.0100
11	0.0276	0.0909	0.0274	0.0083
12	0.0233	0.0833	0.0241	0.0069

As can be seen, for the values of N given in the table $\frac{1}{N^{3/2}}$ best approximates the smallest physical-time step; however, at $N = 12$, this approximation is already slightly less restrictive than what it should be. For this reason, and to make the code extensible to even higher-orders in time, the $\frac{1}{N^2}$ restriction has been used. The use of this restriction accords with the author's experience with the SEMT method: originally, a $\frac{1}{N}$ restriction was placed on the pseudo-time step as it was believed that the pseudo-time step should scale with the degree of the polynomial, which was thought to be analogous to the highest wave number in the time-spectral discretization. At higher orders and when the physical-time element size was small, however, the $\frac{1}{N}$ restriction was found to be unstable, and the $\frac{1}{N^2}$ pseudo-time step restriction was introduced to restore stability.

These restrictions on $\Delta\tau$ result in convergence degradation as the length of the time step, period, or time-element decreases or as N increases for TS, BDFTS, and SEMT. The impact of these pseudo-time step size restrictions can be greatly lessened by resorting to an implicit approach in pseudo-time. Such an approach has been derived in reference [72] using a first-order backwards-difference scheme in pseudo-time.

A more general strategy consists of devising a Newton approach for solving the fully-coupled non-linear equations at all time instances or points in time given by equation (4.1) with the chosen temporal discretization. The Newton scheme takes the form:

$$\underline{\mathbf{A}}\Delta\mathbf{U} = -\mathbf{R}_T\Big(V\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n, t\Big) \tag{4.6}$$

where $\mathbf{R}_T$ is the total residual including the contribution of the selected time discretization and the resulting Jacobian matrix is given as $\underline{\mathbf{A}} = \left[\frac{\partial R_T}{\partial \mathbf{U}}\right]$. For example, the Jacobian is given by the following for BDF2:

$$\underline{\mathbf{A}} = \frac{V^n}{\Delta\tau}\underline{\mathbf{I}} + \frac{3V^n}{2\Delta t}\underline{\mathbf{I}} + \underline{\mathbf{J}} \tag{4.7}$$

where the first term is a BDF1 pseudo-time term included to enhance diagonal dominance, the second term is the BDF2 real-time term, $\underline{\mathbf{I}}$ is the identity matrix of the appropriate size given the spatial discretization, and $\underline{\mathbf{J}}$ is the Jacobian of the spatial discretization alone: $\underline{\mathbf{J}} = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]$. The Jacobian for the TS time discretization is given as follows for N time instances [72]:

$$\underline{\mathbf{A}} = \begin{bmatrix} \frac{V^0}{\Delta\tau_0}\underline{\mathbf{I}} + \underline{\mathbf{J}}_0 & V^1 d_0^1 \underline{\mathbf{I}} & \dots & V^{N-1} d_0^{N-1}\underline{\mathbf{I}} \\ V^0 d_1^0 \underline{\mathbf{I}} & \frac{V^1}{\Delta\tau_1}\underline{\mathbf{I}} + \underline{\mathbf{J}}_1 & \dots & V^{N-1} d_1^{N-1}\underline{\mathbf{I}} \\ \vdots & \vdots & \ddots & \vdots \\ V^0 d_{N-1}^0 \underline{\mathbf{I}} & V^1 d_{N-1}^1 \underline{\mathbf{I}} & \dots & \frac{V^{N-1}}{\Delta\tau_{N-1}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{N-1} \end{bmatrix} \tag{4.8}$$

where a diagonal pseudo-time term can be included as shown for enhanced diagonal dominance of the Jacobian matrix. In the above matrix, $\underline{\mathbf{J}}_j$ corresponds to the Jacobian of the spatial discretization operator evaluated at time instance j.

The full Jacobian for the second-order BDFTS discretization has a similar form to the TS Jacobian; in fact, it is identical except for the first row which is replaced by the BDFTS coefficient as given by $B_{23,n} = -\Big(\sum_{j=1}^{N-1} d_n^j \phi_{23}(t_j) - \phi'_{23}(t_n)\Big)$ which is the nth component of the vector which couples each time instance to the Nth time instance at the end of the

period:

$$\underline{\mathbf{A}} = \begin{bmatrix} \frac{V^1}{\Delta\tau_1}\underline{\mathbf{I}} + \underline{\mathbf{J}}_1 & V^2 d_1^2 \underline{\mathbf{I}} & \dots & V^{N-1} d_1^{N-1} \underline{\mathbf{I}} & V^N B_{23,1} \underline{\mathbf{I}} \\ V^1 d_2^1 \underline{\mathbf{I}} & \frac{V^2}{\Delta\tau_2}\underline{\mathbf{I}} + \underline{\mathbf{J}}_2 & \dots & V^{N-1} d_2^{N-1} \underline{\mathbf{I}} & V^N B_{23,2} \underline{\mathbf{I}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V^1 d_{N-1}^1 \underline{\mathbf{I}} & V^2 d_{N-1}^2 \underline{\mathbf{I}} & \dots & \frac{V^{N-1}}{\Delta\tau_{N-1}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{N-1} & V^N B_{23,N-1} \underline{\mathbf{I}} \\ V^1 d_N^1 \underline{\mathbf{I}} & V^2 d_N^2 \underline{\mathbf{I}} & \dots & V^{N-1} d_N^{N-1} \underline{\mathbf{I}} & \frac{V^N}{\Delta\tau_N}\underline{\mathbf{I}} + V^N B_{23,N} + \underline{\mathbf{J}}_N \end{bmatrix}. \tag{4.9}$$

In a very similar way, the Jacobian for SEMT with either backward-only coupling or monolithic-time coupling is written as follows in terms of the $E_{j,k}$ which are the values in the jth row and kth column of the $\underline{\mathbf{E}}$ matrix that multiplies the conserved variable vectors in equation (2.80). As noted previously, the initial value can be incorporated as a source term, which is added as a constant term to the right hand side (residual) of the equation and thus removed from the matrix. For simplicity, the rows and columns of this matrix number $j, k \in [1, M]$ where $M = N$ for backward-only coupling and $M = N_{ele} \times N$ for the monolithic-time approach:

$$\underline{\mathbf{A}} = \begin{bmatrix} D_1 + \underline{\mathbf{J}}_1 & V^2 E_{1,2} \underline{\mathbf{I}} & \dots & V^{M-1} E_{1,M-1} \underline{\mathbf{I}} & V^M E_{1,M} \underline{\mathbf{I}} \\ V^1 E_{2,1} \underline{\mathbf{I}} & D_2 + \underline{\mathbf{J}}_2 & \dots & V^{M-1} E_{2,M-1} \underline{\mathbf{I}} & V^M E_{2,M} \underline{\mathbf{I}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V^1 E_{M-1,1} \underline{\mathbf{I}} & V^2 E_{M-1,2} \underline{\mathbf{I}} & \dots & D_{M-1} + \underline{\mathbf{J}}_{M-1} & V^M E_{M-1,M} \underline{\mathbf{I}} \\ V^1 E_{M,1} \underline{\mathbf{I}} & V^2 E_{M,2} \underline{\mathbf{I}} & \dots & V^{M-1} E_{M,M-1} \underline{\mathbf{I}} & D_M + \underline{\mathbf{J}}_M \end{bmatrix}. \tag{4.10}$$

where it was necessary to make the following substitution, $D_j = \left(\frac{1}{\Delta\tau_j} + E_{j,j}\right) V^j \underline{\mathbf{I}}$, to allow the Jacobian to fit in the required paper space.

For periodic SEMT, the Jacobian has a very similar form except the rows and columns of this matrix number $j, k \in [0, M-1]$ where $M = N_{ele} \times N$. This numbering arises because the contribution of what would be the Mth row and column is already included in the 0th row and column. Also, as such, there is no column representing an initial condition since

the problem is necessarily periodic.

$$\underline{\mathbf{A}} = \begin{bmatrix} D_0 + \underline{\mathbf{J}}_0 & V^1 E_{0,1} \underline{\mathbf{I}} & V^2 E_{0,2} \underline{\mathbf{I}} & \dots & V^{M-1} E_{0,M-1} \underline{\mathbf{I}} \\ V^0 E_{1,0} \underline{\mathbf{I}} & D_1 + \underline{\mathbf{J}}_1 & V^2 E_{1,2} \underline{\mathbf{I}} & \dots & V^{M-1} E_{1,M-1} \underline{\mathbf{I}} \\ V^0 E_{2,0} \underline{\mathbf{I}} & V^1 E_{2,1} \underline{\mathbf{I}} & D_2 + \underline{\mathbf{J}}_2 & \dots & V^{M-1} E_{2,M-1} \underline{\mathbf{I}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ V^0 E_{M-1,0} \underline{\mathbf{I}} & V^1 E_{M-1,1} \underline{\mathbf{I}} & V^2 E_{M-1,2} \underline{\mathbf{I}} & \dots & D_{M-1} + \underline{\mathbf{J}}_{M-1} \end{bmatrix} \tag{4.11}$$

4.1.1 Spatial Jacobian

For a first-order accurate spatial discretization, $\underline{\mathbf{J}}_{j,1stO}$ is as follows:

$$\underline{\mathbf{J}}_{j,1stO} = \frac{\partial \mathbf{R}_{1stO}}{\partial \mathbf{U}}. \tag{4.12}$$

The first-order accurate Jacobian couples each cell of the mesh to its nearest neighbors with the cell faces acting as interfaces between the cells. For a second-order spatial discretization, $\underline{\mathbf{J}}_{j,2ndO}$ contains many more entries, as each element of the mesh is not only influenced by its nearest neighbors, but also by the neighbors of its neighbors. The second-order Jacobian is derived as follows:

$$\underline{\mathbf{J}}_{j,2ndO} = \left.\frac{\partial \mathbf{R}_{2ndO}}{\partial \mathbf{U}}\right|_{L=constant} + \left.\frac{\partial \mathbf{R}_{2ndO}}{\partial \mathbf{L}}\right|_{U=constant} \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{U}}. \tag{4.13}$$

It should be readily apparent that the full Jacobian is of a size that could not be stored on modern computers for computational mesh sizes that are typically used. Instead, only the non-zero blocks of the Jacobian are stored. The first-order Jacobian is stored as an array of its diagonal blocks, which is (in 2D) $[4 \times 4 \times N_c]$ where N_c is the number of elements in the computational mesh, and an array of its off-diagonal blocks, which is stored on the faces and is $[2 \times 4 \times 4 \times N_f]$ where the 2 represents the opposite sides of each face and N_f is the number of faces. The full second-order Jacobian is not stored explicitly; rather, the evaluation of the exact Jacobian-vector products is enabled, as required, at each linear-solver iteration by storing three sets of diagonal and off-diagonal blocks, as shown in equation (4.13), which are then used to assemble Jacobian-vector products.

4.2 Linear Solver

Returning to equation (4.6), each non-linear Newton iteration requires solving the linear system given by the following:

$$\underline{\mathbf{A}}\Delta\mathbf{U} = -\mathbf{R}_T\Big(V\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n, t\Big) \tag{4.14}$$

where $\mathbf{R}_T$ represents the residual of the complete system, including the time-discretization terms, $\Big($i.e. right hand side of equation (4.6)$\Big)$, and $\underline{\mathbf{A}}$ is a large matrix spanning all spatial and temporal degrees of freedom. Since direct inversion of $\underline{\mathbf{A}}$ is generally intractable, an inexact Newton scheme can be formulated using an approximate representation of $\underline{\mathbf{A}}$ which is simpler to invert. One possible simplification is to replace the exact spatial Jacobian in each diagonal block $\underline{\mathbf{J}}_i$ by the corresponding first-order Jacobian $\underline{\mathbf{J}}_{i,1stO}$ as is often done for steady-state solvers. Another simplification consists of dropping all the off-diagonal terms representing the coupling between different points in time or time instances. When this is done, the linear system becomes decoupled in time and can be written as:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i = -\mathbf{R_T}(\mathbf{U}) \tag{4.15}$$

for each time instance $i = 0, 1, 2, ..., N-1$. Alternatively, the diagonal blocks in the $[\mathbf{A}]$ matrix may be retained and the system solved in a block Jacobi fashion following:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i^{l+1} = -\mathbf{R_T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l \tag{4.16}$$

where the block size corresponds to the entire spatial domain for each time instance and where l denotes the block Jacobi iteration index.

4.2.1 Block-colored Gauss-Seidel Linear Solver

As can be seen, both approaches require the inversion the first-order spatial Jacobian (augmented with a pseudo-time term) at each iteration. This may be accomplished using a suitable iterative solver such as block-colored Gauss-Seidel (BCGS). In this case, the block now corresponds to the 4×4 block diagonal matrix at each cell of the spatial discretization,

and the iterative scheme can be written as:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}} + \left[\underline{\mathbf{D}}_{i,1stO}\right]\right]\Delta\mathbf{U}_i^{l+1} = -\mathbf{R_T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l - \left[\underline{\mathbf{O}}_{i,1stO}\right]\Delta\mathbf{U}_i^l \tag{4.17}$$

where $\underline{\mathbf{D}}_{i,1stO}$ denotes the 4×4 block matrix for the current cell, $\underline{\mathbf{O}}_{i,1stO}$ refers to the off-diagonal blocks for neighboring mesh cells, and $\Delta\mathbf{U}_i^l$ are the spatial off-diagonal elements of the ith time instance. Although this equation describes a block Jacobi iteration (at the mesh-cell level), a block-colored Gauss-Seidel scheme can be recovered with a few simple modifications. First, the computational elements are divided into computational "colors" such that no two adjacent elements share the same color. This coloring allows the Gauss-Seidel method to be run in parallel. The BCGS method then updates (in parallel) all elements of each individual color (with the different colors updated in sequence), such that each update uses the newest information available for all other colors. The BCGS algorithm, written in residual form for a generic linear system is given in Algorithm (1).

Algorithm 1 : Block-colored Gauss-Seidel

```
Given A x = b
for i=1,...,ζ do
   for j=1,...,n_colors do
      for k=1,...,n_ej do
         Compute r_{i,k} = b − A x_current
         Compute Δx_{i,k} = D_k^{-1} r_{i,k}
         Update x_{i+1,k} = x_{i,k} + Δx_{i,k}
      end for
   end for
   Compute R_{L,i} = ||r_i||_2
   If satisfied Stop.
end for
```

In Algorithm (1), ζ is the maximum number of BCGS iterations allowed, n_{colors} is the number of colors into which the elements have been divided, and n_{e_j} is the number of elements having the jth color. Additionally, $\underline{\mathbf{D}}_k$ is the diagonal block of the Jacobian for element k, and is inverted directly using LU-decomposition. The specification $\mathbf{x}_{current}$ is used to indicate that, some colors will have information from the previous iteration $\mathbf{x}_i$ while other colors might have already been updated during the current iteration $\mathbf{x}_{i+1}$; in other words,

whether from the previous or current iteration, the most up-to-date information at the time of the evaluation of line 5 is used.

As noted earlier, the $\underline{\mathbf{A}}$ matrix given in lines 1 and 5 of the algorithm can either include or exclude the off-diagonal terms given the Jacobian for the chosen time discretization given in the previous section. When these time-coupling terms are excluded, this corresponds to a BCGS solver with explicit time treatment, which is abbreviated "BCGS-EX." When the off-diagonal terms are included, implicit time treatment is used in the BCGS solver, and this method is abbreviated "BCGS-IM."

4.2.2 Generalized Minimal Residual Method

Despite the additional stability the above, implicit method affords, the TS, BDFTS, and SEMT Euler equations still become difficult to solve as the number of time instances or points in time increases. With an increasing number of time instances, the Jacobians given in equations (4.8-4.11) proceed farther and farther from diagonal dominance. To restore diagonal dominance, a decreasingly small pseudo-time step must be used. Thus, the potential efficiency gains of the presented high-order time methods over time-implicit methods begin to evaporate.

To regain the efficiency improvements afforded by these methods, a linear solver that does not require diagonal dominance of the Jacobian should be used. The Generalized Minimum Residual method is such a solver. A welcome byproduct of this solver choice is that GMRES also more fully couples the various time instances. This additional coupling arises because the Hessenberg matrix that is directly inverted as part of GMRES is constructed using the full Jacobian of the linear system coupled over all time instances; whereas, the matrix that is directly inverted during each iteration of a stationary iterative method is some easily invertible part of the full Jacobian (usually the diagonal blocks). In other words, stationary iterative methods ignore some information (and thereby some coupling), but GMRES uses this information and preserves the full coupling across all time instances.

A flexible variant of the GMRES algorithm as described by Saad [31] is used. This flexible variant allows the use of an iterative method as preconditioner. The flexible GMRES

(FGMRES) algorithm proceeds as given in Algorithm (2).

Algorithm 2 : Flexible GMRES

1: Given $\underline{\mathbf{A}}\mathbf{x} = \mathbf{b}$
2: Compute $\mathbf{r}_0 = \mathbf{b} - \underline{\mathbf{A}}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
3: **for** j=1,. . . ,n **do**
4: Compute $\mathbf{z}_j := \underline{\mathbf{P}}^{-1}\mathbf{v}_j$
5: Compute $\mathbf{w} := \underline{\mathbf{A}}\mathbf{z}_j$
6: **for** i=1,. . . ,j **do**
7: $h_{i,j} := (\mathbf{w}, \mathbf{v}_i)$
8: $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_j$
9: **end for**
10: Compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
11: Define $\underline{\mathbf{Z}}_m := [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, $\underline{\bar{\mathbf{H}}}_m = \{h_{i,j}\}_{1\leq i\leq j+1; 1\leq j\leq m}$
12: **end for**
13: Compute $\mathbf{y}_m = argmin_y \|\beta\mathbf{e}_1 - \underline{\bar{\mathbf{H}}}_m\mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + \underline{\mathbf{Z}}_m\mathbf{y}_m$
14: If satisfied Stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and GoTo 1.

In this description, $\underline{\mathbf{A}}$ corresponds to the full time-discretized Jacobian matrix [$\mathbf{A}$] defined in equations (4.8-4.11), which may or may not be augmented with a pseudo time step, $\mathbf{b}$ corresponds to the negative of the non-linear total residual $-\mathbf{R_T}(\mathbf{U})$, and x is the non-linear update $\Delta\mathbf{U}$ to be computed.

Preconditioning is applied in line 4 of the algorithm. The BCGS linear solver is used for preconditioning in several different configurations as covered in the next subsection. A pseudo-time step must be applied to the BCGS system (i.e. added to $\underline{\mathbf{P}}$) to ensure diagonal dominance and convergence. To solve the minimization problem in line 13 of the algorithm, QR-factorization by means of Givens rotations is utilized [31]. Finally, line 14 of the algorithm indicates that this algorithm is, in fact, truncated, restarting GMRES using n Krylov vectors per restart.

Although Algorithm (2) shows the minimization problem outside the loop over Krylov vectors, in fact, this minimization is updated as each additional Krylov vector is added. This is done so that the current value of the linear residual is known for each iteration j. The FGMRES algorithm exits whenever this residual has either converged a specified amount or has converged to machine zero.

A pseudo-time step is used in the matrix $\underline{\mathbf{A}}$ in the FGMRES algorithm as well. By

using this pseudo-time step within FGMRES itself the routine is modified positively as follows: an update that will avoid over-correction issues is used; the pseudo-time step is allowed to grow as the residual decreases so quadratic convergence can be retained; and the convergence rate of the FGMRES linear solver itself is increased (i.e. fewer Krylov vectors and/or restarts are required to converge the linear system) because the pseudo-time step term makes the Jacobian better conditioned for Krylov subspace methods. Thus, two different pseudo-time steps are used: one in the second-order Jacobian $\underline{\mathbf{A}}$ used by the FGMRES algorithm itself and another in the first-order Jacobian $\underline{\mathbf{P}}$ used in the BCGS portion of the preconditioner. Because the Jacobian used in the BCGS preconditioner must be diagonally dominant, the preconditioner pseudo-time step is almost always smaller and grows more slowly than the FGMRES pseudo-time step. Conversely, the pseudo-time step used in the FGMRES algorithm grows rapidly such that the diagonal term becomes vanishingly small and an exact Newton method is recovered after several orders of magnitude decrease in the non-linear residual.

Preconditioning Methods for FGMRES

As mentioned above, all preconditioning methods make use of the block-colored Gauss-Siedel solver in one form or another. The first preconditioning method uses BCGS with explicit treatment of time. In other words, Algorithm (1) is used, with a specified number of iterations ζ on each Krylov vector. Here the preconditioner matrix $\underline{\mathbf{P}}$ corresponds to the $\underline{\mathbf{A}}$ matrix, given by equation (4.8-4.11), using the first-order spatial Jacobian with a pseudo-time step that is different, and smaller, than the pseudo-time step used in FGMRES. Additionally, this preconditioner matrix does not include the off-diagonal, time-spectral terms. GMRES using this preconditioner is abbreviated as "GMRES-EX" in the remainder of this work.

The next preconditioner used is BCGS with implicit treatment of time. This preconditioner uses Algorithm (1) with a specified number of iterations ζ on each Krylov vector. The full $\underline{\mathbf{A}}$ matrix, including the time-spectral coupling terms, is used, but again, the first-order spatial Jacobian is used with a smaller pseudo-time step. This solver/preconditioner combination is abbreviated "GMRES-IM".

Both of the preconditioners presented thus far have a serious limitation: the pseudo-time step size needed to preserve diagonal-dominance of the Jacobian in the BCGS algorithm remains relatively small. In order to bypass this limitation, a preconditioner for FGMRES is formed as a defect-correction method applied to the residual of equation (4.16) as:

$$\left[\frac{V^i}{\Delta\tau_{BCGS}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,1stO}\right](\Delta\mathbf{U}_i^{l+1} - \Delta\mathbf{U}_i^l) = \tag{4.18}$$
$$-\mathbf{R_T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l - \left[\frac{V^i}{\Delta\tau_{FGMRES}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_i\right]\Delta\mathbf{U}_i^l$$

where the right-hand-side corresponds to the residual of equation (4.16). A dual-iteration strategy is required for this preconditioner (thus the term defect-correction). Inner BCGS iterations are used to invert the left-hand side matrix providing an updated value for $\Delta\mathbf{U}_{i+1}^l$, which is then substituted into the right-hand side terms, and the process is repeated, effectively driving the right-hand-side residual to zero after a number of outer iterations. The Jacobian used on the RHS could be either the first- or second-order Jacobian given above; the specific Jacobian used will be noted in each case that discusses the solver in the rest of this work. The advantage of this approach comes from the fact that the pseudo-time step on the right-hand side residual is generally much larger than that required for stability of the BCGS iterative scheme. This solver/preconditioner combination is abbreviated "GMRES-DC".

Table 4.2 summarizes the various solvers and solver/preconditioner combinations presented in the current work and the abbreviations used to describe them.

Table 4.2: Summary of solver variants and preconditioners

Linear Solver	**Preconditioner**	**Abbreviation**
Time-explicit BCGS	N/A	BCGS-EX
Time-implicit BCGS	N/A	BCGS-IM
FGMRES	Time-explicit BCGS	GMRES-EX
FGMRES	Time-implicit BCGS	GMRES-IM
FGMRES	Defect Correction with Time-implicit BCGS	GMRES-DC

Aeroelastic Contributions

For aeroelastic problems, the fluid and structure equations are coupled together implicitly; thus, for each iteration, fluid and structure residuals are found using either the BDF2, TS/BDFTS, or SEMT equations as discussed previously. Then, the Jacobian is found for the entire fluid/structure system. This Jacobian not only couples the fluid residuals to the flow variables and the structural residuals to the structural variables, but also the fluid residuals to the structural variables and the structural residuals to the flow variables. Starting with the flow Jacobian (of a single time instance in the case of the TS, BDFTS and SEMT), the coupled Jacobian will have as many additional rows and columns as there are structural variables. Thus, a flow Jacobian that is of size $\Big[(4 \times N_c) \times (4 \times N_c)\Big]$ where N_c is the number of elements or cells in the computational domain of the flow, would become a $\Big[(4 \times N_c + N_s) \times (4 \times N_c + N_s)\Big]$ matrix for the implicitly coupled fluid/structure system, where N_s is the number of structural variables.

While the above represents the size of the Jacobian if all entries were stored, it should be readily apparent that a matrix of this size could not be stored on modern computers for the sizes of computational meshes that are typically used. Instead, only the non-zero blocks of the Jacobian are stored. The first-order Jacobian is stored as an array of its diagonal blocks, which is $[4 \times 4 \times N_c]$, and an array of its off-diagonal blocks, which are stored on the faces and is $[2 \times 4 \times 4 \times N_f]$ where the 2 represents the opposite sides of each face and N_f is the number of faces. The full second-order Jacobian is not stored explicitly; rather, the evaluation of exact Jacobian-vector products is enabled as required at each linear solver iteration by storing three sets of diagonal and off-diagonal blocks as described by equation (4.13), which are then used to assemble Jacobian-vector products.

The changes in the structural residuals with respect to the flow variables occur entirely as a function of the lift and moment coefficient variations with the flow variables, thus:

$$\frac{\partial \mathbf{R_s}}{\partial \mathbf{U}} = f\left(\frac{\partial C_L}{\partial \mathbf{U}}, \frac{\partial C_m}{\partial \mathbf{U}}\right) \tag{4.19}$$

where $\mathbf{R_s}$ are the residuals of the structural equations using the chosen time discretization. This portion of the Jacobian $\frac{\partial \mathbf{R_s}}{\partial \mathbf{U}}$, above, has non-zero values only in cells which include a

face that lies on the surface of the airfoil and only these non-zero values are stored.

The changes in the flow residuals with respect to the structural variables are a function only of the change in the grid motion in response to variations in the structural variables. That is, the structural variables influence both the grid speeds and the orientation of the face normal vectors:

$$\frac{\partial \mathbf{R_f}}{\partial \mathbf{r}} = f\left(\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{r}}, \frac{\partial \tilde{\mathbf{n}}}{\partial \mathbf{r}}\right) \tag{4.20}$$

where $\mathbf{R_f}$ are the residuals of the flow equations using the chosen time discretization. In the present work, mesh motion is achieved by translating and rotating the mesh rigidly. Thus, every face in the mesh changes speed and orientation when the mesh is rotated and translated. Because every face is affected, $\frac{\partial \mathbf{R_f}}{\partial \mathbf{r}}$ has the same size as the diagonal blocks of the first-order Jacobian, $[4 \times 4 \times N_c]$. For three-dimensional, fixed-wing or deforming-blade aeroelastic analysis, where rigid mesh motion is not feasible, a deforming mesh capability would be required. This deforming mesh motion will add additional dependencies on cell volume and face area in the above portion of the coupled Jacobian. Additionally, since all face-normals change orientation when a rigid mesh rotates, the above component of the Jacobian has non-zero values for all cells. This would not necessarily be the case for a deforming mesh. It should also be noted that an additional dependence of the structural residuals on the structural variables is included in this framework. This dependence arises as a result of the grid motion since the coefficient of lift is itself dependent on the orientation of the face normals on the airfoil surface:

$$\frac{\partial \mathbf{R_s}}{\partial \mathbf{r}} = f\left([\Psi], \frac{\partial C_L}{\partial \tilde{\mathbf{n}}} \frac{\partial \tilde{\mathbf{n}}}{\partial \mathbf{r}}\right) \tag{4.21}$$

The preceding portion of the full fluid/structure Jacobian retains its $[4 \times 4]$ size.

The fully-coupled linear system described in the preceding paragraphs and equations is solved for each non-linear iteration using the GMRES-DC solver, as described above, but with the following modifications. First, both the first-order Jacobian used in the BCGS preconditioner and the second-order Jacobian used in both the Defect-correction step and in the FGMRES algorithm contain the full fluid/structure coupling given by $\partial \mathbf{R_f}/\partial \mathbf{U}$, $\partial \mathbf{R_s}/\partial \mathbf{U}$,

$\partial \mathbf{R_f}/\partial \mathbf{r}$, and $\partial \mathbf{R_s}/\partial \mathbf{r}$ as described above. These additions are included in both levels of preconditioning. It has been found that for more difficult aeroelastic problems, convergence could stagnate if structural coupling was neglected from the preconditioning, as had been done in previous work [33]. It is hypothesized that this convergence stagnation occurs primarily because the FGMRES solver will exit when only the flow portion of the residual has been reduced (because of preconditioning) and the structural residual has either increased or not decreased at all. This will continue until the structural residual becomes significantly larger than the flow residual, such that the structural residual dominates the system, and is itself updated. When this structural update occurs, however, the new orientation of the airfoil leads to an increased flow residual. This pattern repeats as the flow and structural residuals alternately increase and decrease. By including the fluid/structure coupling in the preconditioner, this problem has been eliminated.

Second, the defect-correction step described for the flow-alone solver has, in general, used the first-order flow Jacobian; however, for aeroelastic cases, it was found that use of the second-order Jacobian in the defect-correction step proved slightly more efficient. As such, equation (4.18) has been modified to use the second-order flow Jacobian on the right-hand side, as follows:

$$\left[\frac{V^i}{\Delta\tau_{BCGS}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,1stO}\right](\Delta\mathbf{U}_i^{l+1} - \Delta\mathbf{U}_i^l) = \tag{4.22}$$
$$-\mathbf{R_T}(\mathbf{U}) - \sum_{j\neq i}\left[V^j d_i^j \underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l - \left[\frac{V^i}{\Delta\tau_{FGMRES}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{i,2ndO}\right]\Delta\mathbf{U}_i^l$$

Finally, just as is the case with the flow Jacobian, a pseudo-time term is added to the diagonal-block of the structural contribution to the Jacobian. This pseudo-time term has the same form as that of the flow, given by equations (4.2-4.5); however, the value for the spectral radius λ that couples the fluid and structure together is not found analytically. Through trial and error, a value of $\lambda = 6.0$ is used for all cases presented herein. This value was found to allow for the efficient solution of all the cases considered, but is most likely specific to the structural parameters used for these test cases. Future work should include the derivation of an analytic value for the spectral radius of the structural contribution.

Chapter 5

Other Details of Implementation

This chapter includes all the remaining theory and implementation that have been used in the thesis but are neither its focus nor described elsewhere.

5.1 Gust Prescription by the Split Velocity Method

Second-order unstructured, unadapted CFD methods, with their greatly varying cell sizes, quickly dissipate vortices and gusts. Usually, this is not a problem as the only vortex of importance is the one shed by the airfoil itself, which need be preserved only as long as it is affecting the flow around that same airfoil. In the cases herein, however, the preservation of a gust that is prescribed very far forward of the airfoil is paramount. One way to preserve such a gust would be to have a uniformly refined mesh where all cells are very small or one could adapt only the cells that the gust occupies at each time step, and then de-refine them once the gust has moved on. Both these solutions, however, greatly increase computational cost. Instead of refinement, gusts in the present work are prescribed using the Split Velocity Method (SVM) [73–77].

To derive the SVM, the Euler equations as given in equation (2.1) are written explicitly in terms of the conserved variables as:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho E+p)u\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho E+p)v\end{bmatrix}=0 \tag{5.1}$$

and the ideal gas equation of state is given as:

$$p=(\gamma-1)\rho\Big[E-\frac{1}{2}(u^2+v^2)\Big]. \tag{5.2}$$

In the SVM, the velocity and energy are decomposed into flow and disturbance components as follows:

$$\begin{aligned} u&=\bar{u}+u_d\\ v&=\bar{v}+v_d\\ E&=\bar{E}+E_d \end{aligned} \tag{5.3}$$

where $\bar{u}$, $\bar{v}$, and $\bar{E}$ are the flow components and u_d, v_d, and E_d are the prescribed disturbance components, in this case, the gust components. The energy component itself can be found from the equation of state:

$$E=\frac{p}{\rho(\gamma-1)}+\frac{1}{2}(u^2+v^2)=\left\{\frac{p}{\rho(\gamma-1)}+\frac{1}{2}(\bar{u}^2+\bar{v}^2)\right\}+\left\{(\bar{u}u_d+\bar{v}v_d)+\frac{1}{2}(u_d^2+v_d^2)\right\} \tag{5.4}$$

where the first component in braces is $\bar{E}$ and the rightmost component in braces is E_d. It should be noted that density and pressure remain unchanged by the prescription of the disturbance quantities. Now, the decomposed components given by equation (5.3) are substituted into the Euler equations (5.1):

$$\begin{aligned}\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho(\bar{u}+u_d)\\ \rho(\bar{v}+v_d)\\ \rho(\bar{E}+E_d)\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho(\bar{u}+u_d)\\ \rho(\bar{u}+u_d)^2+p\\ \rho(\bar{u}+u_d)(\bar{v}+v_d)\\ (\rho(\bar{E}+E_d)+p)(\bar{u}+u_d)\end{bmatrix}\\ +\frac{\partial}{\partial y}\begin{bmatrix}\rho(\bar{v}+v_d)\\ \rho(\bar{u}+u_d)(\bar{v}+v_d)\\ \rho(\bar{v}+v_d)^2+p\\ (\rho(\bar{E}+E_d)+p)(\bar{v}+v_d)\end{bmatrix}=0.\end{aligned} \tag{5.5}$$

All the disturbance components are then isolated as follows:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\bar{u}\\ \rho\bar{v}\\ \rho\bar{E}\end{bmatrix}+\frac{\partial}{\partial x}\left\{\begin{bmatrix}\rho\bar{u}\\ \rho\bar{u}^2+p\\ \rho\bar{u}\bar{v}\\ (\rho\bar{E}+p)\bar{u}\end{bmatrix}+\begin{bmatrix}\rho u_d\\ \rho\bar{u}u_d\\ \rho\bar{v}u_d\\ \rho\bar{E}u_d\end{bmatrix}\right\} \tag{5.6}$$
$$+\frac{\partial}{\partial y}\left\{\begin{bmatrix}\rho\bar{v}\\ \rho\bar{u}\bar{v}\\ \rho\bar{v}^2+p\\ (\rho\bar{E}+p)\bar{v}\end{bmatrix}+\begin{bmatrix}\rho v_d\\ \rho\bar{u}v_d\\ \rho\bar{v}v_d\\ \rho\bar{E}v_d\end{bmatrix}\right\}+\begin{bmatrix}0\\ s_m(u_d)\\ s_m(v_d)\\ s_e(u_d,v_d)\end{bmatrix}=0$$

where all of the omitted terms are contained in the source terms in the rightmost vector on the left hand side. These source terms are as follows:

$$\begin{aligned}s_m(\xi)&=\rho\left\{\tfrac{\partial\xi}{\partial t}+(\bar{u}+u_d)\tfrac{\partial\xi}{\partial x}+(\bar{v}+v_d)\tfrac{\partial\xi}{\partial y}\right\}\\ s_e(u_d,v_d)&=\bar{u}s_m(u_d)+\bar{u}s_m(v_d)+p(\tfrac{\partial u_d}{\partial x}+\tfrac{\partial v_d}{\partial v})\end{aligned} \tag{5.7}$$

where ξ is either u_d or v_d and the pressure is found using only the flow components as follows:

$$p=(\gamma-1)\rho(\bar{E}-\frac{1}{2}(\bar{u}^2+\bar{v}^2)). \tag{5.8}$$

When these equations are considered in their entirety, the disturbance component of energy E_d naturally vanishes. If the split-velocity Euler equations are transformed into integral form and compared to equation (2.5), it becomes apparent that the SVM equations correspond to the Euler equations in arbitrary-Lagrangian-Eulerian (ALE) form with the addition of the volume-integrated source terms and an additional grid-velocity component as follows:

$$\dot{\mathbf{x}}=-\begin{bmatrix}u_d\\ v_d\end{bmatrix}. \tag{5.9}$$

If the source terms are neglected, the SVM corresponds to the Field Velocity Method [73], which has been used for disturbance calculations in the past by other authors and neglects the effect of the presence of the airfoil on the disturbance itself, i.e. the flow does not change as the gust interacts with the airfoil in its path. The velocities calculated as the flow variables are $\bar{u}$ and $\bar{v}$, so the disturbance velocity must be added to the flow output in order

to reflect the correct total velocity. The density and pressure calculated, however, are the true density and pressure. The final energy can be found by using equation (5.4) including the disturbance velocity contributions. Finally, it should be noted that no approximation was introduced when the Euler equations are rearranged into the SVM form. Equations (5.6) exactly contain all of the same contributions as equations (5.1).

5.1.1 Prescribed Gusts

All of the gusts prescribed herein are $1-cosine$ vertical gusts. The disturbance velocity for such gusts is prescribed by the following:

$$v_d = \begin{cases} 0 & when \quad x' < 0 \quad x' > l_g \\ \frac{v_{g,max}}{2}\left[1-\cos\frac{2\pi x'}{l_g}\right] & when \quad 0 \leq x' \leq l_g \end{cases} \tag{5.10}$$

where

$$x' = -x + U_\infty t + x_0 \tag{5.11}$$

where x_0 is the initial position of the leading edge of the gust. This formula shows that $x' = 0$ is the leading edge of the gust and $x' = l_g$ is the trailing edge of the gust at time t.

5.2 Implementation

The various time instances in the time-spectral method and quadrature points in time within a time element in the spectral-element method in time are coupled and must be solved simultaneously. This coupling can be implemented serially, whereby a single point in time is solved at any given moment, which then transmits its update to the next time instance, which in turn is solved, and the process repeated sequentially until all points in time have been updated. However, since the coupling occurs as a source term in the residual (on the right hand side of the equation), each individual time instance may be solved in parallel with the other time instances. This introduces an additional dimension for achieving parallelism compared to time-implicit computations, where progress in the time dimension is necessarily sequential. In the implementation in the present work, two levels of parallelism

are introduced, the first in the spatial dimension, and the second in the time dimension where the various time instances are solved by spawning multiple instances of the spatial solver on a parallel computing cluster. The implementation uses MPI for parallelism in the time dimension and OpenMP for additional parallelism in the space dimension. This additional parallelism in time may prove to be particularly enabling with the advent of rapidly expanding massively parallel computing clusters, particularly for cases where parallelism in the spatial dimension has been exhausted (perhaps due to the adequacy of moderate grid sizes).

One of the drawbacks of these alternative methods for the discretization of time is that each time instance must broadcast its entire solution field to all other points in time (within a given time element in the case of the SEMT), which can result in a significant amount of communication. Various strategies for communicating the different time instances to all processors have been investigated. Currently, a round-robin approach is implemented, where each processor sends its time instance to a single neighboring processor. The received time instance is added to the time derivative source term on the local processor, and then passed on to the next processor. By repeating this procedure $N - 1$ times, where N is the number of time instances, the complete time derivative involving summations from all points in time is accumulated without the requirement of creating a local temporary copy of all the additional time-instance solution vectors or performing any communication intensive broadcast operations.

Chapter 6

Time-spectral Flow Results

6.1 Test Case 1: Pitching Motion in Two Harmonics

A two-dimensional inviscid flow test case is constructed with the forced, pitching oscillation of a NACA-0012 airfoil at two reduced frequencies $k_{c_1} = 0.02$ and $k_{c_2} = 0.14$. It should be noted that the second frequency is seven times the first; in other words, motion occurs in the first and seventh harmonics. This case is run at a free stream Mach number $M_\infty = 0.50$ and pitches about the airfoil quarter chord. This motion is given by the following equation:

$$\alpha(t) = \alpha_0 + \alpha_1 \sin(\omega t) + \alpha_7 \sin(7\omega t). \tag{6.1}$$

The amplitude of pitching in the first harmonic is $\alpha_1 = 2.5^o$ and $\alpha_7 = 1.0^o$ in the seventh harmonic; additionally, $\alpha_0 = 0.0^o$. The angular frequency is related to the reduced frequency as follows:

$$\omega = \frac{2U_\infty k_c}{c} \tag{6.2}$$

where U_∞ is the freestream velocity and c is the airfoil chord length. The unstructured, computational mesh consists of 8,747 triangles. Figure (6.1) shows a near field view of the mesh. It should be noted that the $L_2 - norm$ of the non-linear residual is converged to be less than a tolerance of 1×10^{-11} for each time step in the time-implicit method and for every time instance in the time-spectral method. Figure (6.2) plots the lift and moment coefficients as functions of non-dimensional time for a reference, time-implicit solution which uses 512

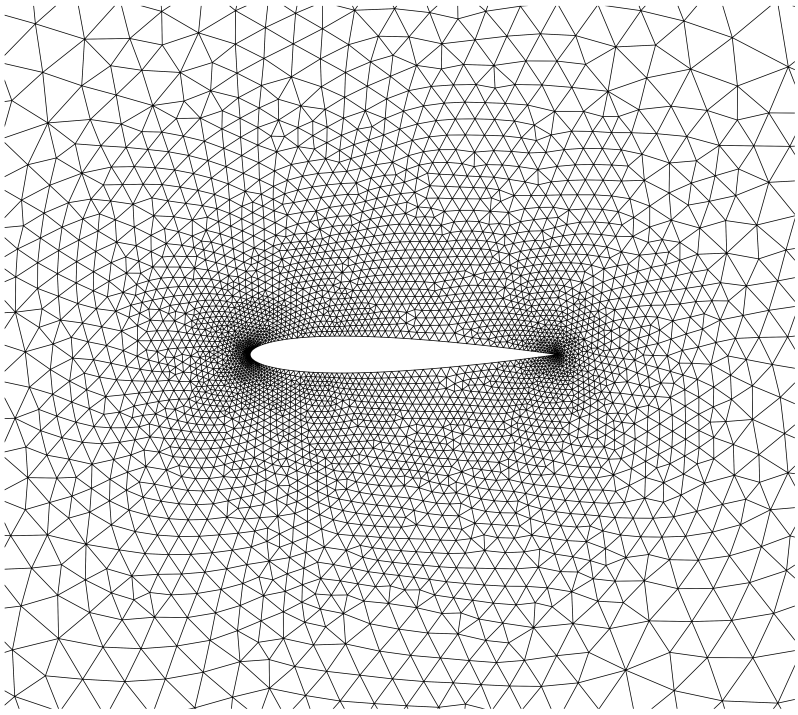
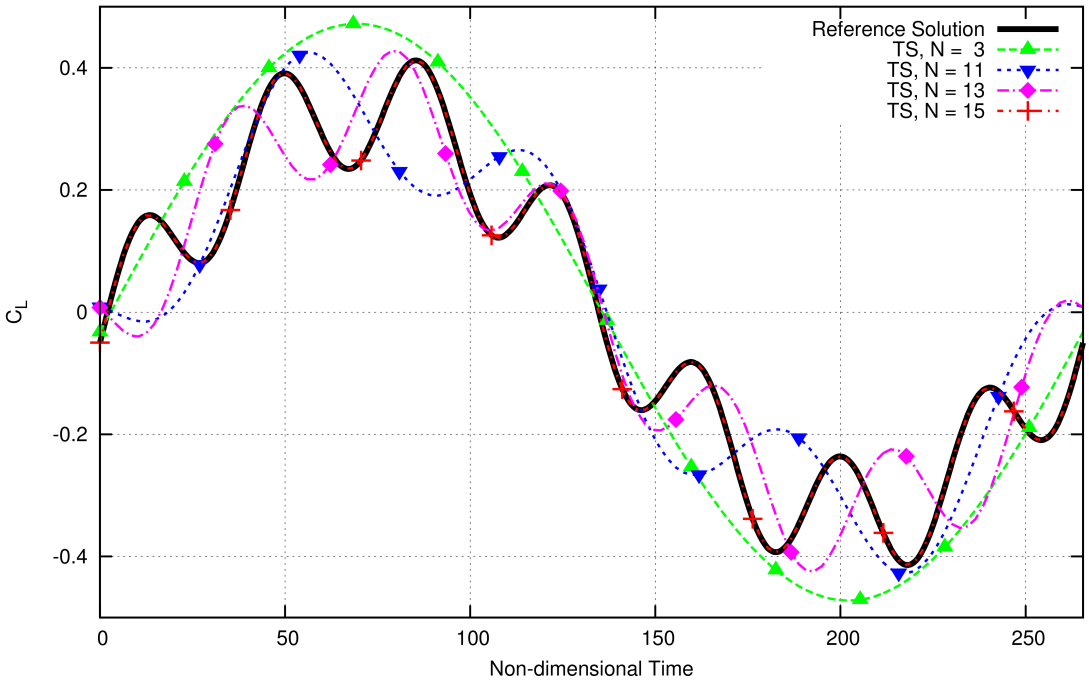


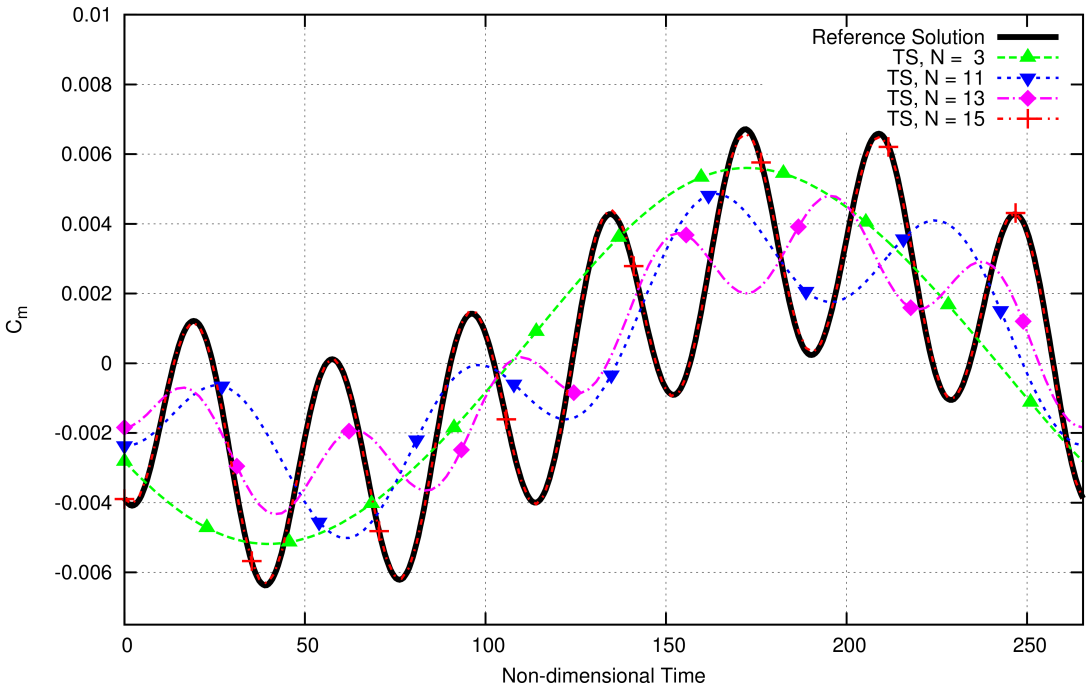
Figure 6.1: Near field mesh for the NACA-0012 airfoil

time steps per period and time-spectral solutions using $N = 3, 11, 13$, and 15 time instances. As can be seen, $N = 15$ time instances matches the reference time-implicit solution quite well for both lift and moment coefficients. This is the expected result as the number of time instances required is $N = H \times 2 + 1$ where H is the highest harmonic of content that can be resolved by these N time instances. Since lift is primarily a function of angle-of-attack, and the 7th harmonic is the highest harmonic of pitching motion, 15 time instances can accurately resolve the lift for this low Mach number test case. In this case, the moment coefficient is primarily a function of lift; thus 15 time instances can also resolve the moment coefficient.

Figure (6.3) plots the convergence history of the non-linear residual versus both the non-linear iteration count and wall-clock time for $N = 3, 7, 11$, and 15 time instances obtained using the GMRES-IM solver, which was the best performing solver for most numbers of time instances for this test case. As can be seen, all cases converge very smoothly. Additionally, this case converges along very similar non-linear iteration paths for all four time instance

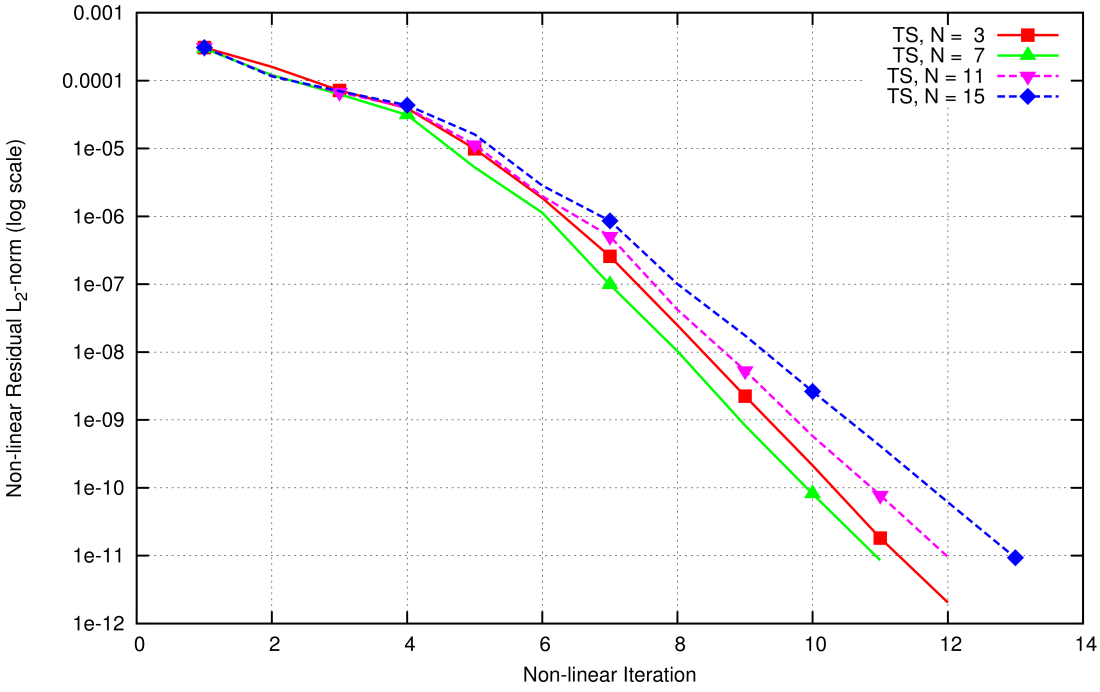


(a)

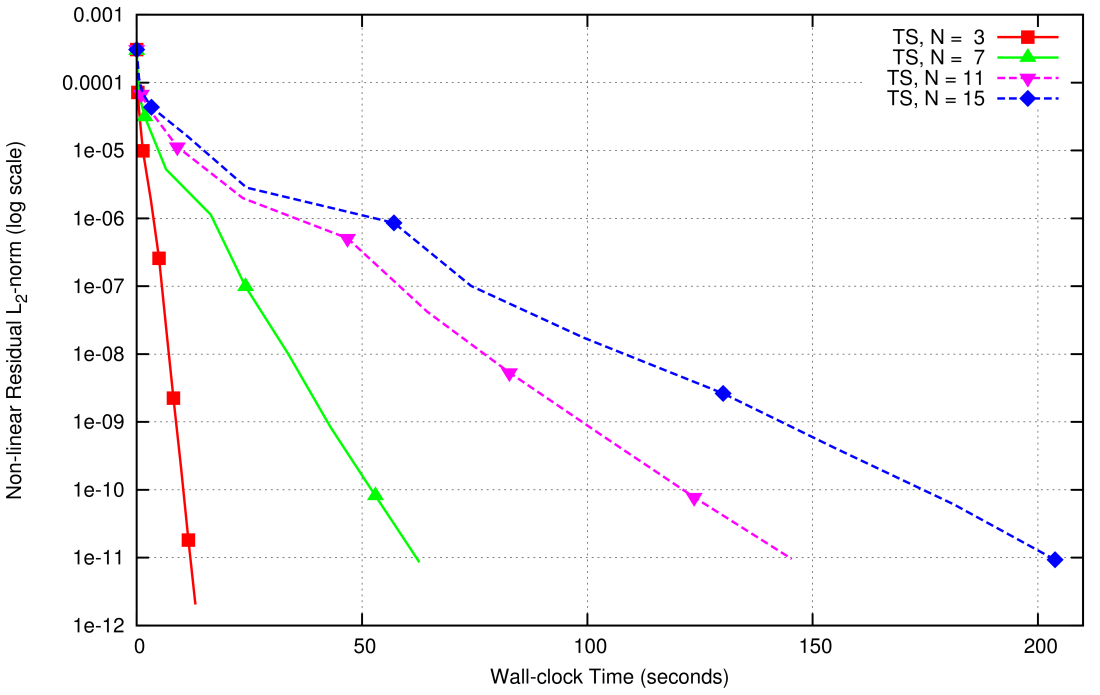


(b)

Figure 6.2: Comparison of computed lift coefficient (a) and moment coefficient (b) using the time-spectral method to a reference, time-implicit solution for aerodynamic test case 1



(a)



(b)

Figure 6.3: Comparison of convergence of aerodynamic test case 1 using different number of time-spectral time instances with non-linear iterations (a) and wall-clock time (b) on the x-axis

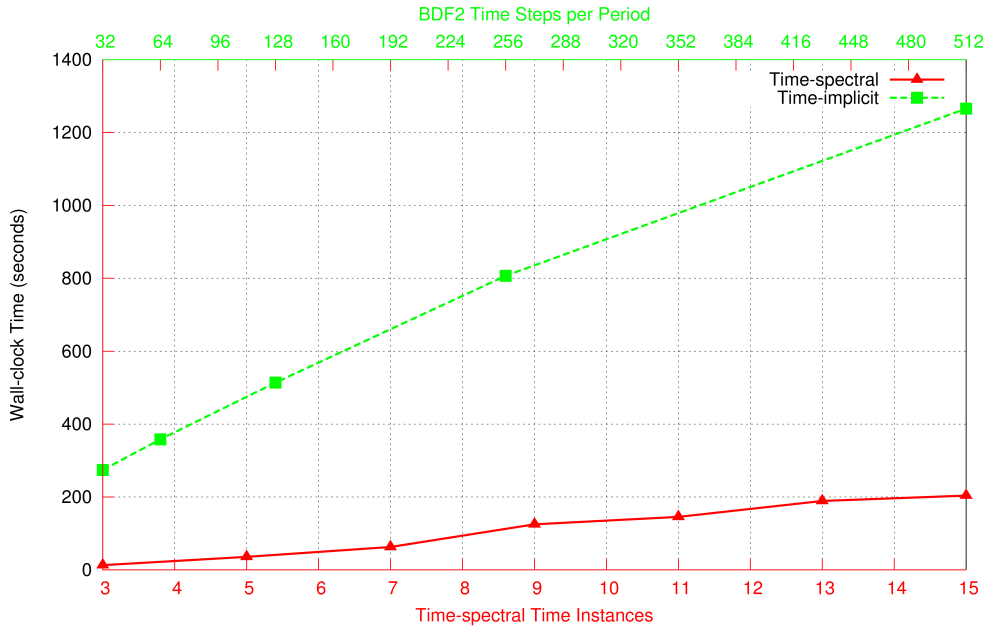


Figure 6.4: Comparison of wall-clock time to convergence for time-implicit and time-spectral methods with different numbers of time steps per period (top axis) and time instances (bottom axis) for aerodynamic test case 1

numbers shown. Wall-clock time convergence scales along expected lines since all cases were run on the same hardware. Finally, Figure (6.4) plots the wall-clock time needed to converge this test case for $32, 64, 128, 256$, and 512 time steps per period (top axis) using the time-implicit method and for $3, 5, 7, 9, 11, 13$, and 15 time instances (bottom axis) using the time-spectral method. As can be seen, the time-spectral method, even using the maximum of $N = 15$ time instances, converges more quickly than the time-implicit method in all cases. Both the BDF2 and TS solutions are run on a single 16-core cluster node (where BDF2 uses all 16 cores by utilizing the shared memory OpenMP and TS uses both MPI and OpenMP to use as many cores as possible give the number of time instances), with the BDF2 solutions each having been run for five periods in order to achieve periodic solution behavior.

6.1.1 Performance Comparison of Solver Variants

Figure (6.5) plots the convergence of the time-spectral method for the this test case using $N = 3$ time instances for the five different solver configurations summarized in Table (4.2). In order to resolve any details for the GMRES-based solvers, it was necessary to scale the x-

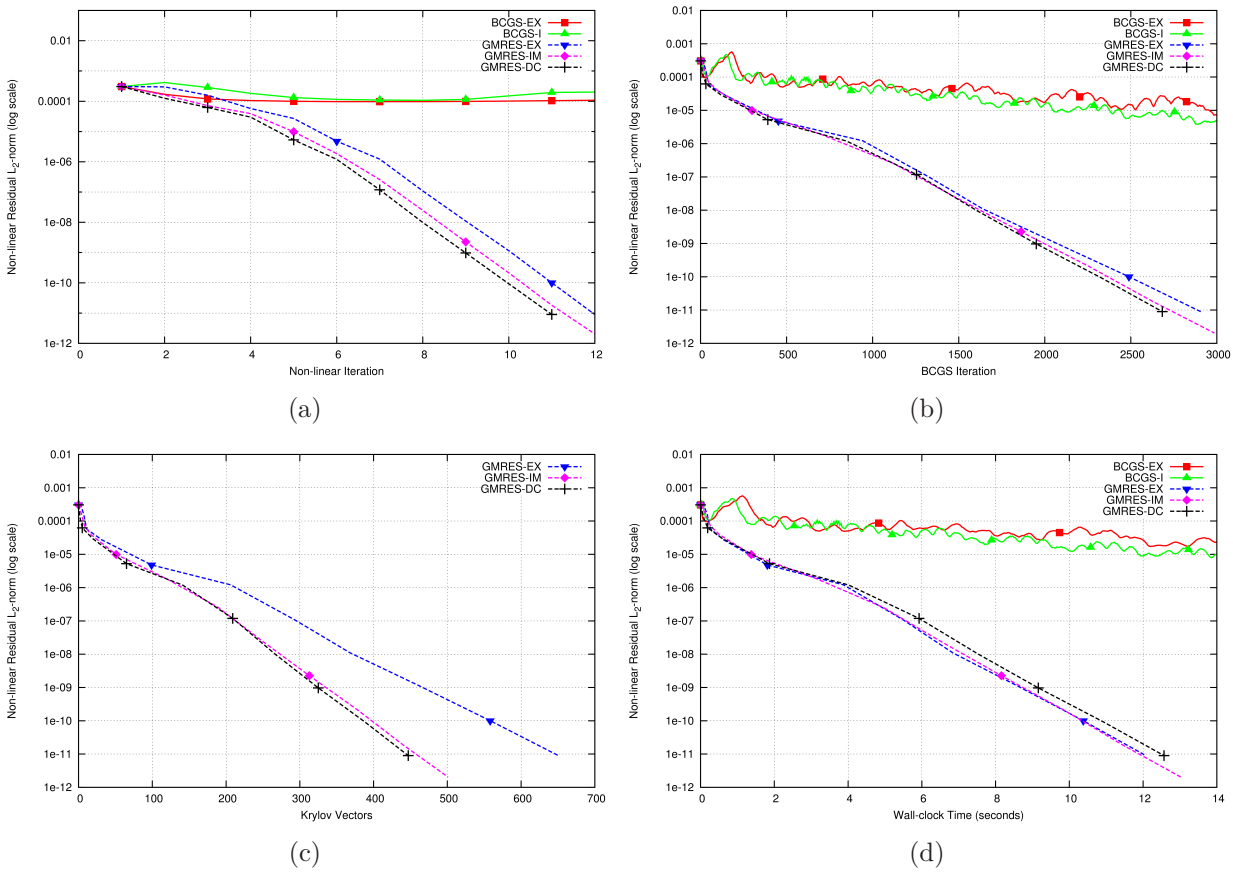


Figure 6.5: Comparison of convergence for aerodynamic test case 1 with 3 time-spectral time instances using the 5 different solver variants discussed with non-linear iterations (a), BCGS iterations (b), Krylov vectors (c), and wall-clock time (d) on the x-axis

axis such that the full convergence histories of both BCGS-EX and BCGS-IM are not visible. However, it can readily be seen how much more efficient the GMRES-based solvers are. Also note that all solvers were run on identical hardware. Table (6.1) summarizes the convergence of the various solvers that are shown in Figure (6.5), including at what point both BCGS-EX and BCGS-IM eventually reach convergence. It can be seen that the GMRES-based

Table 6.1: Convergence of the first aerodynamic test case with $N = 3$ time instances

Solver	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
BCGS-EX	6,699	21,442	N/A	161
BCGS-IM	2,777	16,870	N/A	98
GMRES-EX	12	2,907	651	12
GMRES-IM	12	2,986	501	13
GMRES-DC	11	2,683	447	13

solvers are by far the fastest in every measure. In comparison to the BCGS-EX solver, the BCGS-IM solver reduces the number of non-linear iterations by a factor of 2.4, the number of BCGS iterations by a factor of 1.3, and the wall-clock time by a factor of 1.6. These gains are accomplished almost exclusively by the increased CFL number allowed by BCGS-IM. A CFL number of 7.2 can be used for BCGS-EX, while a CFL number 4.17 times higher (i.e. 30.0) can be used for the BCGS-IM solver.

The GMRES-EX method gains over the BCGS-IM by using 17% the number of BCGS iterations, while decreasing the number of non-linear iterations by a factor of 231 and the wall-clock time by a factor of about 8.2. The gains of GMRES-EX first occur by allowing each BCGS iteration to use a higher CFL number (i.e. 20.0) than can be used by BCGS-EX as the linear solver, itself. This solver also allows each non-linear iteration to make use of more BCGS iterations. This aspect decreases the overhead cost of each BCGS iteration, which decreases the total wall-clock time even further. Because this test case using three time instances is well-conditioned for time-spectral solution, the more complicated GMRES-IM and GMRES-DC solvers exhibit few efficiency gains. Most notably, the number of Krylov vectors used is reduced, but the increased overhead costs (because of increased MPI commu-

nications) of these two methods more than compensate for this computational savings when it comes to wall-clock time.

Results for the various solvers when $N = 15$ are presented in Table (6.2) except BCGS-EX is omitted since it is so clearly inefficient. The convergence of this test case, run with

Table 6.2: Convergence of the first aerodynamic test case with $N = 15$ time instances

Solver	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
BCGS-IM	16,730	38,566	N/A	3,550
GMRES-EX	12	8,687	2,502	316
GMRES-IM	13	5,469	1,832	204
GMRES-DC	13	5,859	1,111	196

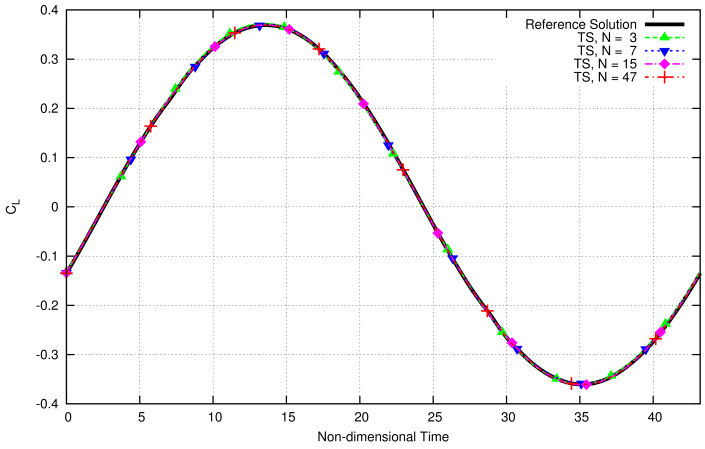
$N = 15$ time instances, follows similar trends to $N = 3$. Because of the increased stiffness of the problem introduced by the additional time instances, the GMRES-IM and GMRES-DC solvers both show efficiency gains. These benefits occur in reduced numbers of BCGS iterations and Krylov vectors primarily as a result of the implicit coupling among the time-instances present in these solvers. Overall, it can be seen that the addition of FGMRES as the linear solver greatly increases the efficiency of time-spectral methods.

6.2 Test Case 2: AGARD Test Case No. 5

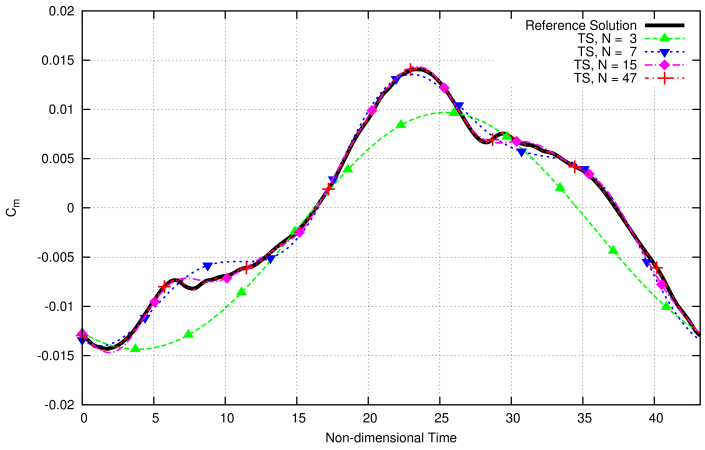
The second test case uses the same computational mesh as the first. A forced, pitching oscillation of the NACA-0012 airfoil at a Mach number of 0.755 and a mean incidence α_0 of 0.016 degrees is prescribed at the quarter chord of the airfoil as follows:

$$\alpha(t) = \alpha_0 + \alpha_A \sin(\omega t). \tag{6.3}$$

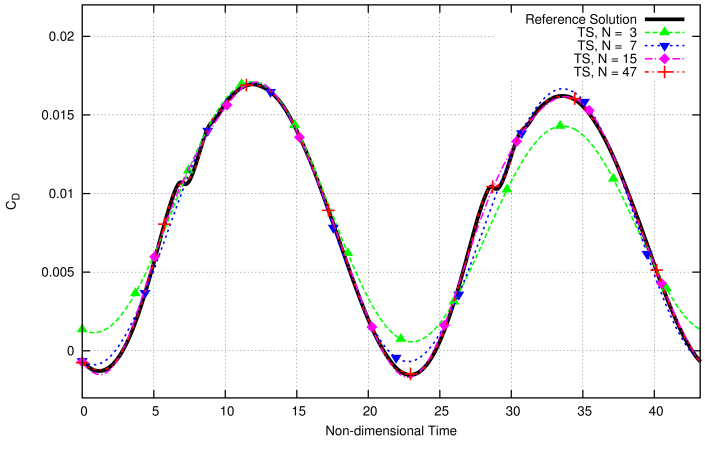
The reduced frequency k_c is equal to 0.0814 and the amplitude α_A is equal to 2.51 degrees. This test case corresponds to the AGARD test case No. 5 [78]. Figure (6.6a) shows the comparison of the lift coefficient versus non-dimensional time between a reference solution obtained using a second-order accurate time-implicit solver with $\Delta t = T/4096$ (where T



(a)



(b)



(c)

Figure 6.6: Comparison of computed lift coefficient (a), moment coefficient (b), and pressure drag coefficient (c) using the time-spectral method to a reference, time-implicit solution for aerodynamic test case 2

denotes the period of airfoil motion) and the time-spectral method with $N = 3, 7, 15$, and 47 time instances. For this case, the lift generated by the time-spectral method with even 1 harmonic or 3 time instances shows reasonable agreement with the reference solution. Figure (6.6b) shows the comparison of the moment coefficient versus non-dimensional time for the same reference and time-spectral solutions. The moment history contains multiple harmonics and thus is not captured accurately with $N = 3$ in the TS method; in fact, a rather large number of time instances, $N = 47$, is needed to produce near-exact agreement. Figure (6.6c) shows the comparison of the drag coefficient versus non-dimensional time for those same solutions. It should be noted that, since the results use the Euler equations, the drag shown is pressure drag and does not contain viscous effects. The drag coefficient shows good agreement for the $N = 47$ solution. The most difficult areas of the pressure drag curve to resolve occur around $t = 7$ and $t = 28$ non-dimensional seconds, where the $N = 47$ and reference solutions show a slight reversal. It can be seen that the results of the TS method converges to the reference solution as the number of time instances increases.

Figure (6.7) compares the wall-clock time required to complete a converged solution for different numbers of time instances for the time-spectral method using the GMRES-DC solver and for different numbers of time steps per period for the time-implicit method. Visual examination of the accuracy of the time-implicit solutions show that at least 256 time steps per period are needed to replicate the reference solution reasonably well. As can be seen from this figure, time-spectral solutions using up to 27 time instances can be converged in about the same or less time than the 256 time-step-per-period time-implicit solution. In fact, 47 time instances are needed to reproduce the reference solution to the same visual accuracy as the 256 time-step-per-period time-implicit solution. This $N = 47$ time-spectral solution requires 2.4 times the wall-clock time of the 256 time-step-per-period time-implicit solution to converge, indicating that future work remains to make the time-spectral method more efficient than the time-implicit method for cases like the AGARD 5 test case, which has only a single harmonic of prescribed motion, but a flow response a large number of harmonics of the frequency of motion.

It should be noted that all time-implicit solutions are run on a single 16-core cluster

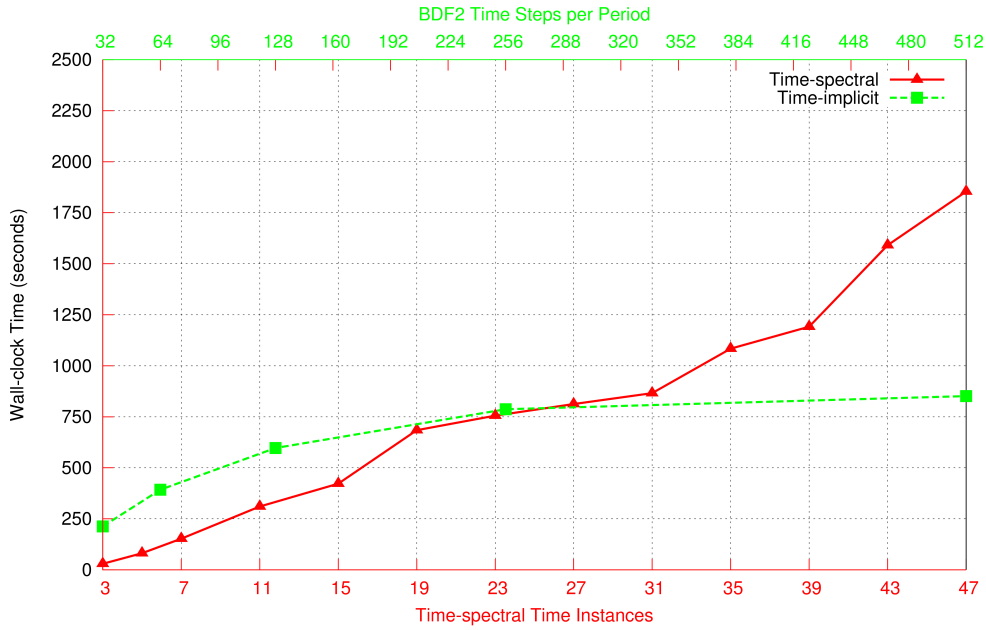


Figure 6.7: Comparison of wall-clock time to convergence for time-implicit and time-spectral methods with different numbers of time steps per period (top axis) and time instances (bottom axis) for aerodynamic test case 2

node (by splitting the spatial grid across 16 cores). Time-spectral solutions for $N = 3, .., 15$ are run on a single 16-core cluster node. The time-spectral solutions for 17 or more time instances are run on as many nodes as are necessary such that each time instance uses one CPU core. For example, the 39 time-instance solution is run on 3× 16-core cluster nodes with each time instance using one core; thus 9 cores among the three nodes are left idle. It should also be noted that the wall-clock time of the time-spectral solutions can be decreased by utilizing additional computational resources. For example, if each time instance is run on 16 cores (similarly to time-implicit) even the $N = 47$ time instances, time-spectral solution can be completed in less wall-clock time than the time-implicit solution.

If the convergence of the first and second test cases are compared, it is obvious that the first test case converges much more quickly than the second case when using the time-spectral method. Two factors are the main causes of the increased convergence time of the second test case: first, the transonic nature of the freestream Mach number of the second case, and second, the higher reduced frequency of the first harmonic of its airfoil motion.

The transonic Mach number naturally means that shock waves will be present in the

flow. For this particular test case, the shock waves develop on the the upper and lower surfaces at different moments in time, periodically depending on the angle-of attack of the airfoil at each moment. These moving, disappearing and reappearing shock waves complicate the flow solution and introduce higher frequency (possibly non-smooth) content into the time-dependent solution. The increased reduced frequency of the second test case also leads to convergence degradation since the coupling between the time-spectral instances is increased, as determined by the coefficients in equations (2.36) and (2.37), which are directly proportional to the reduced frequency. It can also be seen from equations (4.3) and (4.4) that an increase in the reduced frequency results in a smaller pseudo-time step size for a given CFL number. Despite these phenomena that cause the second test case to be more difficult to converge time-spectrally, time-spectral methods still prove competitive when compared to time-implicit methods when the higher harmonic content can be neglected.

6.2.1 Convergence Study

Figure (6.8) plots the convergence of the time-spectral method for the AGARD 5 test case using $N = 3$ time instances for the five different solver configurations summarized in Table (4.2), just as was done for the first test case. Similarly, in order to resolve any details for the convergence of the GMRES based solvers, it is necessary to scale the x-axis such that the full convergence history of both BCGS-EX and BCGS-IM are not visible. Table (6.3) summarizes the convergence of the various solvers that were shown in Figure (6.8). For the second test

Table 6.3: Convergence of the AGARD 5 test case with $N = 3$ time instances

Solver	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
BCGS-EX	106,844	213,687	N/A	2,317
BCGS-IM	8,744	17,487	N/A	214
GMRES-EX	151	9,205	4,564	72
GMRES-IM	25	6,101	2,061	37
GMRES-DC	21	6,800	540	30

case, it can be seen that the GMRES-DC solver is the most computationally efficient in

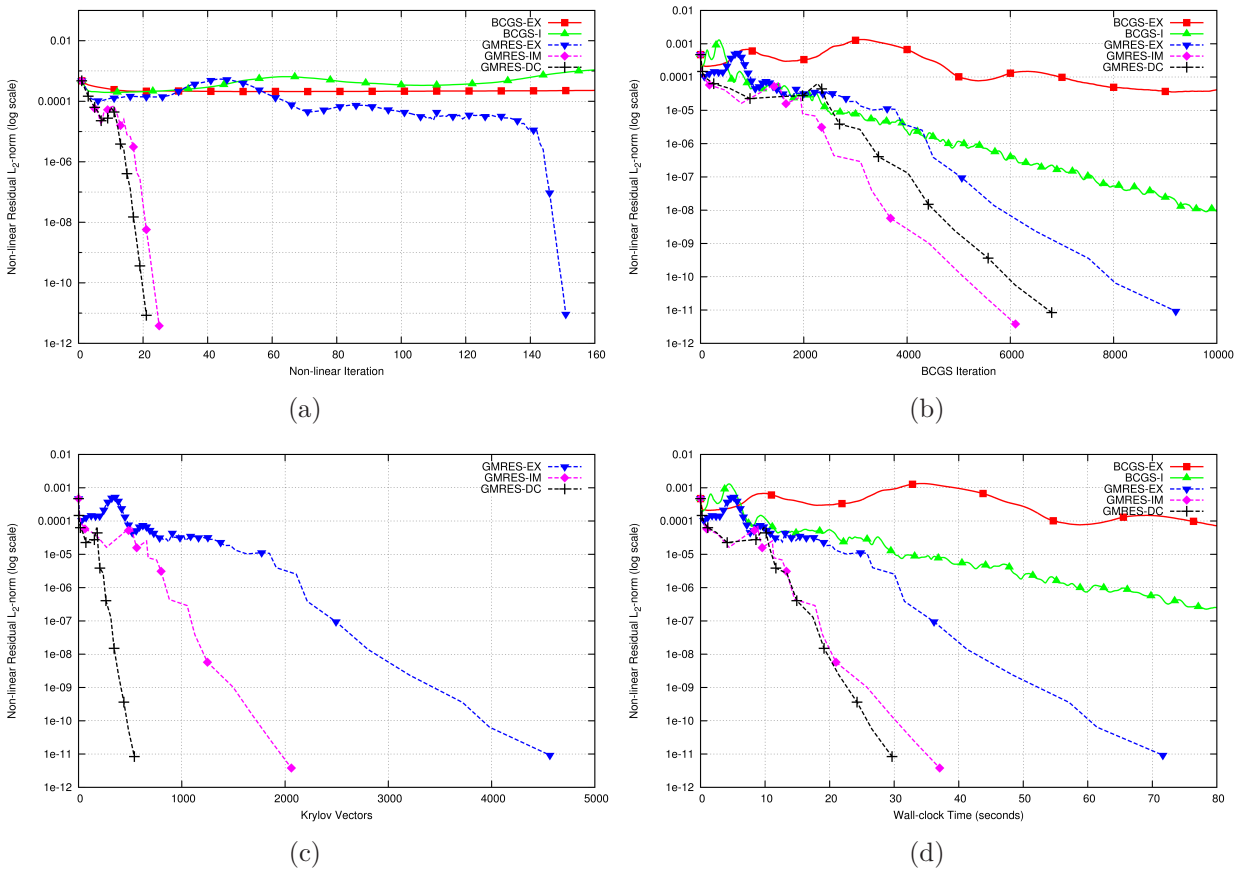


Figure 6.8: Comparison of convergence for aerodynamic test case 2 with 3 time-spectral time instances using the 5 different solver variants discussed with non-linear iterations (a), BCGS iterations (b), Krylov vectors (c), and wall-clock time (d) on the x-axis

three of the four measures. When comparing the BCGS-EX solver to the BCGS-IM solver, the latter reduces the number of non-linear iterations, the number of BCGS iterations, and the wall-clock time all by more than a factor of 10. These gains are accomplished almost exclusively by the increased CFL number allowed by the time-implicit BCGS. A CFL number of only 0.25 can be used for BCGS-EX, while a CFL number more than 10 times higher, i.e. 2.6, can be used for the BCGS-IM solver.

The GMRES-EX method gains over the BCGS-IM method by halving the number of BCGS iterations used, while decreasing the number of non-linear iterations by a factor of 58 and the wall-clock time by a factor of almost 3. The gains of GMRES-EX were already outlined for the first test case but are henceforth summarized as follows: each BCGS iteration is more efficient when used as a preconditioner rather than as the main linear solver and GMRES allows the use of more BCGS iterations for every non-linear iteration, reducing overhead costs. The further efficiency increases provided by the GMRES-IM solver over the GMRES-EX solver occur, again, because of the implicit coupling among time instances.

The gains of adding a defect-correction step to the GMRES preconditioner, which were not significant for the first test case, accrue mainly because of the decreased numbers of non-linear iterations and Krylov vectors, which decreases the total overhead cost of convergence to the final solution. This is evidenced by the number of BCGS iterations increasing slightly as compared to the GMRES-IM method while the number of non-linear iterations and the number of Krylov vectors used decreases.

Results for the various solvers when $N = 15$ are presented in Table (6.4) except BCGS-EX is again omitted. As can be seen once again, the same trends hold for the $N = 15$

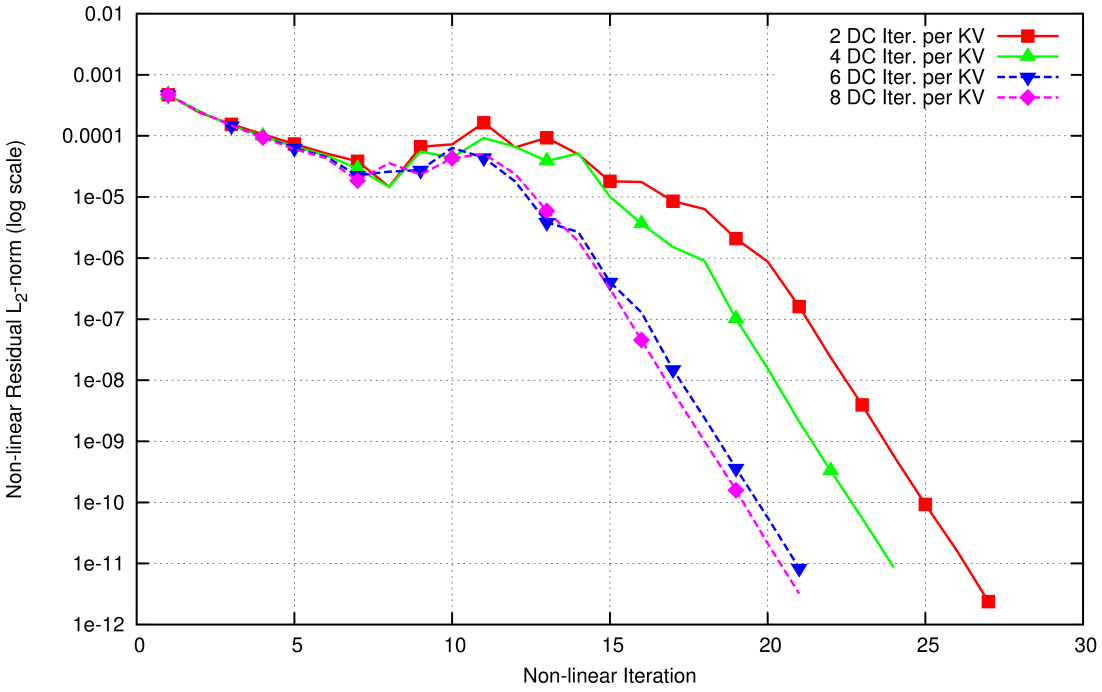
Table 6.4: Convergence of the AGARD 5 test case with $N = 15$ time instances

Solver	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
BCGS-IM	12,992	27,653	N/A	2,430
GMRES-EX	125	22,197	11,098	873
GMRES-IM	124	11,551	3,922	440
GMRES-DC	22	14,525	383	423

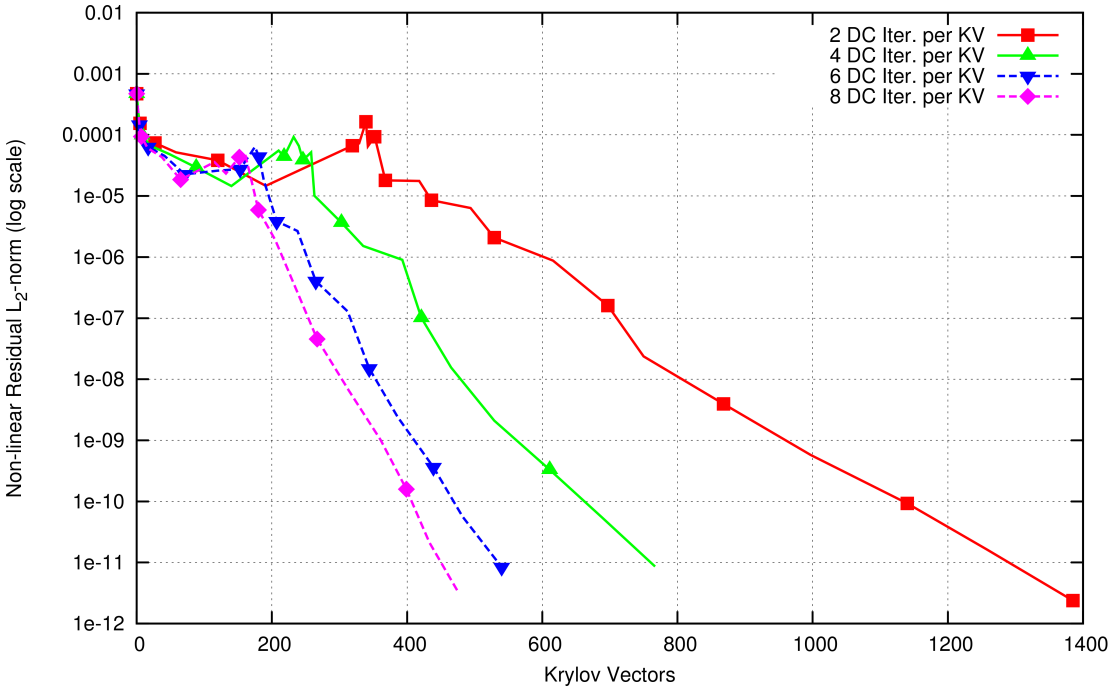
solutions as were discussed for the $N = 3$ solutions. It should be reiterated that all cases for both $N = 3$ and $N = 15$ have been run on a single 16-core cluster node. This identical hardware means that each time instance of the $N = 3$ solutions is run on 5 cores, whereas each time instance of the $N = 15$ solution is run on a single core. The wall-clock time to convergence of the $N = 15$ solution can be greatly decreased if each time instance is run on 5 cores, as was done in the $N = 3$ case. This will have required a total of 75 cores for the $N = 15$ case.

6.2.2 The Effects of Additional Preconditioning

To establish a better idea of how the time-spectral solver might be further improved, particularly for stiffer cases such as the second test case, the convergence effects of the amount of preconditioning performed on each Krylov vector is examined. It is expected that increasing the number of defect-correction steps used to precondition each Krylov vector will reduce the number of Krylov vectors needed to converge a solution. The total number of Krylov vectors needed for a converged solution depends mainly on the number of Krylov vectors needed for each linear solve. This value is in turn determined by the convergence criteria of the linear FGMRES solver each time it is called and how much residual reduction each Krylov vector provides (i.e. how efficient the Krylov vector is). For this study, the convergence criteria was held constant by requiring the linear residual to be reduced by 0.8 orders of magnitude by FGMRES. Thus, the number of Krylov vectors used is a good proxy to measure the efficiency of each Krylov vector. It is believed that more preconditioning will make each Krylov vector more efficient, thereby requiring fewer Krylov vectors in sum. Figure (6.9) plots the non-linear iteration and Krylov vector convergence for $N = 3$ time instances using $2, 4, 6$, and 8 defect-correction steps per Krylov vector. This data is then summarized with additional data points in Table (6.5). As can be seen from this figure and table, the number of cumulative Krylov vectors needed to produce a converged solution can be reduced substantially when more preconditioning iterations are used. Additionally, added preconditioning also moderately reduces the number of non-linear iterations needed. However, both the cumulative number of BCGS iterations used and the wall-clock time



(a)



(b)

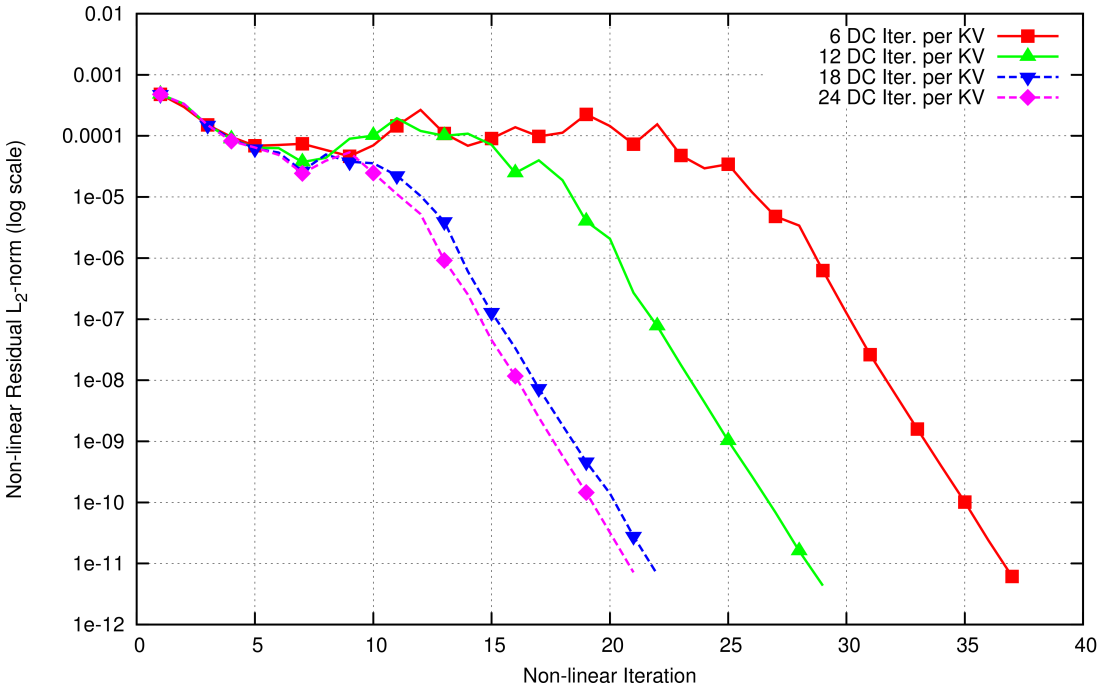
Figure 6.9: Comparison of convergence of aerodynamic test case 2 using 3 time-spectral time instances and the GMRES-DC solver using different numbers of defect-correction iterations per Krylov vector with Non-linear iterations (a) and Krylov vectors (b) on the x-axis

Table 6.5: Comparison of the nature of convergence for aerodynamic test case 2 with 3 time instances

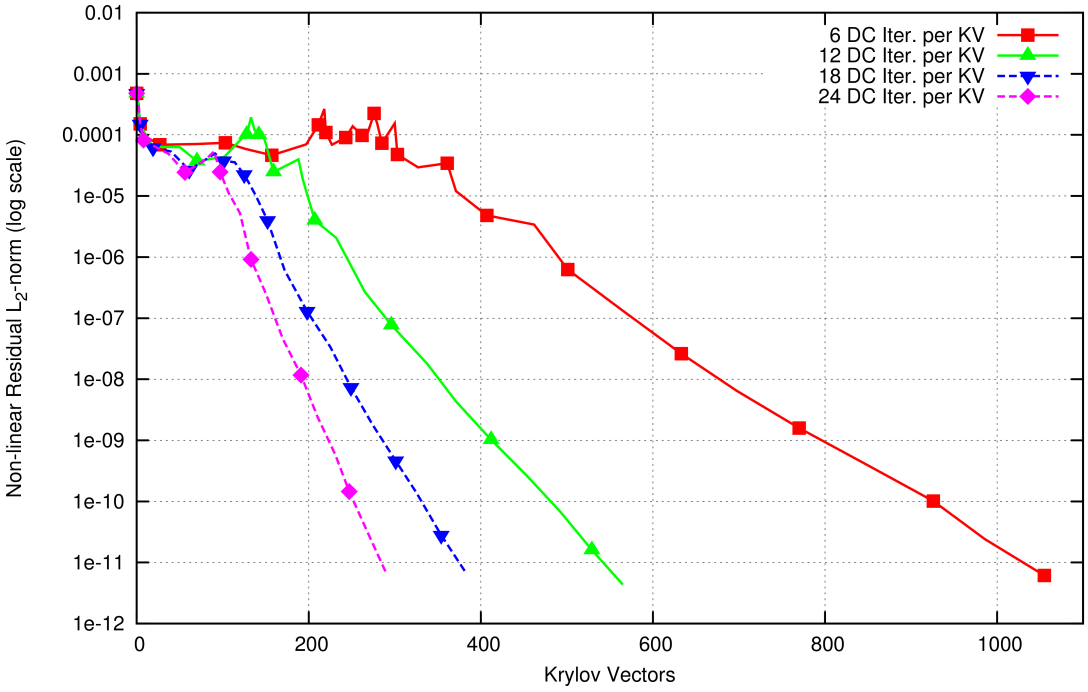
Defect Correction Steps	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
2	27	7,529	1,385	35.9
3	24	7,044	994	32.7
4	24	6,923	767	31.8
5	22	6,937	659	30.5
6	21	6,800	540	29.7
7	21	7,529	509	32.1
8	21	7,902	476	34.7
9	21	7,994	431	35.6
10	21	8,376	405	36.5

are little affected. These results indicate that one way to make the solver more efficient is to use a better preconditioning method than BCGS, such that each preconditioning cycle accomplishes a greater residual reduction, i.e. is more computationally efficient.

Figure (6.10) plots the same information as the last figure except for $N = 15$ time instances using $6, 12, 18$, and 24 defect-correction iterations per Krylov vector, and Table (6.6) summarizes these results with more data points. Similarly, Figure (6.11) is a comparison of the convergence for $N = 31$ time instances using $5, 15, 25$, and 35 defect-correction preconditioning iterations per Krylov vector. Again, this data is summarized in Table (6.7), with additional data points. The same Krylov vector trend as was seen with the $N = 3$ case holds for the $N = 15$ and $N = 31$ cases. It can also be seen that the number of BCGS iterations increases as the number of time instances increases, while the number of non-linear iterations needed remains roughly the same (for the 3 and 15 time instance cases) when enough preconditioning iterations are used, and the number of cumulative Krylov vectors needed does not necessarily grow as more time instances are utilized. When examining the data for the $N = 31$ case, it is seen that the number of non-linear iterations decreases monotonically at first, then spikes at 30 defect-correction steps per Krylov vector, after which it again decreases monotonically. For some as yet unknown reason, when 30 or more defect-correction steps are used, the GMRES CFL number growth rate must be reduced for

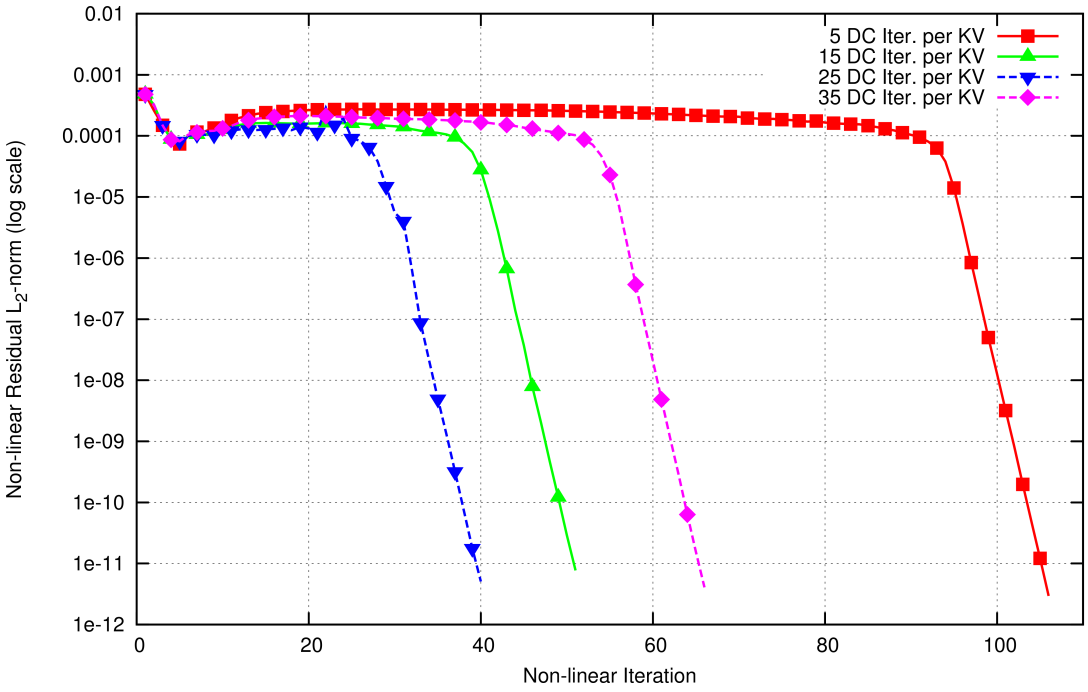


(a)

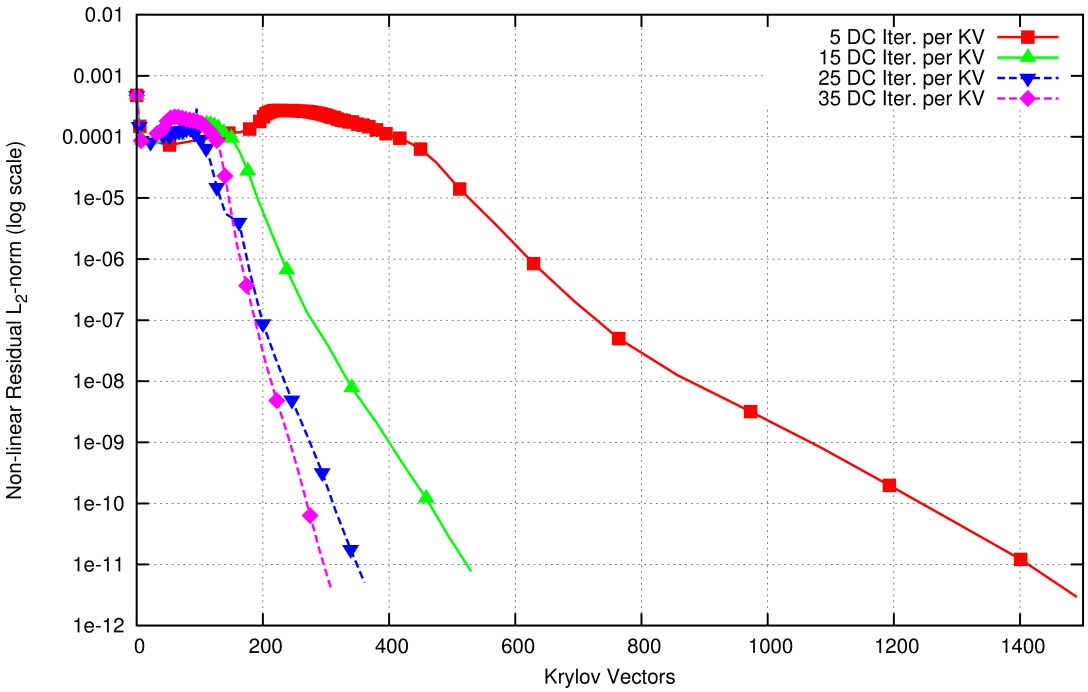


(b)

Figure 6.10: Comparison of convergence of aerodynamic test case 2 using 15 time-spectral time instances and the GMRES-DC solver using different numbers of defect-correction iterations per Krylov vector with non-linear iterations (a) and Krylov vectors (b) on the x-axis



(a)



(b)

Figure 6.11: Comparison of convergence of aerodynamic Test case 2 using 31 time-spectral time instances and the GMRES-DC solver using different numbers of defect-correction iterations per Krylov vector with non-linear iterations (a) and Krylov vectors (b) on the x-axis

Table 6.6: Comparison of the nature of convergence for aerodynamic test case 2 with 15 time instances

Defect Correction Steps	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
6	37	13,351	1,055	538
9	30	13,592	713	524
12	29	14,311	565	461
15	25	14,668	463	478
18	22	14,525	383	423
21	22	14,963	338	479
24	21	14,596	289	453
27	22	15,940	282	468

Table 6.7: Comparison of the nature of convergence for aerodynamic test case 2 with 31 time instances

Defect Correction Steps	**Non-linear Iter.**	**BCGS Iter.**	**Krylov Vectors**	**Wall-clock Time (s)**
5	106	15,650	1,490	888
10	58	16,270	794	1,033
15	51	16,342	530	865
20	49	17,259	423	935
25	40	18,342	361	974
30	79	19,432	350	1,068
35	66	20,320	308	1,103
40	41	19,899	250	1,232

the solution to be convergent. Why this reduction in CFL number growth occurs and how non-linear convergence can be restored warrants future investigation.

It can also be observed that the number of cumulative BCGS iterations needed for solution convergence is inversely proportional to the BCGS CFL number. Table (6.8) summarizes this proportionality in its fourth column which contains the product of the CFL number and the cumulative BCGS iteration count. The inverse proportionality is shown by the numbers in this column being roughly equal. This result indicates that another avenue to obtain a better preconditioner would eliminate this CFL number constraint altogether through the use of a more global preconditioner, such as incomplete LU-decomposition (ILU).

Table 6.8: Proportionality of BCGS iteration count to BCGS CFL number

Time Instances	**BCGS CFL**	**Fewest BCGS Iter. (NBCGS)**	**BCGS CFL $\times$ NBCGS**
3	2.6	$6,923$	$18,000$
15	1.4	$13,351$	$18,691$
31	1.2	$15,650$	$18,780$

Chapter 7

Time-spectral Aeroelastic Results

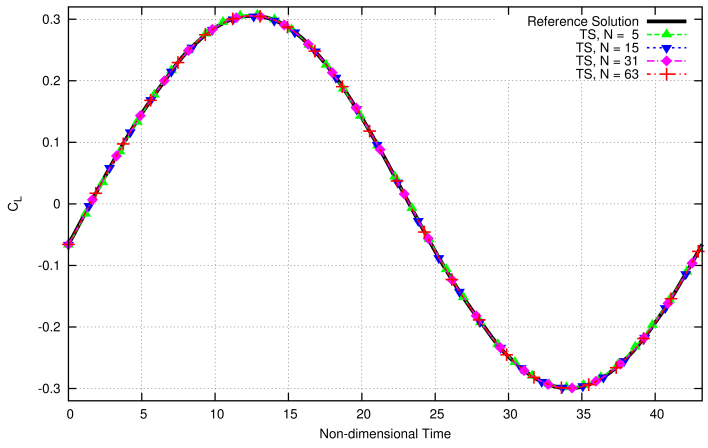
7.1 Aeroelastic Test Case

To examine the convergence of the aeroelastic solver, the AGARD5 test case presented previously is modified so that it responds aeroelastically. The same NACA-0012 unstructured, computational mesh of 4,471 nodes and 8,747 triangles is used. The freestream Mach number of 0.755 and mean incidence α_0 of 0.016 degrees are retained. The airfoil again undergoes a forced pitching oscillation of $\alpha_A = 2.51^o$, however, this oscillation occurs about the airfoil elastic axis, instead of the quarter-chord as was true for the AGARD5 test case. In addition to the forced pitching, the airfoil is allowed to respond aeroelastically in both pitch and plunge such that the true pitch angle of the airfoil at a given time is the combination of the prescribed and aeroelastic contributions to pitch. The reduced frequency of the forced pitching is $k_c = 0.0814$ just as in the AGARD5 test case. The structural parameters for this and all other aeroelastic cases considered are as follows:

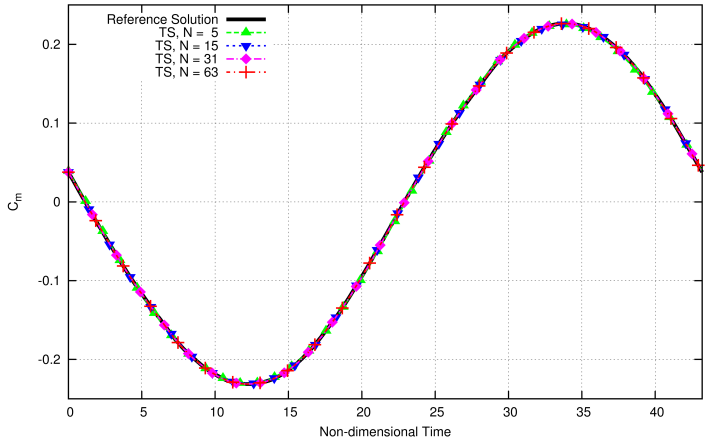
$$\begin{aligned} V_f &= 0.50 \\ x_\alpha &= 1.8 \\ r_\alpha^2 &= 3.48 \\ \frac{\omega_h}{\omega_\alpha} &= 1.0 \\ \mu &= 64.0 \\ a &= \text{-}2.0. \end{aligned}$$

The quantity a is the non-dimensional elastic axis location along the chord of the airfoil measured from the mid-chord of the airfoil when it is in the neutral position. Since it is non-dimensionalized by the semi-chord of the airfoil, the elastic axis is located half a chord length ahead of the leading edge of the airfoil in this particular case. As was noted above, this is also the axis about which the forced pitching occurs.

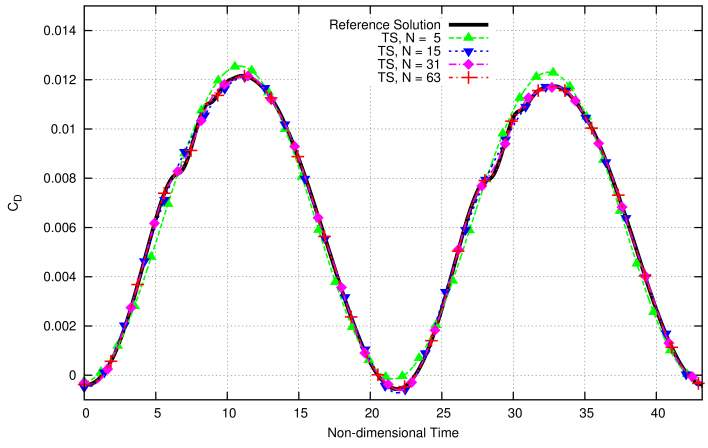
Figure (7.1) compares the lift, moment, and pressure drag coefficients of a single period of motion of this test case for different numbers of time-spectral time instances to a reference time-implicit solution with 4,096 time steps per period. It appears that all the time-spectral solutions show good agreement in lift and moment, while only the 63 time instance solution shows visually exact agreement in pressure drag. However, visual inspection alone is not always adequate to assess accuracy. Figure (7.2) plots the root-mean-square (RMS) error of the solutions in lift, moment, and drag coefficients as well as pitch and plunge for different numbers of time steps per period and time instances as compared to the reference solution. For the time-spectral solutions, RMS error is calculated by reconstructing the flow and structural variables for each time-spectral solution to the same 4,096 points in time per period as are contained in the reference solution. Lift, moment, and drag are then calculated by integrating the pressure over the surface of the airfoil for these same 4,096 points. On the other hand, for the calculation of the RMS error for the BDF2 solutions, the error is calculated using the number of time steps of each of those BDF2 solutions; this necessarily means that the BDF2 solutions must have a number of time steps per period that is a factor of the 4,096 used by the reference solution. For both time-spectral and time-implicit methods, the error presented in Figure (7.2) is the root-mean-square of the difference between the approximate solutions and the reference solution over a single period of airfoil motion. This same method is used for all $L_2 - norm$ error calculations presented in the rest of this work. Since peak lift and moment coefficients are on the order of 0.1, lift and moment are considered accurate when RMS error in these quantities is less than 0.0002 or 0.2% which occurs with 47 time instances and 256 time steps per period. Peak drag, pitch, and plunge are all on the order of 0.01 so the solution is precise when the error in these quantities is also less than 0.2% or 2×10^{-5}. An accurate solution in drag occurs for 512 time steps per



(a)



(b)



(c)

Figure 7.1: Comparison of computed lift coefficient (a), moment coefficient (b), and pressure drag coefficient (c) using the time-spectral method to a reference, time-implicit solution for first aeroelastic test case

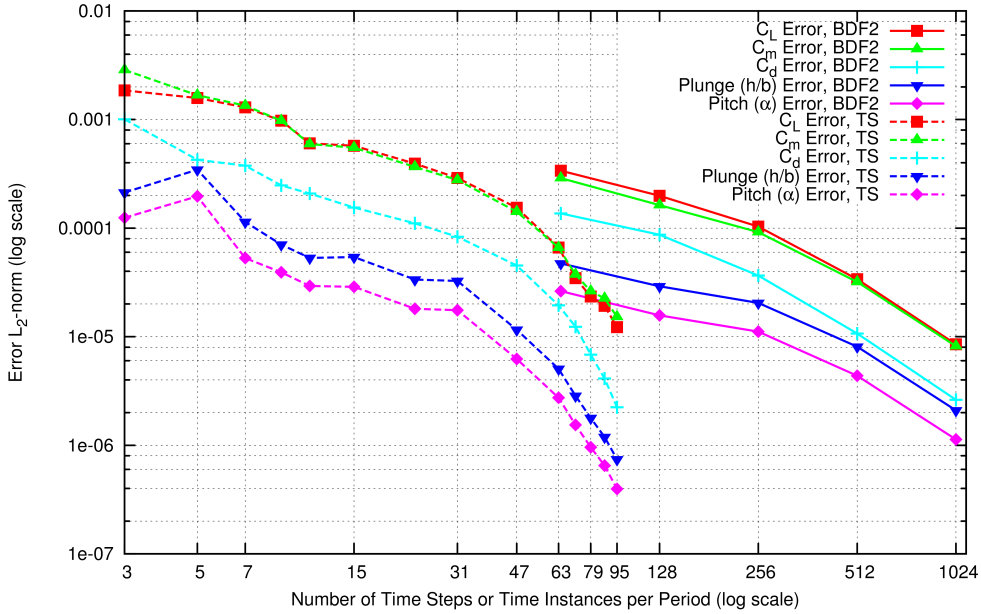


Figure 7.2: Aeroelastic test case error L_2-norm for time-implicit and time-spectral methods with varying numbers of time steps per period and time instances ($M_\infty = 0.755, k_c = 0.0814$)

period and 63 time instances. By this definition, an accurate solution in pitch occurs for 128 time steps per period and 23 time instances, while a precise solution in plunge occurs using 256 time steps per period and 47 time instances. For this case, the drag coefficient is the most limiting quantity.

Another trend that should be noted in Figure (7.2) is the exponential convergence of the accuracy of the time-spectral method as time instances are added. Specifically, this trend is noted above 31 time instances. Exponential convergence is the expected trend for spectral methods and means that the rate of error decrease of the method accelerates as more time instances are added so convergence can become faster than any polynomial order. This contrasts with the time-implicit method which has second-order accuracy convergence. For a second-order accurate method in time, the slope of the error convergence is expected to be 2.0. For this test case, the slope from 512 to 1024 time steps per period for all five error measures ranges between 1.946 for plunge and 2.025 for drag, while the greatest slope for the time spectral method is 6.907 and occurs for C_D between 87 and 95 time instances.

Figure (7.3) compares the wall-clock time required to complete a converged solution for different numbers of time instances for the time-spectral method and for different numbers

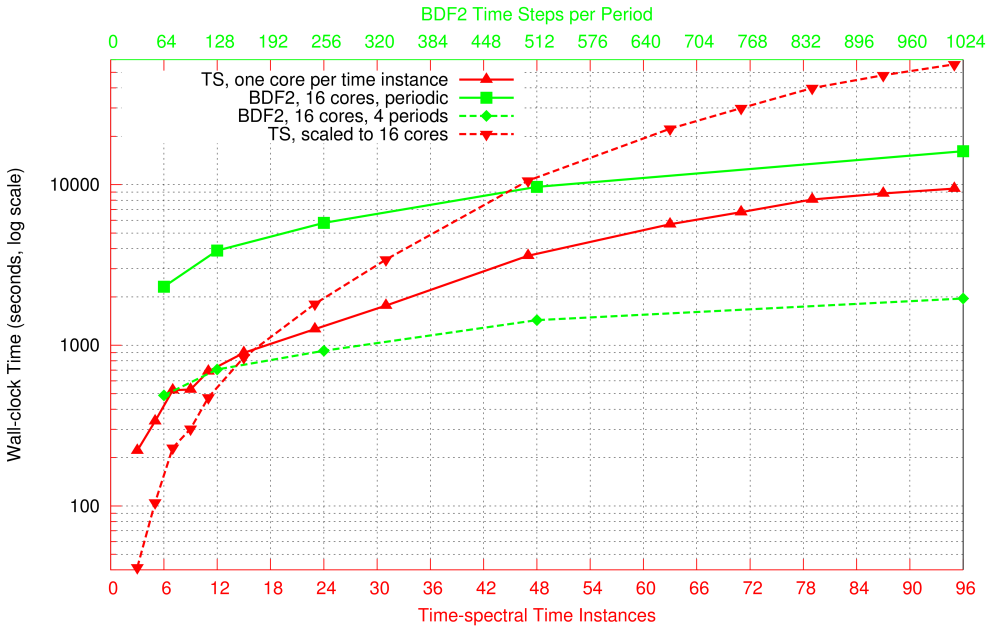


Figure 7.3: Comparison of wall-clock time to convergence for time-implicit and time-spectral methods with different numbers of time steps per period (top axis) and time instances (bottom axis) for the first aeroelastic test case

of time steps per period for the time-implicit method. Two series are included for each solution type. The time implicit data are all run on one cluster node with 16 CPU cores. The two time-implicit series plot the time needed for a purely-periodic solution (solid line, squares) and 4 periods of the solution (dashed line, diamonds). It should be noted that 19 periods are needed before the 64 time step per period solution attains pure periodicity while 33 periods are needed before the 1024 time steps per period solution becomes purely periodic. The determination of periodicity is made by comparing the successive periods of the time-implicit solution to the reference solution; the period where the first four non-zero digits of the error in all three force coefficients and both structural variables match those of the next period is used as the period at which the solution attains a purely periodic state. The two time-spectral series plot the time needed for convergence when one CPU core is used per time instance (solid, triangles) and when the solution is calculated on a single, 16 CPU core node (assuming linear scaling, dashed, inverted triangles). Figures (7.4) and (7.5) combine the data contained in Figures (7.2) and (7.3) in a way that might be more accessible. Specifically, wall-clock time is plotted as a function of the error in drag with the

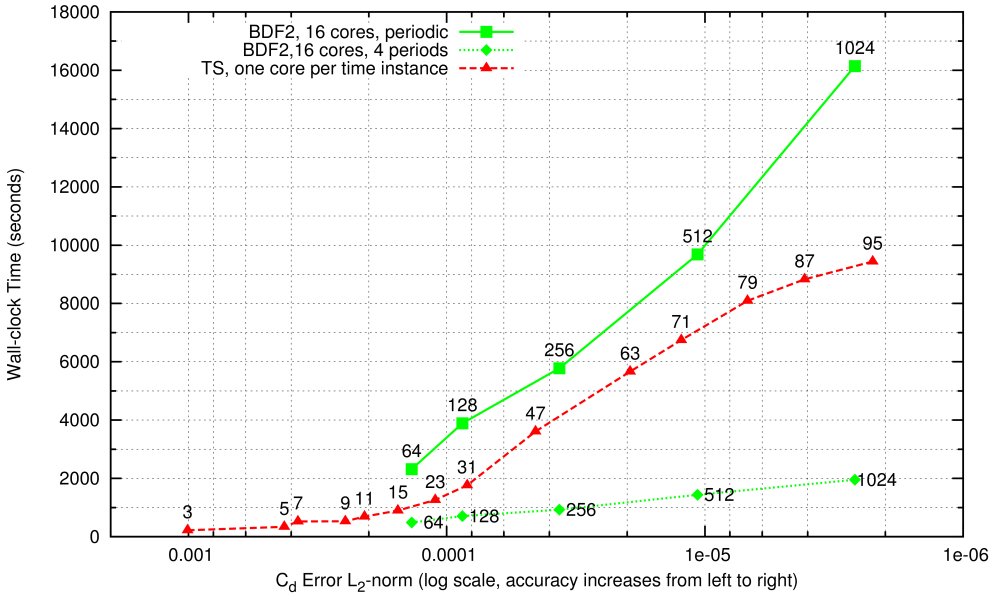


Figure 7.4: Comparison of wall-clock time to convergence for time-implicit and time-spectral methods as a function of the $L_2 - norm$ of the error in C_D

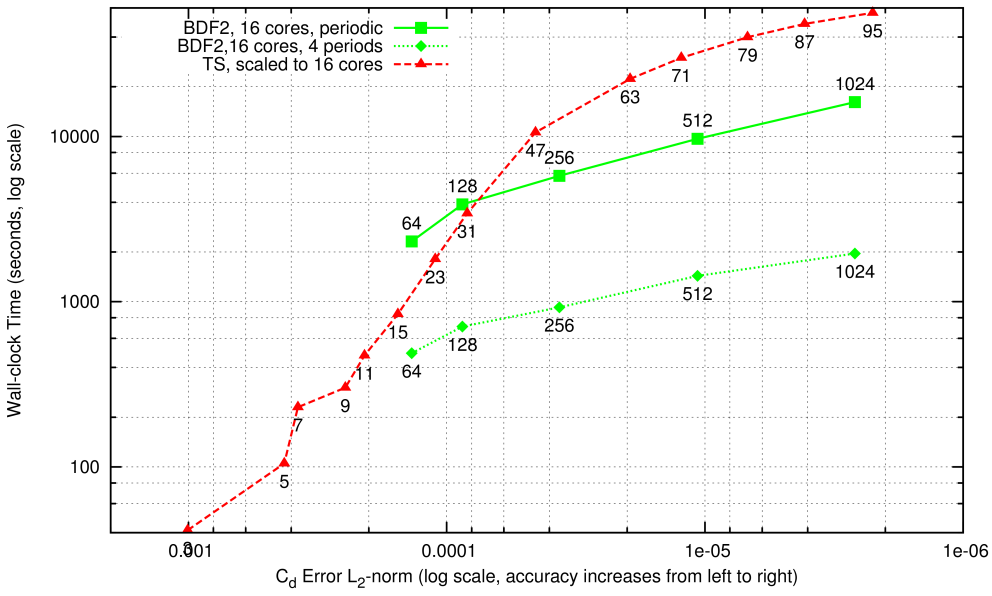


Figure 7.5: Comparison of wall-clock time to convergence for time-implicit and time-spectral methods as a function of the $L_2 - norm$ of the error in C_D with the time-spectral method linearly scaled to the same hardware as the time-implicit results

points labeled with their numbers of time steps per period or time instances. Drag error was chosen since it is the most limiting quantity, as noted above. For solutions that are precise in the limiting drag coefficient, the 512 time step per period time-implicit solution is compared to the 63 time instance time-spectral solution. To run enough periods to reach a purely periodic time-implicit solution (27 periods) 9,680 seconds are required. If, however, only 4 periods are used, only 1,434 seconds are needed. To converge a time-spectral solution with 63 time instances, 5,665 seconds are needed when using one time instance per CPU core. If, however, the same 16 computational cores as were used for the time-implicit solution were used for the time-spectral solution as well, 22,306 seconds would be required.

All aeroelastic solutions compared up to this point have used the same convergence tolerance where the $L_2 - norm$s of both the flow and structural non-linear residuals must be below 1×10^{-9} both for the time-implicit scheme and for all time instances in the time-spectral method.

7.1.1 Error Variation as Convergence Tolerance is Relaxed

The convergence tolerance noted above is close to machine zero. Such a restrictive tolerance may not be necessary to attain the error limits desired, but was used for validation purposes to guarantee that the error being measured was the temporal error and did not include algebraic error as a result of incomplete convergence. It has been suggested that perhaps a less strict convergence tolerance can be used to achieve the same RMS error in a given period, but at a reduced wall-clock time cost for both the time-implicit and time-spectral solutions. Figure (7.6) plots the error in the usual quantities on the left axis and the wall-clock time on the right axis versus the convergence tolerance on the x-axis (with stricter tolerances to the left). Since the 512 time-step-per-period solution was considered precise, it is this solution that is rerun with the lower convergence tolerances. As can be seen, the convergence tolerance can be relaxed to 1×10^{-7} while keeping the error below the tolerances established above. This reduces the wall-clock time to obtain a periodic solution by almost 50% from 9,680 seconds for the original tolerance to 4,980 seconds for the less strict 1×10^{-7}.

Figure (7.7) plots the same quantities as the last figure, except for the time-spectral

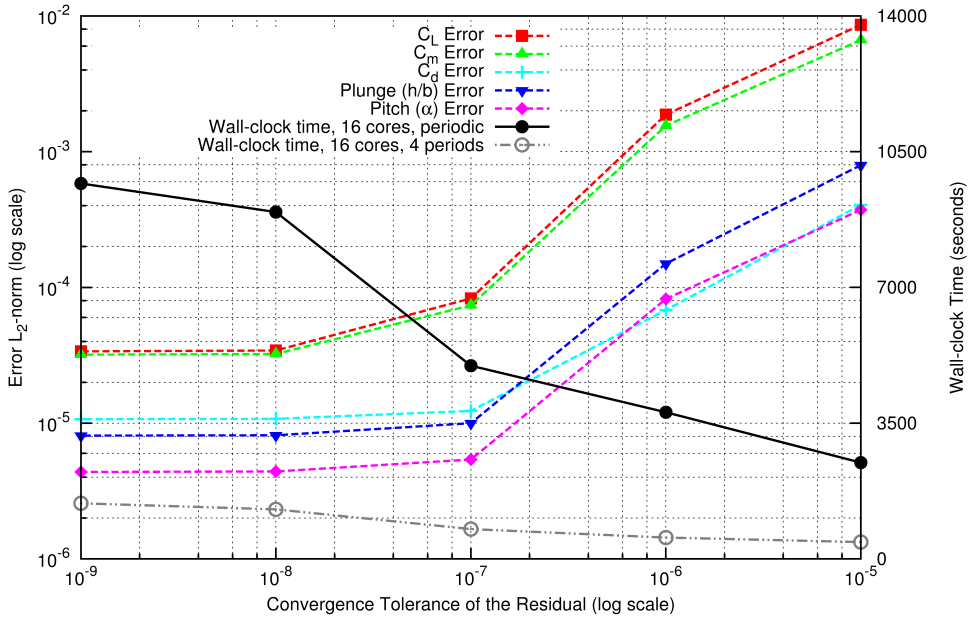


Figure 7.6: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (right axis) for the aeroelastic time-implicit method as a function of the convergence tolerance (512 time steps per period, $M_\infty = 0.755, k_c = 0.0814$)

solution with 63 time instances. As can be seen, relaxing the convergence tolerance to even the surprisingly high value of 1×10^{-4} allows the error in all five measures to increase only slightly, and it remains below the tolerances established above. Wall-clock time required is decreased from 5,665 to 1,696 seconds when using one core per time instance and from 22,306 to 6,678 seconds when scaled to 16 CPU cores. These results show that for engineering applications, less strict convergences tolerances can be used and the required precision still attained. However, it is surprising that the time-spectral tolerance can be relaxed so much more than the time-implicit tolerance. Perhaps this is a result of errors in the time-implicit solution accumulating over the thousands of time steps required to reach a purely periodic solution; whereas, no such accumulation can occur for the time-spectral solution.

Previously, it has been shown that both the freestream Mach number and reduced frequency of motion have a significant impact on the convergence of time-spectral methods for the flow alone cases. As such, it only follows to examine the impacts that both these variables have on the convergence time of aeroelastic, time-spectral methods. For all of the

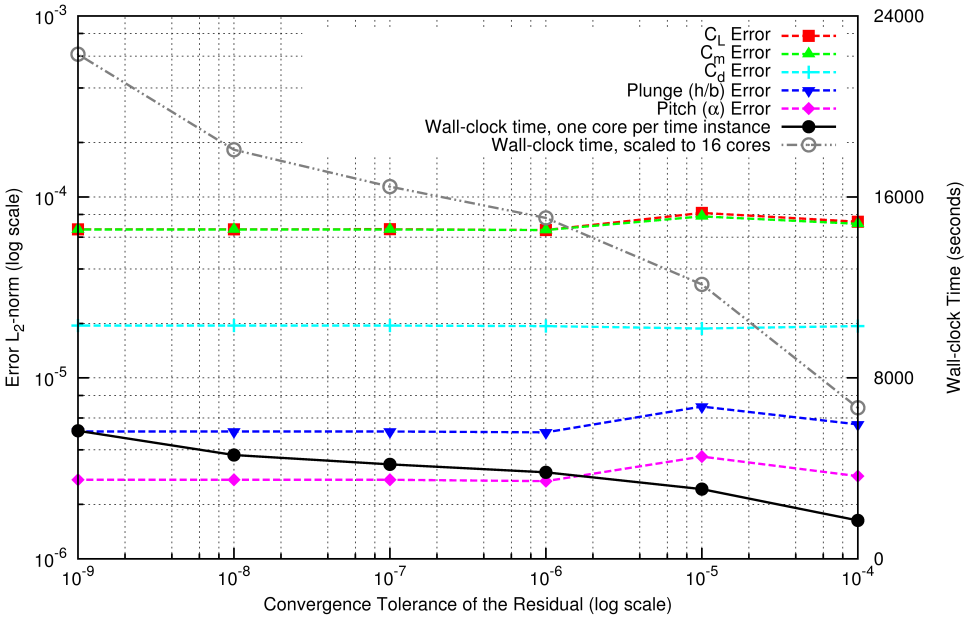


Figure 7.7: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (right axis) for the aeroelastic time-spectral method as a function of the convergence tolerance (63 time instances, $M_\infty = 0.755, k_c = 0.0814$)

cases presented in the next two subsections, the same precision limits that are used in the baseline case are used, i.e. differences of less than 2×10^{-4} for lift and moment coefficients and less than 2×10^{-5} for drag coefficient, pitch angle, and plunge height. Additionally, all the wall-clock times given in the subsequent plots use one computational core per time instance. Finally, for each of the alternate Mach number and reduced frequency cases, a reference time-implicit solution that has 4,096 time steps per period and is run for 100 periods is used to compute the RMS error in the time-spectral solutions. Also, the 1×10^{-9} convergence criterion is restored for all cases in the next two sections.

7.1.2 Convergence Variation with Freestream Mach Number

The aeroelastic test case run above is re-run with all of the same input parameters except that the freestream Mach number is varied. The new cases use freestream Mach numbers of 0.555, 0.705, and 0.805, or -0.20. -0.05, and +0.05 added to the baseline Mach number, respectively.

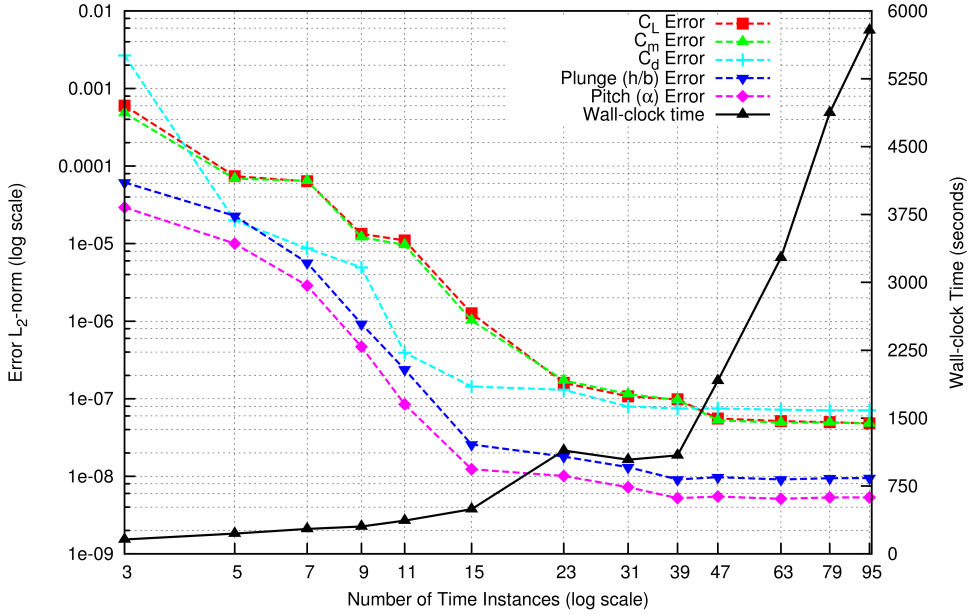


Figure 7.8: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (one core per time instance, right axis) for the aeroelastic time-spectral method as a function of the number of time instances ($M_\infty = 0.555, k_c = 0.0814$)

Figure (7.8) plots the error and the wall-clock time for the $M_\infty = 0.555$ solution. A precise solution in all measures can be found using only seven time instances for this case and requires only 276 seconds to compute. Exponential convergence is seen in plunge and pitch between three and nine time instances before the error convergence flattens. It is believed this error flattening occurs either because the 1×10^{-9} convergence criteria limits accuracy, which seems unlikely in light of the results given in the last section or because the accuracy of the time-spectral solutions has exceeded the accuracy of the reference, time-implicit solution. Further investigation is needed to make this determination.

Figures (7.9) and (7.10) are analogous to Figure (7.8) but for the $M_\infty = 0.705$ and $M_\infty = 0.805$ solutions, respectively. For $M_\infty = 0.705$, drag error limits the accuracy of the solution. A precise solution occurs for 47 time instances which takes $2,263$ seconds to compute on 47 cores. Exponential error convergence occurs for more than 31 time instances. Error in pitch and plunge flatten for more than 79 time instances, for one of the two reasons noted above. The $M_\infty = 0.805$ case is more difficult, with its strong shock waves; however,

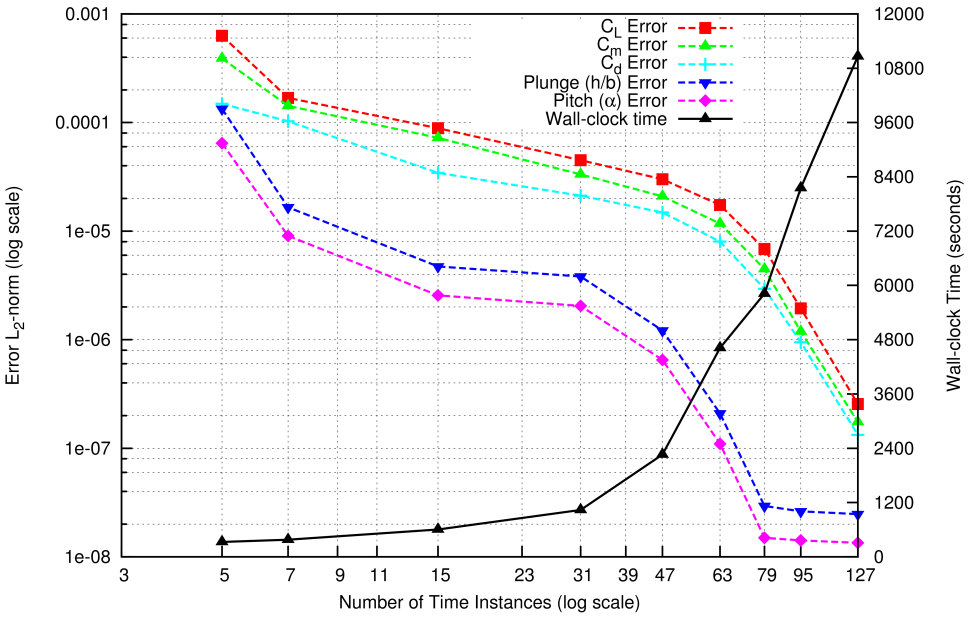


Figure 7.9: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (one core per time instance, right axis) for the aeroelastic time-spectral method as a function of the number of time instances ($M_\infty = 0.705, k_c = 0.0814$)

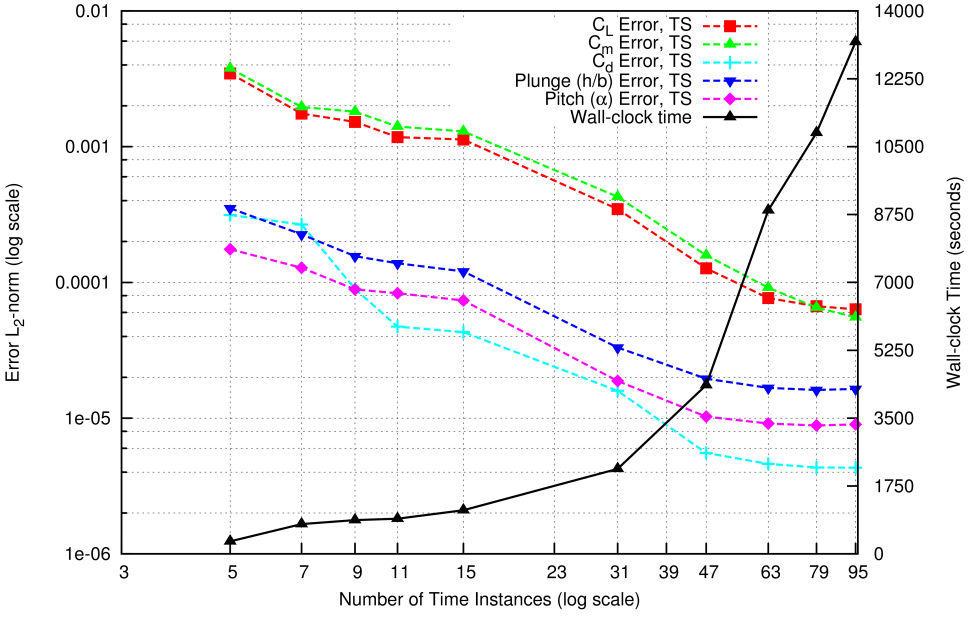


Figure 7.10: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (one core per time instance, right axis) for the aeroelastic time-spectral method as a function of the number of time instances ($M_\infty = 0.805, k_c = 0.0814$)

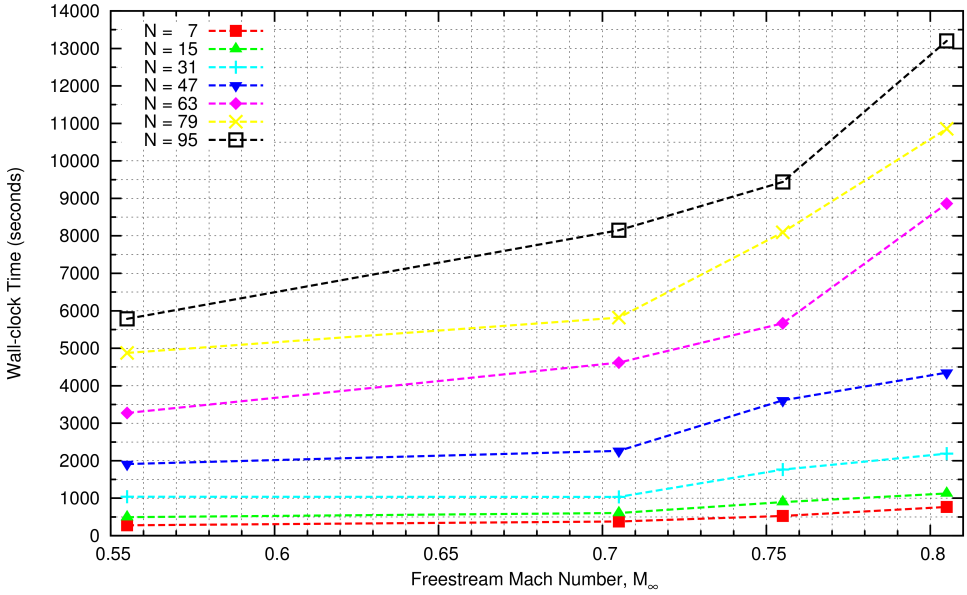


Figure 7.11: Wall-clock time to convergence (one core per time instance) versus freestream Mach number M_∞ for the aeroelastic time-spectral method with different numbers of time instances ($k_c = 0.0814$)

a precise solution can still be found using 47 time instances which requires $4,349$ seconds to converge. For this test case, exponential convergence is not seen; further, the error flattens while it is still quite large, possibly because of insufficient accuracy in the reference solution used to compute the error. It should be noted that although both of these solutions require 47 time instances, almost twice as much wall-clock time is needed for the $M_\infty = 0.805$ solution.

The manner in which the freestream Mach number affects convergence efficiency is now examined. Figure (7.11) plots convergence time versus Mach number for the three cases presented here and the baseline case. As can be seen, for a given number of time instances, convergence time grows at an accelerating rate as Mach number increases. Although more data points between Mach 0.555 and 0.705 are needed to confirm the following hypothesis, it appears that once shock waves begin occurring in the flow, which usually happens around $M_\infty = 0.7$, convergence time greatly increases.

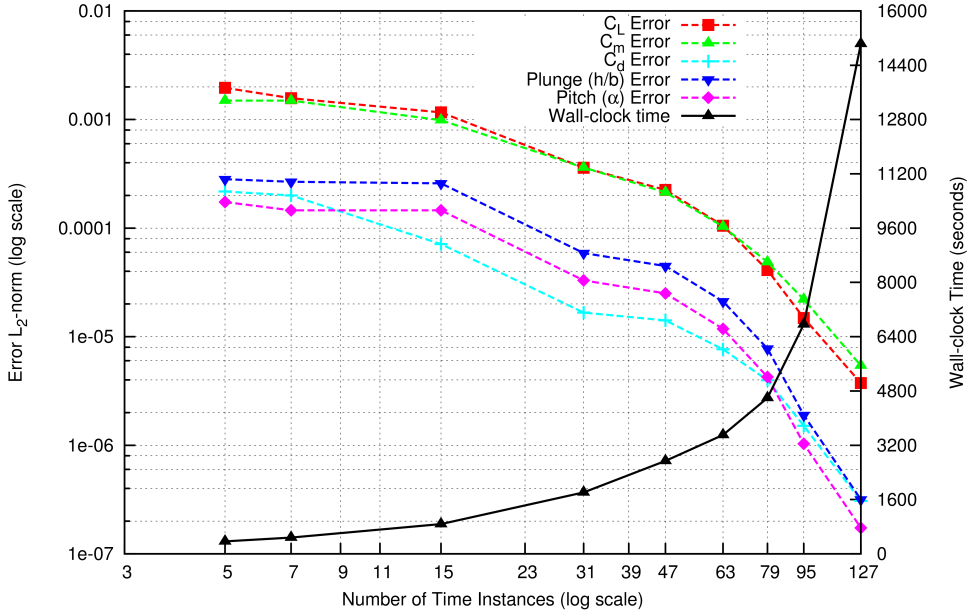


Figure 7.12: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (one core per time instance, right axis) for the aeroelastic time-spectral method as a function of the number of time instances ($M_\infty = 0.755, k_c = 0.0407$)

7.1.3 Convergence Variation with Reduced Frequency

To examine the effect of reduced frequency on convergence, the baseline, aeroelastic test case examined above is recomputed for one-half and two times the baseline reduced frequency, i.e. $k_c = 0.0407$ and $k_c = 0.1628$, respectively. Error plots for these two frequencies are given in Figures (7.12) and (7.13). In the $k_c = 0.0407$ case, plunge limits the precision of the solution and 79 time instances, which run in $4,598$ seconds, are needed. As in previous cases, exponential convergence begins to occur after 31 time instances. When $k_c = 0.1628$, even 95 time instances do not produce a precise solution in the lift coefficient, C_L. Because this case also uses, by far, the most wall-clock time to converge, solutions with even higher numbers of time instances are not attempted. It is noteworthy that exponential error convergence is not yet clearly present for this case; whether this is because of imprecision in the reference solution or because even more time instances are required is not known and is a subject for future examination.

As was the case with Mach number, reduced frequency has a great impact on convergence

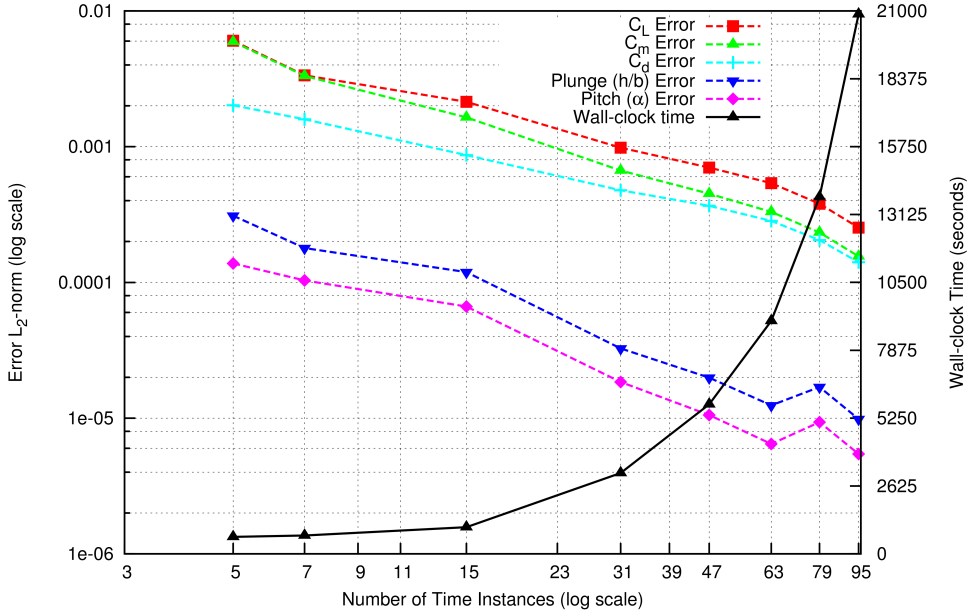


Figure 7.13: Error (compared to a reference time-implicit solution with 4,096 time steps per period, left axis) and wall-clock time to convergence (one core per time instance, right axis) for the aeroelastic time-spectral method as a function of the number of time instances ($M_\infty = 0.755, k_c = 0.1628$)

time and solver efficiency. Figure (7.14) plots the convergence time versus the reduced frequency for the three reduced frequencies presented in the present work. Unlike with Mach number, there is no clear frequency after which convergence time gets noticeably worse; however, increasing reduced frequency has a much more noticeable impact when the number of time instances is high, as evidenced by the slope of the $N = 79$ curve being so much greater than the slope of the $N = 47$ curve. As with Mach number, further investigation is necessary to understand how and why reduced frequency impacts convergence time in this way.

7.2 Other Aeroelastic Results

In the following sections, the wide variety of purely periodic problems to which the aeroelastic, time-spectral method can be applied is explored. First, two aeroelastic responses of an airfoil with the same prescribed pitching motion, but different flutter velocities are examined.

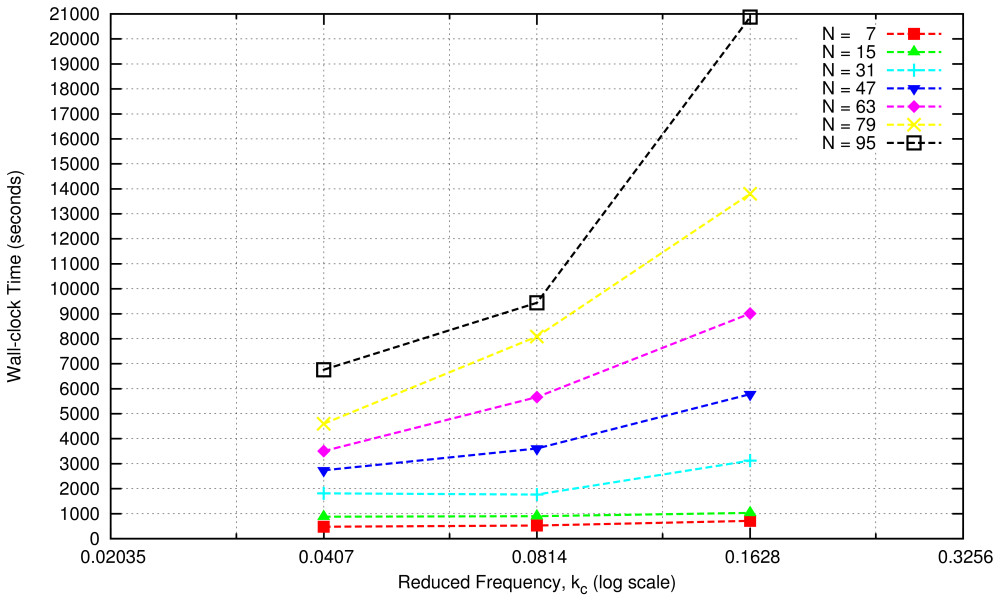


Figure 7.14: Wall-clock time to convergence (one core per time instance) versus reduced frequency k_c for the aeroelastic time-spectral method with different numbers of time instances ($M_\infty = 0.755$)

Next, the aeroelastic responses of an airfoil approximating a fixed wing to two different gusts are examined. Then, the two previous analyses are combined by examining the interactions of a pitching airfoil to the two different gusts profiles, in cases that approximate the responses of feathering and flapping helicopter rotor blades to gusts.

7.2.1 Rotary Wing Aeroelastic Response

To validate the capabilities of the time-spectral aeroelastic method to accommodate rotorcraft aeroelasticity problems, a two-dimensional test problem is devised. This simulation is analogous to taking a single section of a rotor blade, for example at 60-70% span, and simulating the flow around it independently of the other sections of the rotor. As such, a NACA-0012 airfoil is prescribed a cyclic input:

$$\theta_{con} = \theta_p + \theta_{cyc} \sin(\omega_\theta t). \tag{7.1}$$

The airfoil is then allowed to respond aeroelastically, both in pitch and plunge, to the aerodynamic forces produced. This can be thought of as a flapping blade (plunge response) with

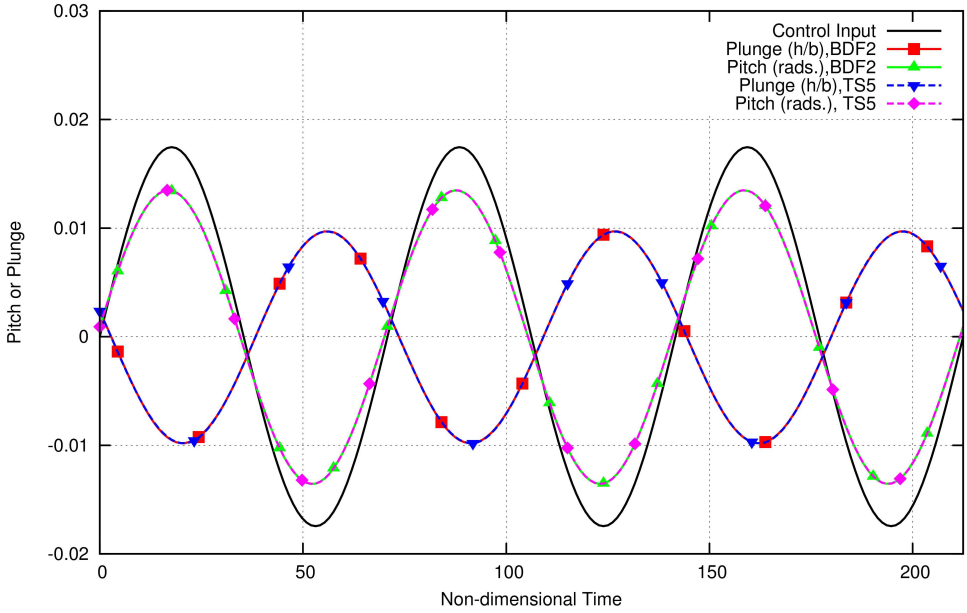


Figure 7.15: Comparison of rotary wing aeroelastic response for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.5$; 64 time steps and 5 time instances per period)

a pitch link of finite stiffness (aeroelastic pitch response). The same structural parameters as were used for the aeroelastic test case are used here as well. Additionally, the same computational mesh composed of 8,747 triangular elements, as shown in Figure (6.1), is used again. These structural parameters and this mesh are used for the remaining computations in this chapter.

Figures (7.15) and (7.16) plot both the control input and the the aeroelastic response, in pitch and plunge, of the airfoil to the control input: $\theta_p = 0^o$, $\theta_{cyc} = 1^o$, and $k_\theta = 0.05$, where k_θ is the reduced frequency of the control input, which is related to the angular frequency of control input by equation (6.2). For a rotor blade with an aspect ratio of 15, these control inputs would correspond to the airfoil section at 67% span. These cases and all remaining cases in this chapter are run at a freestream Mach number $M_\infty = 0.75$ for consistency. Both Figures (7.15) and (7.16) show good agreement between the time-implicit (using 64 time steps per period) and time-spectral results (using 5 and 15 time instances, respectively) and were run at the flutter velocities V_f as indicated in the figures, with higher flutter velocity corresponding to lower structural stiffness.

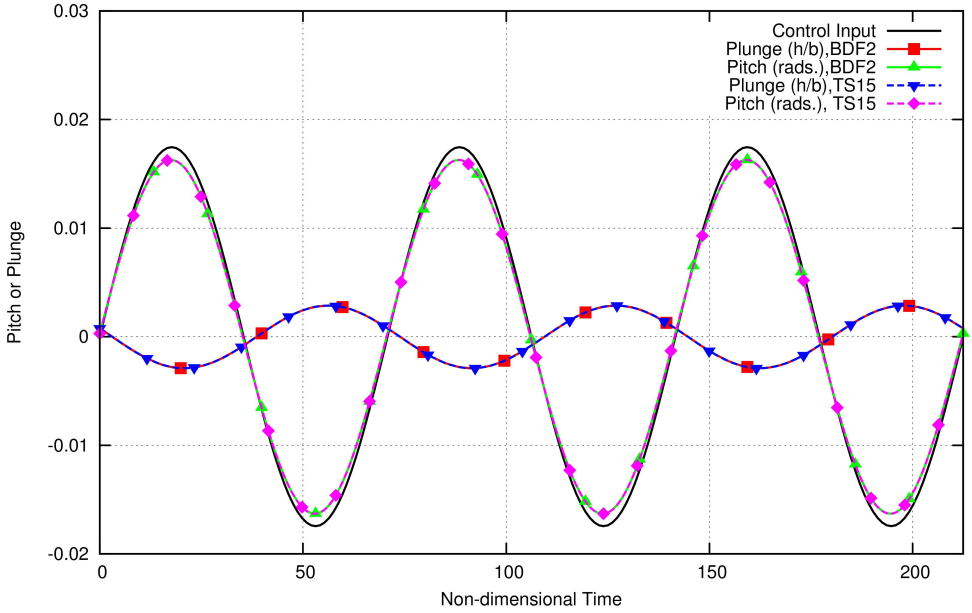


Figure 7.16: Comparison of rotary wing aeroelastic response for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 64 time steps and 15 time instances per period)

7.2.2 Gust Response

Gust responses for periodic gusts are presented in the following subsections. Two different gust profiles are used. Profile A is a short gust such that each gust is only 10% of the length between gust peaks. Profile B is a long gust whose length is fully half the length between gust peaks. Both gusts have a non-dimensional peak velocity $v_{g,max} = 0.0155$ which corresponds to an effective 1^o increase in the airfoil angle of attack. Both gust profiles repeat with a reduced frequency of $k_g = 0.05$ and have their leading edges initially positioned a full half period in front of the airfoil leading edge (31.4159 chords). Figure (7.17) presents a graphical representation of the two gust profiles just described. The two black bars represent the period of the gusts (peak to peak distance), while the collection of blue dots shows the location and chord of the airfoil in relation to the initial position and width of the gusts.

The following subsections compare time-spectral results to time-implicit results for these two gust profiles, first for a fixed wing and then for a simulated rotary-wing section with cyclic control input.

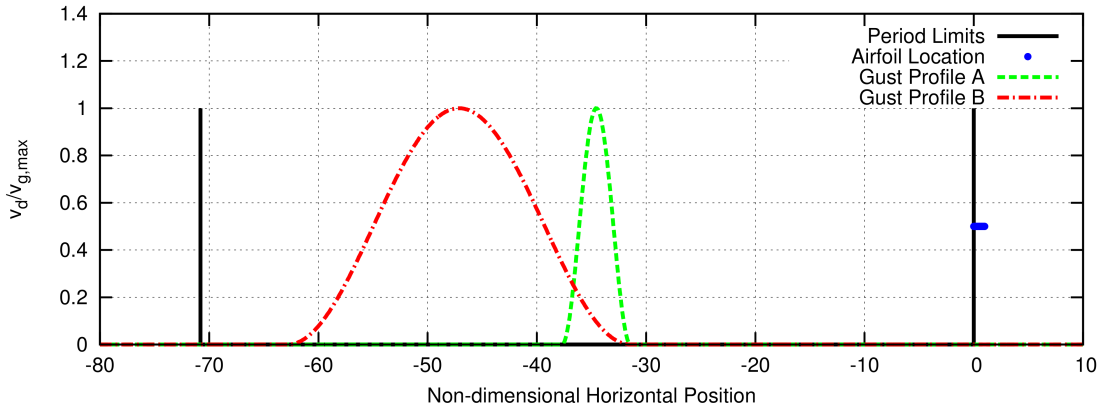


Figure 7.17: Periodic gust profiles at their initial positions in relation to the airfoil @$x = 0$ with its relative size shown by the small collection of dots on the right and the length of the spatial period of the gusts as shown by the two bars at either end

Fixed Wing Gust Response

In the pursuit of the aeroelastic solution of the fixed-wing response to the two gust profiles, the airfoil is first held rigidly at zero angle of attack and the forces resulting from the gust passing by the rigid airfoil are plotted. Figures (7.18) and (7.19) plot the lift and moment generated by gust profiles A and B, respectively. Figure (7.18) uses 55 time instances in order to achieve good agreement between time-spectral and time-implicit results, while only 15 time instances are required to produce good agreement for the longer gust B. For both gust profiles, 128 time steps per period are used for the time-implicit results.

Having obtained good agreement for the forces generated on the rigid airfoil, the same gust profiles are applied to the aeroelastic system. For both profiles, and, in fact, all results presented subsequently in this chapter, the flutter velocity is held constant at $V_f = 0.25$ which corresponds to a structural reduced frequency of $k_c = 0.50$ or ten times the reduced frequency of the gusts. Figures (7.20) and (7.21) plot the modified lift and moment experienced by the airfoil as it is allowed to respond aeroelastically. As can be seen, good agreement is obtained for the forces for both gust profiles. Likewise, Figures (7.22) and (7.23) relate the aeroelastic response to the two gust profiles. Again, good agreement is obtained between the time-implicit and time-spectral results. The preceding solutions use 1024 time steps and 45 time instances per period for gust A and 512 time steps and 27 time instances per period

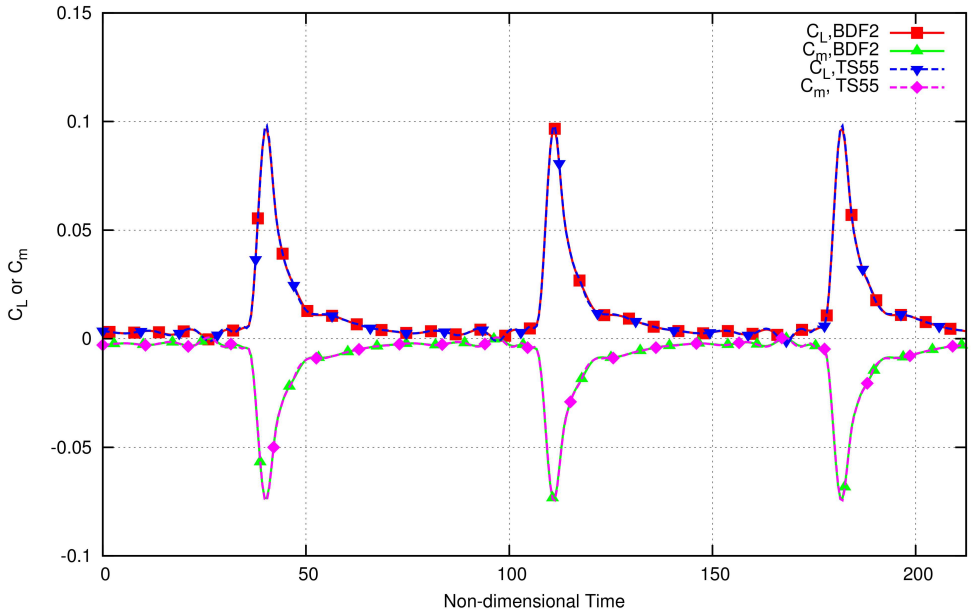


Figure 7.18: Comparison of static, fixed wing gust forces, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75$; 128 time steps and 55 time instances per period)

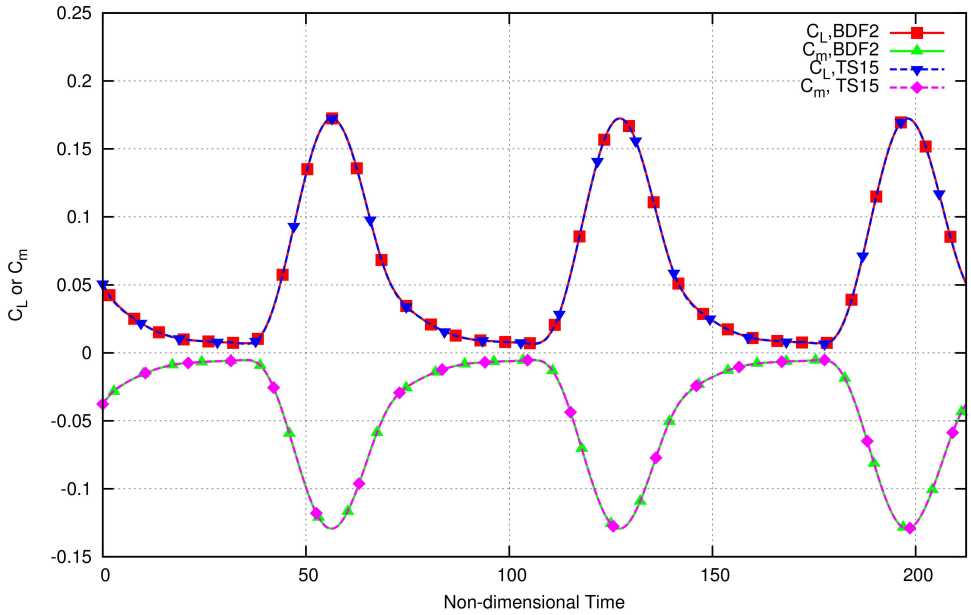


Figure 7.19: Comparison of static, fixed wing gust forces, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75$; 128 time steps and 15 time instances per period)

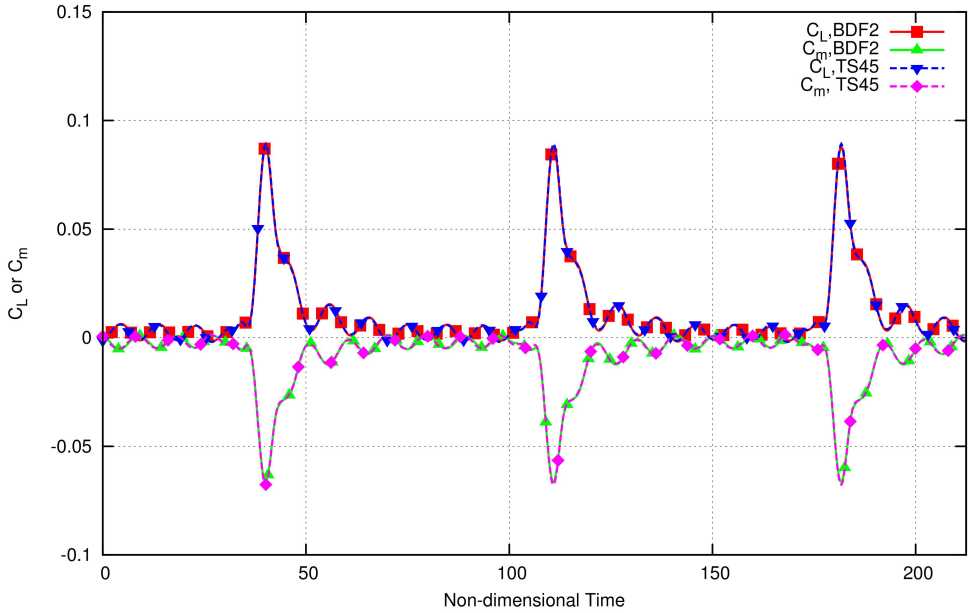


Figure 7.20: Comparison of aeroelastic, fixed wing gust forces, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 1,024 time steps and 45 time instances per period)

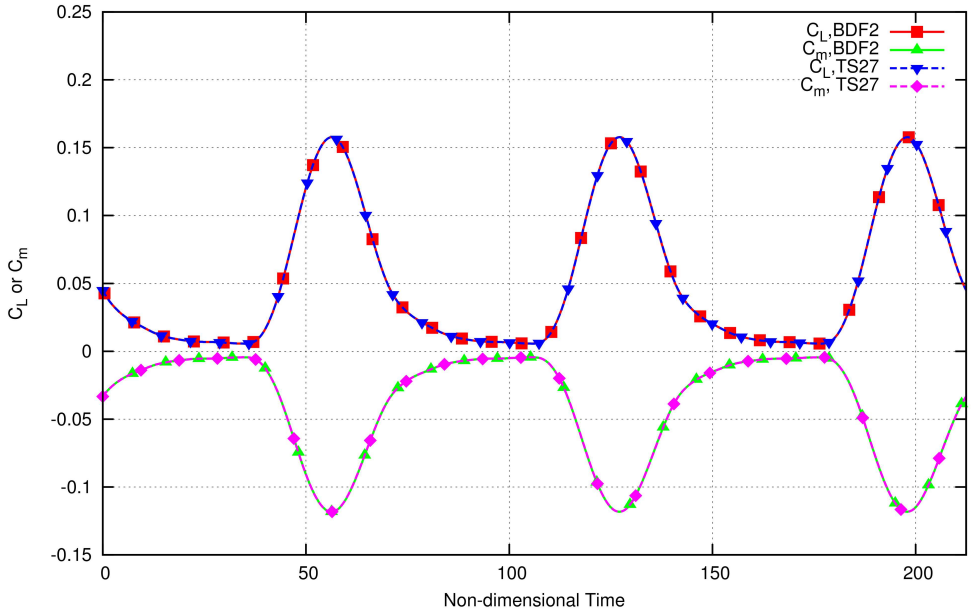


Figure 7.21: Comparison of aeroelastic, fixed wing gust forces, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 512 time steps and 27 time instances per period)

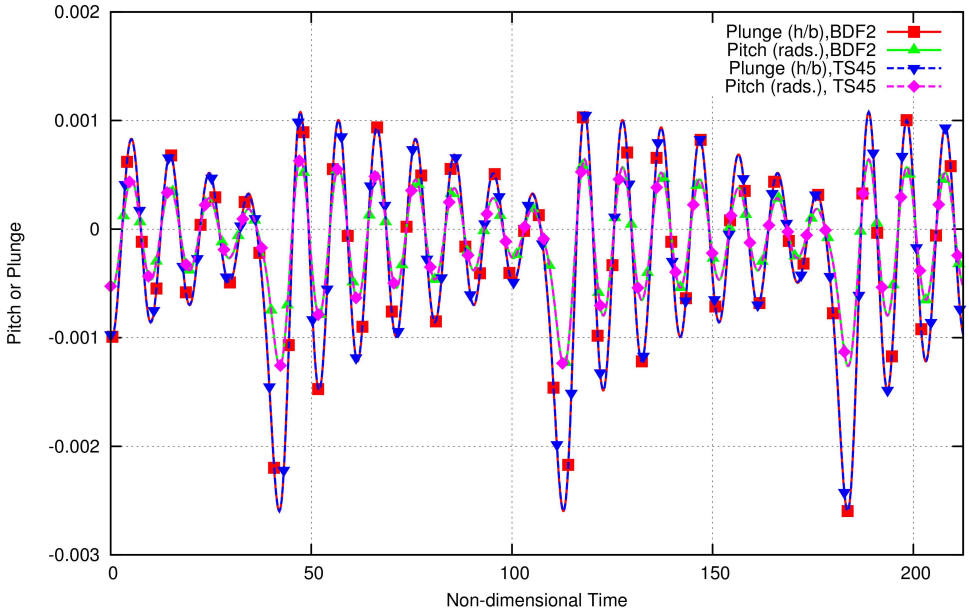


Figure 7.22: Comparison of aeroelastic, fixed wing gust response, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 1,024 time steps and 45 time instances per period)

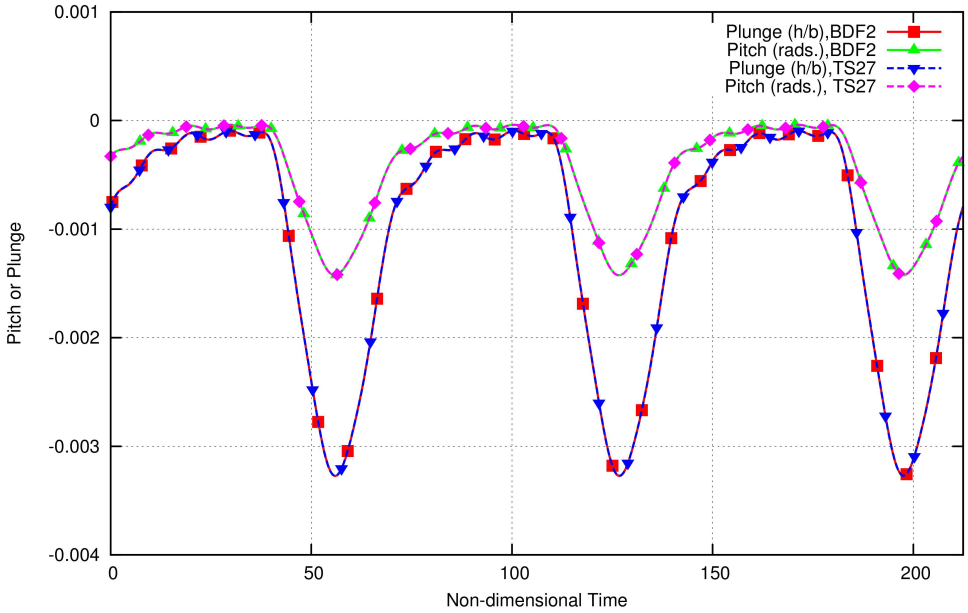


Figure 7.23: Comparison of aeroelastic, fixed wing gust response, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 512 time steps and 27 time instances per period)

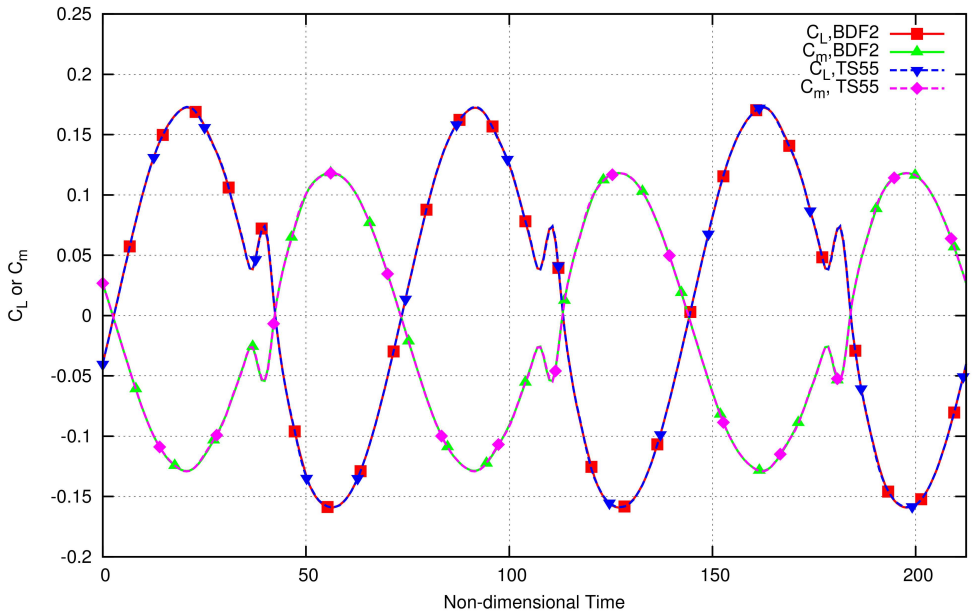


Figure 7.24: Comparison of static, rotary wing gust forces, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75$; 96 time steps and 55 time instances per period)

for gust B.

Rotary Wing Gust Response

The rotary wing gust response involves the NACA-0012 airfoil with the same prescribed control input as used in the "Rotary Wing Aeroelastic Response" subsection, above. It should be emphasized that the frequency of control input and the gust frequency are identical, indicating that the gust interacts with the blade once per revolution. First, the control input is coupled with a rigid airfoil and the resultant forces and moments are obtained. Figures (7.24) and (7.25) show the lift and pitching moment coefficients for gusts A and B, respectively. As can be seen, good agreement is obtained between the time-spectral and time-implicit results by using 55 time instances and 96 time steps per period for gust profile A and 15 time instances and 128 time steps per period for gust profile B.

With agreement of the force and moments obtained, we now remove the rigidity constraint and allow the airfoil to respond aeroelastically. As noted previously, a flutter velocity of $V_f = 0.25$ is used. Figures (7.26) and (7.27) plot the lift force and pitching moment

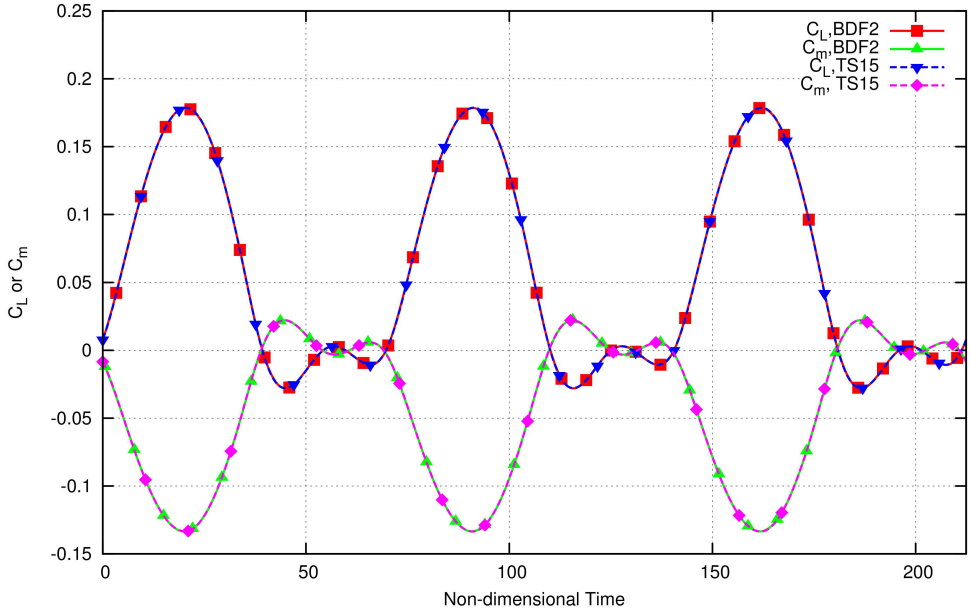


Figure 7.25: Comparison of static, rotary wing gust forces, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75$; 128 time steps and 15 time instances per period)

produced by the aeroelastically pitching airfoil. Similarly, Figures (7.28) and (7.29) plot the aeroelastic response of the same airfoil to the two gust profiles. Once again, good agreement between time-spectral and time-implicit results is obtained with the use of a sufficient number of time instances, as noted in the legends of these figures. Also note that the time-implicit results in both figures use 256 time steps per period.

As can be seen from the variety of problems presented, the aeroelastic, time-spectral method has wide applicability to periodic problems. Although not specifically examined, it should be apparent that both the split velocity method and the aeroelastic solver as presented in this thesis could be used to predict the more complicated aeroelastic interaction of an airfoil (or rotor blade) to periodically passing vortices. This problem, known in the rotorcraft community as blade-vortex interaction or BVI, is of particular interest as it represents a major source of rotorcraft noise. Unfortunately, although use of the split velocity method can mitigate some aspects of the numerical dissipation of vortices (specifically the dissipation of the velocity) it cannot address dissipation of the vortex density and energy. As such, to properly analyze such a problem, the implementation of an adaptive grid strategy is necessary. Nevertheless, with the inclusion of this one additional component, the aeroelastic

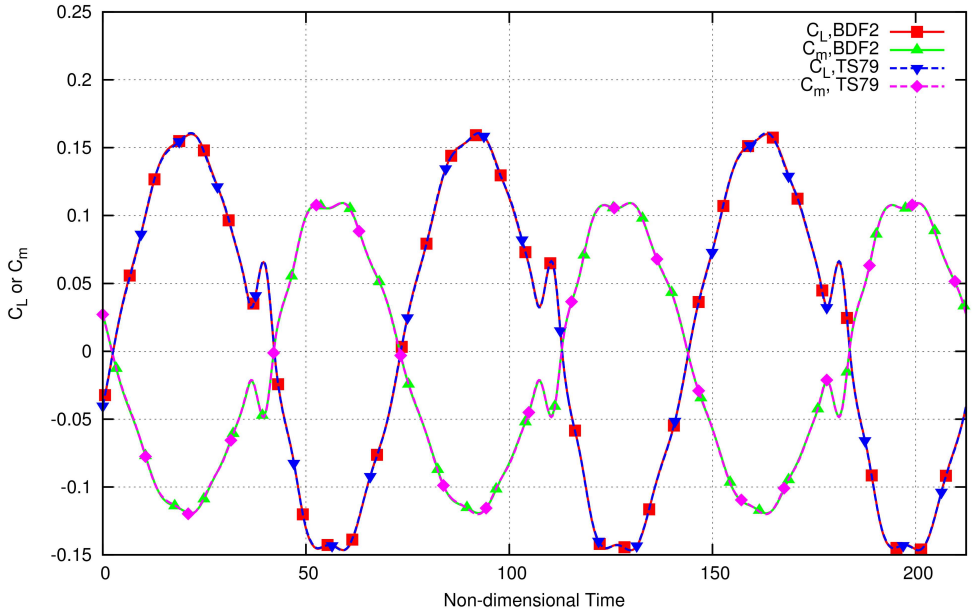


Figure 7.26: Comparison of aeroelastic, rotary wing gust forces, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 256 time steps and 79 time instances per period)

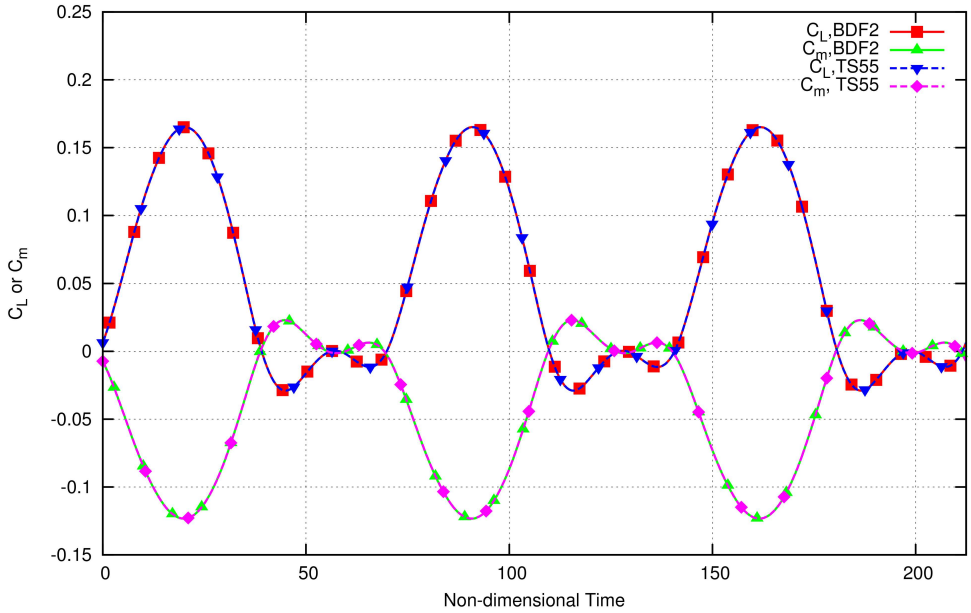


Figure 7.27: Comparison of aeroelastic, rotary wing gust forces, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 256 time steps and 55 time instances per period)

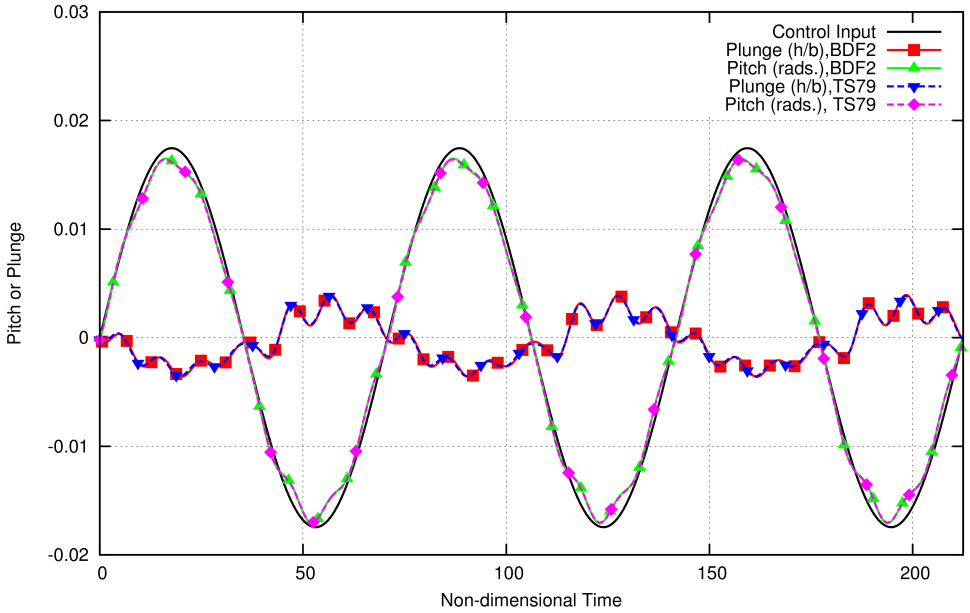


Figure 7.28: Comparison of aeroelastic, rotary wing gust response, gust profile A (short gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 256 time steps and 79 time instances per period)

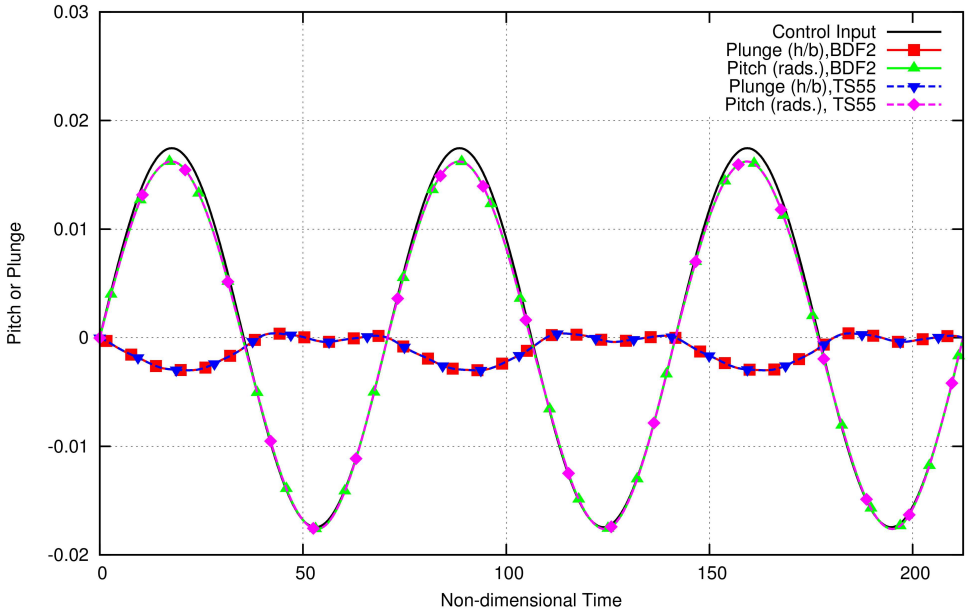


Figure 7.29: Comparison of aeroelastic, rotary wing gust response, gust profile B (long gust), for BDF2 and TS methods ($M_\infty = 0.75, V_f = 0.25$; 256 time steps and 55 time instances per period)

time-spectral method would have the potential to analyze BVI problems to a high degree of accuracy and perhaps more efficiently than time-implicit methods.

Chapter 8

BDFTS Flutter Results

The BDFTS flow solver is combined with the aeroelastic structural equations discretized with the BDFTS approach to form a BDFTS aeroelastic solver. The solution process for the BDFTS implementation is as follows: first a purely periodic time-spectral solution is found at the actual frequency of flutter. Then, a single period of first-order BDFTS is solved as the airfoil is allowed to respond aeroelastically. Next, enough additional periods are run using second-order BDFTS to determine if the solution is aeroelastically damped, neutral, or excited (i.e. experiencing flutter). It should be noted that the period of flutter is assumed to be known and prescribed for the BDFTS solutions.

To validate the BDFTS aeroelastic solver the two-dimensional swept wing model exhibiting the transonic dip phenomenon suggested by Isogai [79] is chosen as the test case. The structural parameters for this case are:

$$\begin{aligned} x_\alpha &= 1.8 \\ r_\alpha^2 &= 3.48 \\ \frac{\omega_h}{\omega_\alpha} &= 1.0 \\ \mu &= 60 \\ a &= \text{-}2.0. \end{aligned}$$

The quantity a is the non-dimensional elastic axis location along the chord of the airfoil measured from the midchord of the airfoil when it is in the neutral position. Since it is non-dimensionalized by the semi-chord of the airfoil, the elastic axis is located half a chord

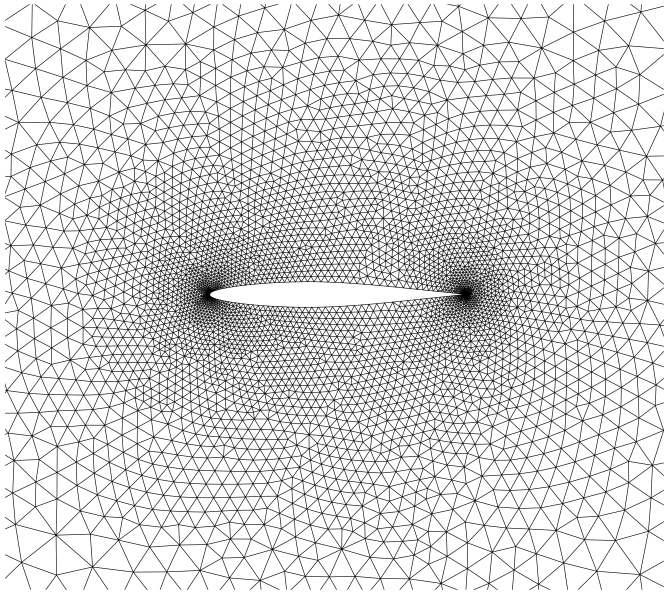


Figure 8.1: Near-body computational mesh for the NACA-64A010 airfoil (4821 nodes, 9455 elements)

length ahead of the leading edge of the airfoil in this particular case. The airfoil under consideration is the NACA-64A010 (Ames) airfoil operating with a mean angle-of-attack α_0 of 0^o and an amplitude of forced pitching α_{max} of 1^o. A computational mesh composed of 9,455 triangular elements is used, as shown in Figure (8.1). At two separate Mach numbers, damped, neutral, and excited cases are run and compared to the corresponding time-implicit solutions. The results for a freestream Mach number $M_\infty = 0.875$ are given in Figures (8.2-8.4). It should be noted that the BDF2 solutions used for comparison were rerun using the actual flutter frequency for the first three periods of prescribed motion to facilitate direct comparison of the data. As can be seen from these figures, there is close agreement between the time-implicit and time-spectral solutions, indicating that the BDFTS method is capable of solving the coupled fluid/structure equations. Most importantly, the solutions exhibit the correct flutter response, i.e. damped, neutral, or excited. As has already been noted, the cases presented in Figures (8.2-8.4) were all run at a freestream Mach number $M_\infty = 0.875$, using five time-spectral instances, and at the flutter velocity indicated by the figure caption. In addition, the BDF2 solutions for the damped, neutral, and excited cases used 64, 128, and 64 time steps per period, respectively. It should also be noted that, although BDFTS and

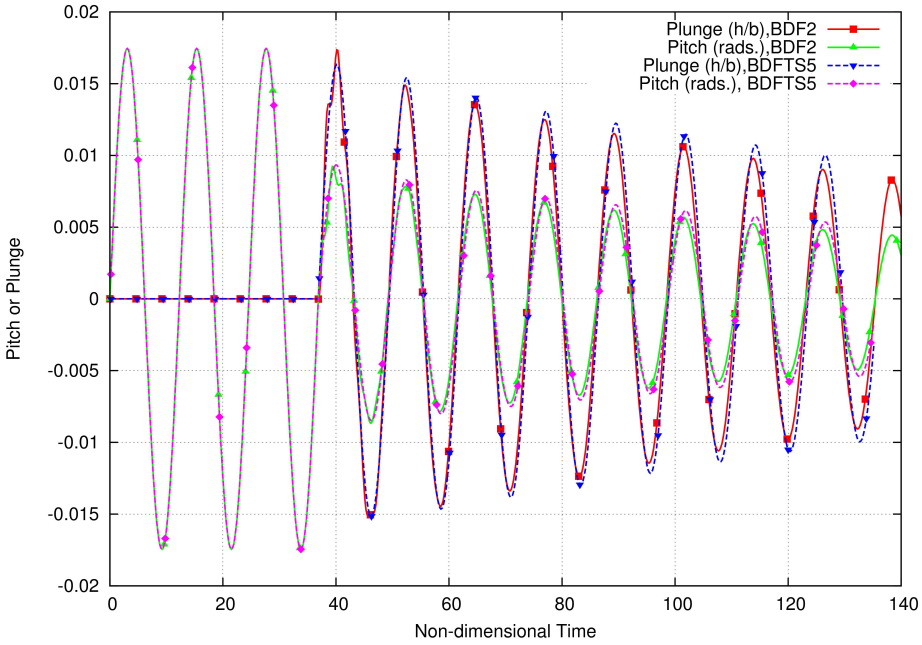


Figure 8.2: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Damped response ($M_\infty = 0.875, V_f = 0.4$; 64 time steps and 5 time instances per period)

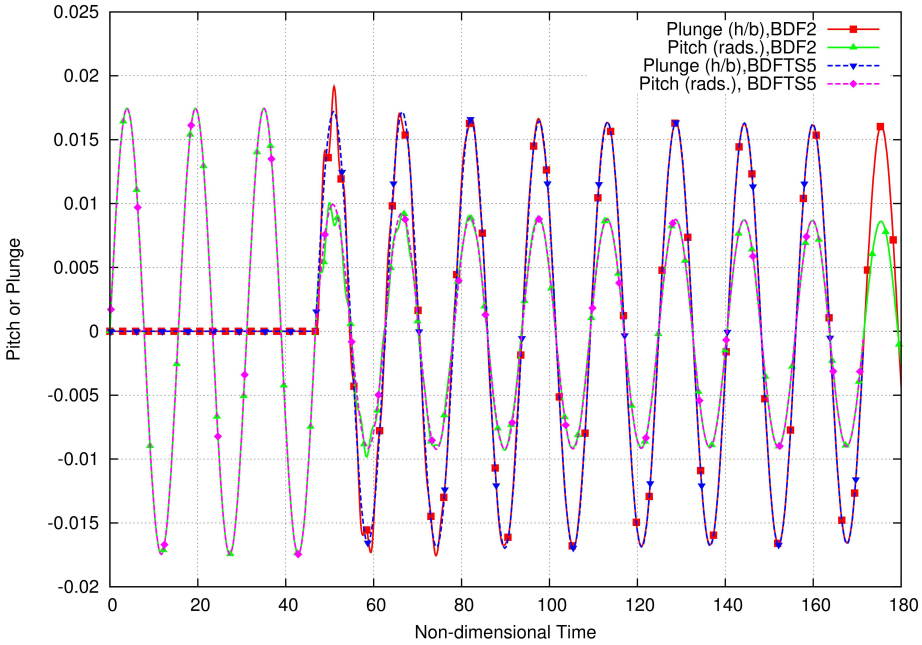


Figure 8.3: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Neutral response ($M_\infty = 0.875, V_f = 0.537$; 128 time steps and 5 time instances per period)

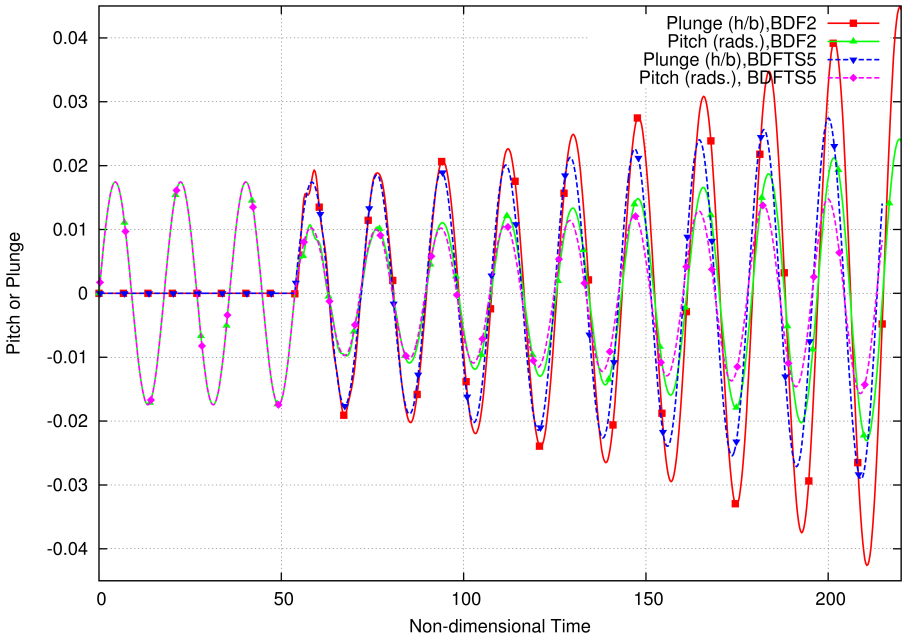


Figure 8.4: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Excited response ($M_\infty = 0.875, V_f = 0.65$; 64 time steps and 5 time instances per period)

BDF2 response plots match almost exactly for the neutral response, the BDFTS response decays more slowly in the damped case and grows more slowly in the excited case. This discrepancy is not of paramount importance as flutter analysis simply seeks to find the point of neutral response, which does show good agreement; however, the addition or more time instances brings the damped and excited cases into better agreement as can be seen subsequently. The trade-off comes in the form of better solution accuracy at the cost of increased run-time. If the the flutter velocity of the neutral response is the only quantity being sought, using fewer time-instances to produce a faster solution seems like the better choice.

Figures (8.5-8.7) chart a damped, neutral, and excited airfoil response at a freestream Mach number $M_\infty = 0.75$, using fifteen time instances for the damped and excited cases and seven time instances for the neutral case. Also, the BDF2 results use 96, 128, and 128 time steps per period for the damped, neutral, and excited cases, respectively. As previously, the corresponding flutter velocity is noted in the figure caption. For the $M_\infty = 0.875$ case, it

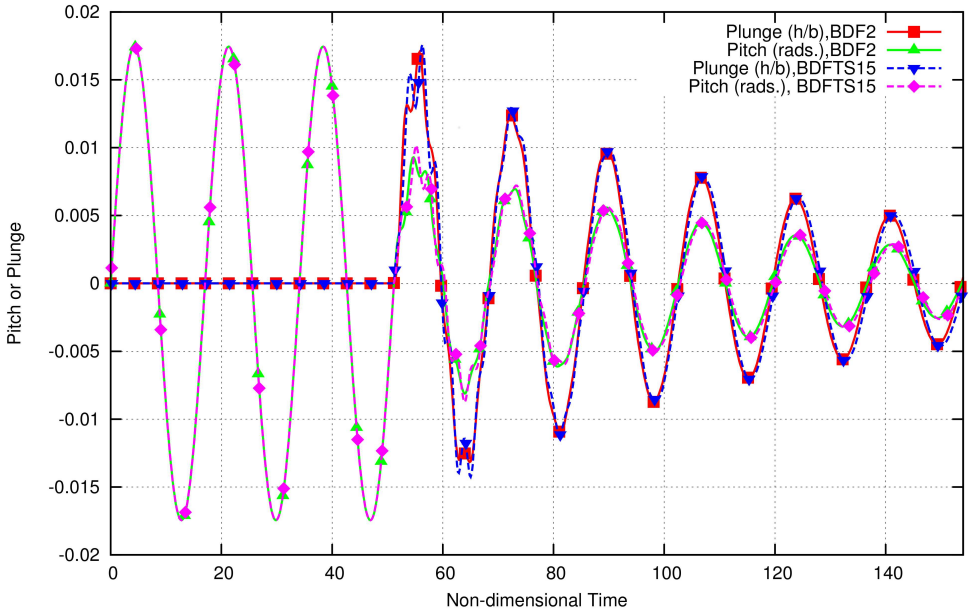


Figure 8.5: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Damped response ($M_\infty = 0.75, V_f = 0.5$; 96 time steps and 15 time instances per period)

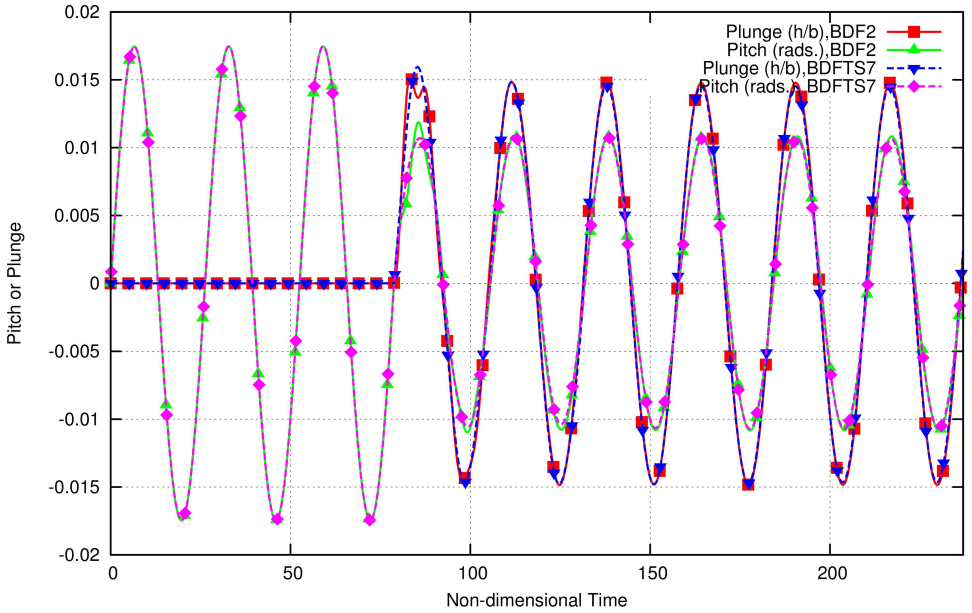


Figure 8.6: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Neutral response ($M_\infty = 0.75, V_f = 1.116$; 128 time steps and 7 time instances per period)

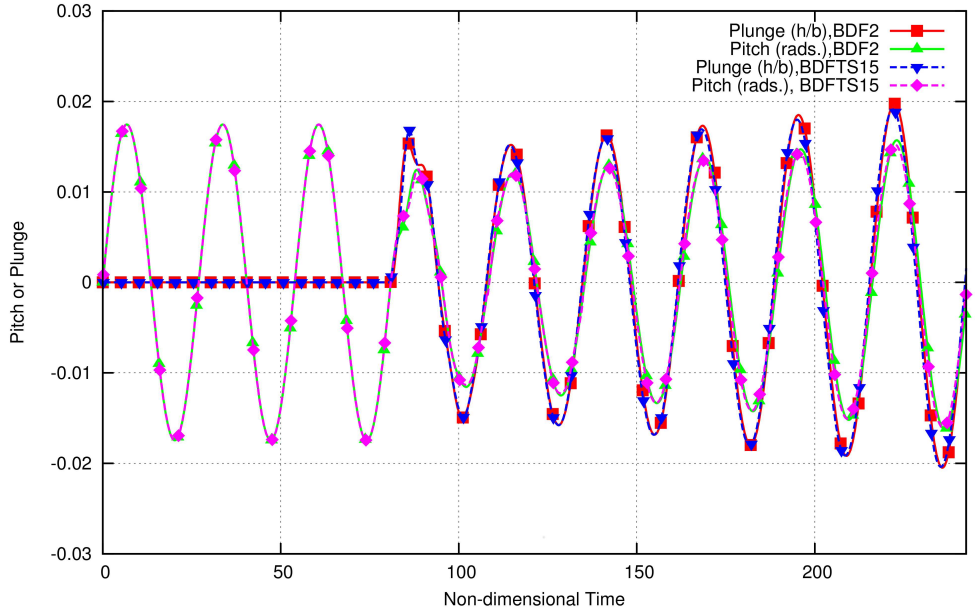


Figure 8.7: Comparison of the fixed-wing aeroelastic flutter response for BDF2 and BDFTS methods: Excited response ($M_\infty = 0.75, V_f = 1.2$; 128 time steps and 15 time instances per period)

was noted that, although BDFTS and BDF2 response plots match almost exactly for the neutral response, the BDFTS response decayed more slowly in the damped case and grew more slowly in the excited case. This discrepancy is not observed in the $M_\infty = 0.75$ case as just presented. As noted previously, the use of more time-spectral time-instances per BDFTS period explains the better agreement.

The flutter velocity of the neutral response for the $M_\infty = 0.875$ case was found to be $V_f = 0.537$. Similarly, when $M_\infty = 0.75$ a flutter velocity of $V_f = 1.116$ produces a neutral response. Both of these data points correspond well with the flutter boundary plot given in Figure (8.8). The BDFTS method has been demonstrated to be capable of solving the coupled unsteady fluid-structure equations and results compare favorably with the standard time-implicit solutions. However, further improvements are needed in order for this method to solve flutter problems without any prior knowledge about the airfoil response characteristics. Specifically, a method must be developed that allows for the resolution of the flutter period as the BDFTS aeroelastic solution progresses. In addition, once a method to find the flutter frequency is developed, timing tests must be run to demonstrate how the

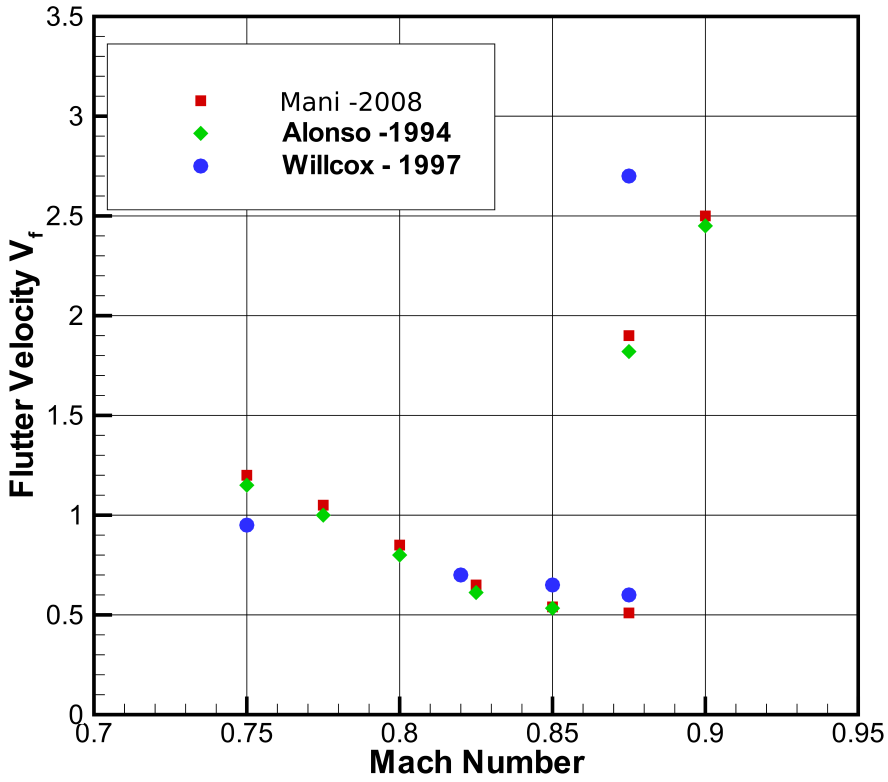


Figure 8.8: Flutter boundary from other references [1–3]

efficiency of solutions of flutter problems compares when using the BDFTS discretization versus traditional time-implicit methods.

Chapter 9

SEMT Results

The recently implemented SEMT time-discretization using backward-only coupling is validated first by solving an ordinary differential equation and then by examining a full solution of the SEMT Euler equations. In some cases, the expected theoretical error convergence behavior is observed; however, in other cases it is not. Attempts are made to explain the causes of less than theoretically possible error convergence. While monolithic-time and periodic SEMT coupling have been discussed previously in Chapter 2, they have not yet been implemented and no results are shown for them in this thesis.

9.1 Validation of the SEMT Discretization

To validate the correct implementation of the backward-only SEMT, the following ordinary differential equation is solved using the SEMT:

$$\frac{dU}{dt} = U\Big(\cos(t) - b\Big) \tag{9.1}$$

where U is the scalar solution variable and b is a constant to be specified. The exact solution of this ODE has the following form:

$$U(t) = e^{\Big(\sin(t) - bt\Big)} + c_0 \tag{9.2}$$

where c_0 is an integration constant whose value depends in the initial condition. In the case presented herein, $b = {}^1\!/_{100}$ and the initial condition is chosen such that $U(0) = 1$ which

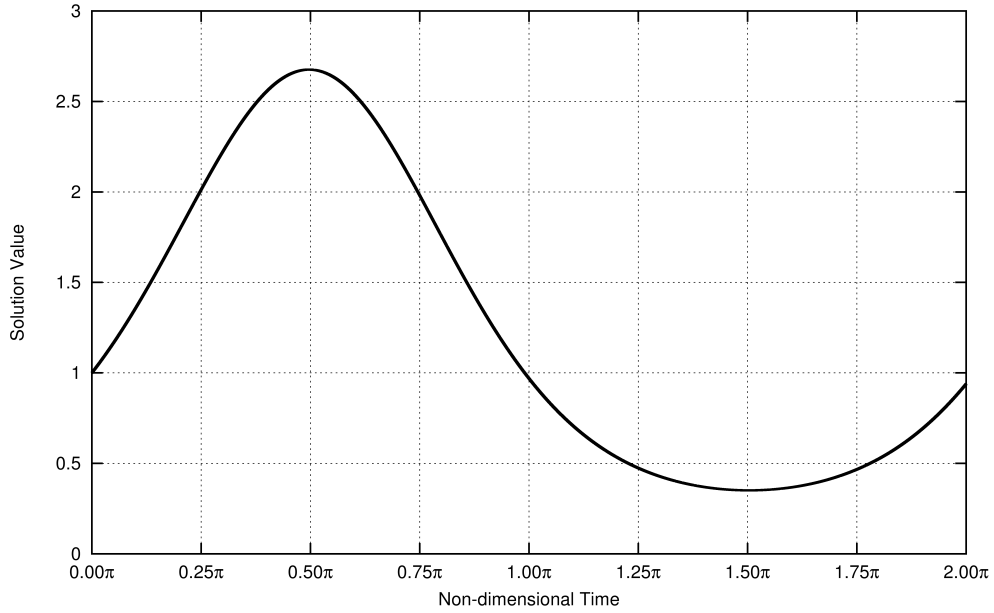


Figure 9.1: Exact solution of the ODE in equation (9.1) with $b = 1/100$ and $U(0) = 1$

corresponds to $c_0 = 0$. This solution contains strong periodic content with a period of 2π and a decaying transient that asymptotically approaches zero as time approaches infinity, i.e. $\lim_{t\to\infty} U(t) = 0$, as shown in Figure (9.1) for one period. Equation (9.1) is solved using the exact same solver as the has been used previously for the flow equations with the following modifications to allow an ODE to be solved using the already established framework of the code:

1. A 4×4 grid of squares with edge length 1 is used.
2. The residual at each grid cell center is redefined according to the ODE used.
3. Preconditioning in FGMRES is turned off so that the Jacobian need not have invertible diagonal blocks.
4. The CFL number is set to a large number (larger than the inverse of machine zero) such that the diagonal term, which contains the inverse of the pseudo-time step, will be less than machine zero.

Figure (9.2) shows the $L_2 - norm$ of the error when the exact solution is compared to the analytical solution within the period $[0 : 2\pi]$. Each curve corresponds to a different

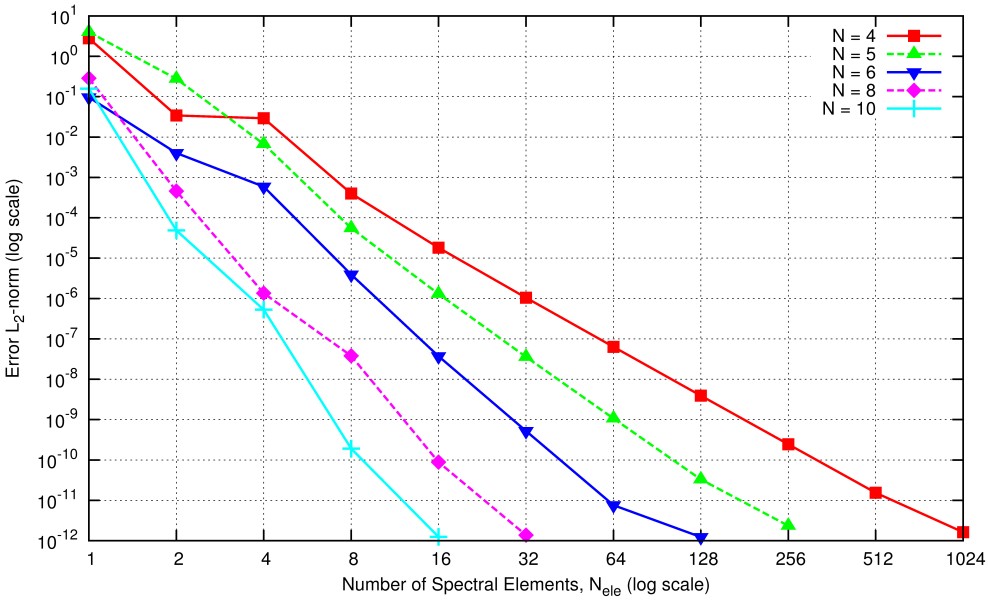


Figure 9.2: $L_2 - norm$ of the SEMT solution error for the ODE in equation (9.1) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

polynomial order within each time element as given in the legend, while the x-axis gives the number of elements into which the specified time range is discretized. If the solution is sufficiently smooth, the slope of these lines should be the same as, if not greater than, the degree of the polynomial used within the elements [80]. Indeed, for $N = 4$ the slope is 4.10 between $N_{ele} = 8$ and $N_{ele} = 512$. For $N = 5$ the slope is 5.52 between $N_{ele} = 4$ and $N_{ele} = 128$. For $N = 6$ the slope is 6.56 between $N_{ele} = 4$ and $N_{ele} = 64$. For $N = 8$ the slope is 7.89 between $N_{ele} = 1$ and $N_{ele} = 16$. For $N = 10$ the slope is 9.88 between $N_{ele} = 1$ and $N_{ele} = 8$. The slopes are calculated over what appears to be the longest linear region of the error convergence curve and neglect regions that appear to either have too few elements to show the proper slope, such as between $N_{ele} = 1$ and $N_{ele} = 8$ for $N = 4$, and regions where the lower bound of machine zero seems to have affected the slope. The good agreement of these slopes with the theoretically possible values gives confidence that the SEMT discretization is implemented correctly. Figure (9.3) plots error versus polynomial degree for $N_{ele} = 4$ and $N_{ele} = 8$. As can be seen, except for between $N = 8$ and $N = 10$ for $N_{ele} = 4$, the expected exponential convergence of a spectral method is exhibited.

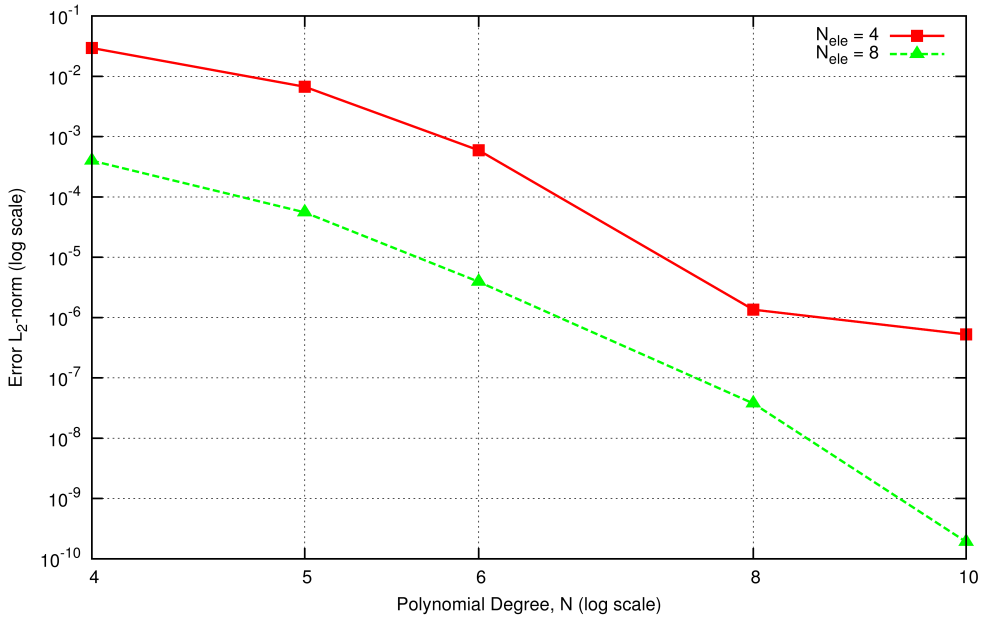


Figure 9.3: $L_2 - norm$ of the SEMT solution error for the ODE in equation (9.1) with the given polynomial degree (x-axis) and the specified number of spectral elements in time

9.2 Full SEMT Solution of the Euler Equations

With the implementation of the SEMT having been verified in the last section, it now follows that a full solution of the time-accurate Euler equations will be found. The flow problem chosen uses the same NACA-0012 mesh as was used for the time-spectral solutions as given in Figure (6.1). A test case with a free stream Mach number $M_\infty = 0.25$ was chosen to ensure no shock waves, which could be possible sources of error, were present. The airfoil motion itself is also rather simple and is given by the following equation:

$$\alpha(t) = \begin{cases} 0 & t \leq 0.05T \\ \frac{\alpha_A}{2}\left[1 - \cos\left(\frac{\omega t - 0.05T}{0.9}\right)\right] & 0.05T < t < 0.95T \\ 0 & t \geq 0.95T \end{cases} \tag{9.3}$$

where T is the total time frame (or period) of the solution, ω is the angular frequency that corresponds to that period, and $\alpha_A = 1.0^o$ is the maximum pitch angle attained. The reduced frequency of the motion $k_c = 0.001$ is very low so that the solution will be "quasi-steady," i.e. lift will vary linearly with angle of attack since unsteady effects are negligible. This reduced

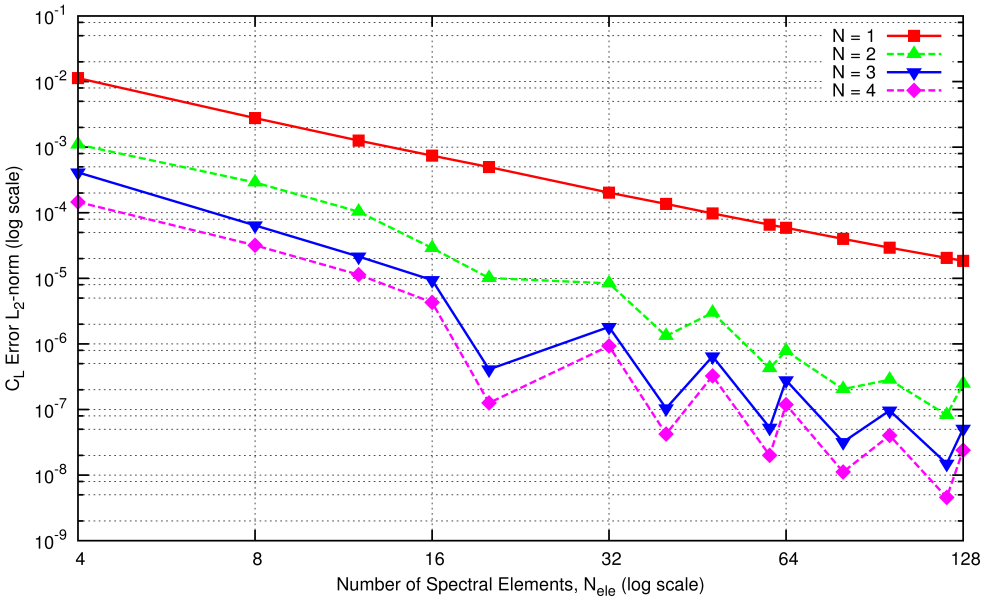


Figure 9.4: $L_2 - norm$ of the SEMT solution C_L error for the Euler equations with the airfoil motion prescribed in equation (9.3) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

frequency corresponds to a period length of approximately $T = 10620.5$ non-dimensional seconds.

Figure (9.4) plots the $L_2 - norm$ of the C_L error for SEMT solutions, when compared to a reference BDF2 solution using 322,560 time steps per period, with the number of spectral elements used N_{ele} on the x-axis. It should be noted that when this RMS error is calculated, the SEMT solutions are reconstructed such that the C_L can be calculated at exactly the same 322,560 points in time as are used in the reference BDF2 solution. SEMT solutions using 1st, 2nd, 3rd, and 4th degree polynomials within the elements are shown as given in the legend. When the error convergence slopes are calculated between $N_{ele} = 4$ and $N_{ele} = 128$ for these four curves, they are found to be 1.85 for $N = 1$, 2.42 for $N = 2$, 2.60 for $N = 3$, and 2.51 for $N = 4$. Although the first-order and second-order accurate solutions, exhibit the expected slope, the higher order solutions do not. The saw-tooth pattern of the 2nd and higher-order accurate solutions is also noteworthy. All of the relative valleys in the error occur, not coincidentally, when N_{ele} is a multiple of 20. Taking into consideration the motion of the airfoil given by equation (9.3), it is apparent that when N_{ele} is a multiple of 20, the

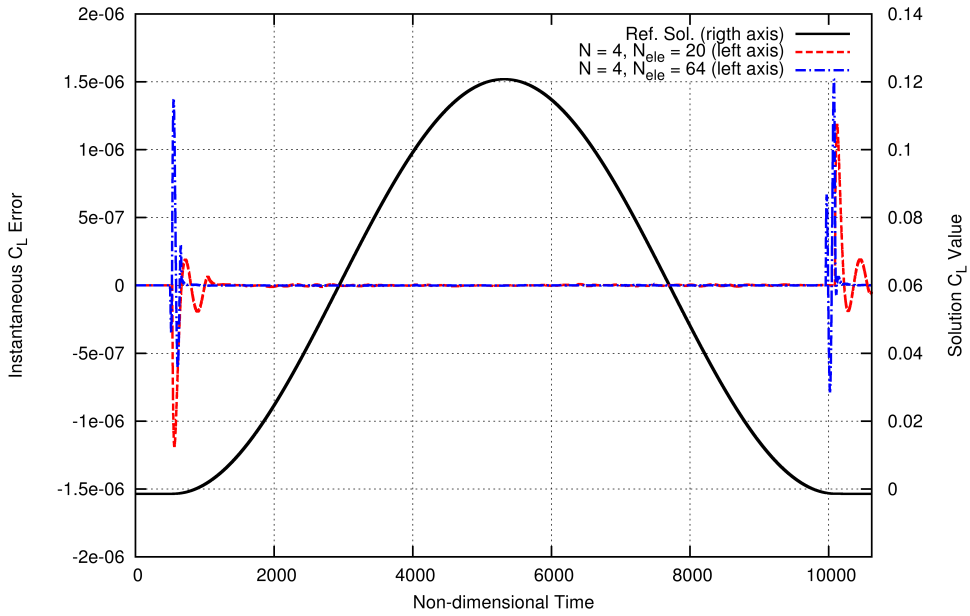


Figure 9.5: Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (9.3) and specified polynomial degree and number of spectral elements in time

endpoint of an element lies exactly on the point at which the airfoil begins its motion and also exactly on the point at which the airfoil returns to zero angle of attack.

Figure (9.5) plots the reference solution in solid black on the right axis and the instantaneous C_L error for the $N_{ele} = 20$ and $N_{ele} = 64$ 4th-order solutions, which have about the same C_L error $L_2 - norm$ over the time frame. As can be seen, for both solutions, the C_L error is concentrated around the points at which the airfoil begins and ends its motion. In addition, the magnitude of these oscillations is larger when an element spans this transition ($N_{ele} = 64$), than when the endpoint of an element and this transition coincide ($N_{ele} = 20$). Figure (9.6) plots exactly the same quantities as the previous plot, except on a scale one-hundred times finer so that the details of the error between the transition regions can be seen. This figure shows that if the transition regions were neglected, the $L_2 - norm$ of the C_L error for $N_{ele} = 64$ would indeed be considerably lower than it would be for the $N_{ele} = 20$ solution. In other words, the non-ideal error convergence observed for this case seems to result almost entirely from the transition regions.

In hindsight, this result should have been expected: the motion as defined by equation

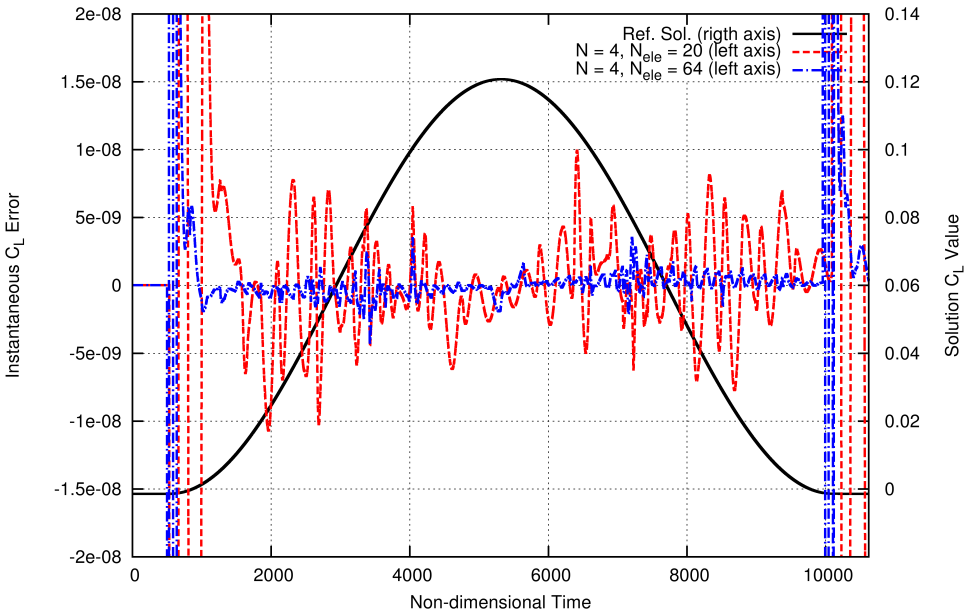


Figure 9.6: Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (9.3) and specified polynomial degree and number of spectral elements in time

(9.3) is continuous in pitch angle and its first derivative; however, discontinuities exist in the second and higher derivatives. On the other hand, the polynomial basis is continuous in all derivatives (although all derivatives higher than the degree of the polynomial are identically zero). Because of these facts, it seems intuitive that having such a polynomial span a second-derivative discontinuity would causes oscillations in that polynomial. When the SEMT discretization is applied to first-order partial differential equations, like the Euler equations, derivative discontinuities are allowed at the end points of the elements; hence, when the end point of an element coincides with the point at which the discontinuity occurs, there are fewer oscillations, but, as observed, oscillations still do occur, contrary to what would be expected. It is believed that, although the discontinuity in pressure (which is integrated to find lift) occurs at exactly the end point of a spectral-element in time for the cells that have a boundary face on the airfoil surface, it would occur at a later time, i.e. not at the element end point in the next layer of cells because of the additional time needed for the fluid to convect to those cells after the airfoil begins its motion. In essence, the first-order effect of the transition of the airfoil into motion can be mitigated by having a

spectral element in time end at the point at which the airfoil motion begins; however, the second-order effect, in the off-body cells, still affects the C_L solution by causing oscillations in the subsequent spectral element in time.

9.2.1 An ODE with Second-derivative Discontinuity

To confirm the accuracy of the previous analysis regarding derivative discontinuities, an ordinary differential equation whose solution is similar to the motion as described by equation (9.3) is solved numerically. This ODE and its solution are given as follows, respectively:

$$\frac{dU}{dt} = \begin{cases} 0 & t \leq \frac{\pi}{10} \\ \frac{U_A}{0.9} \sin\left(\frac{t-\frac{\pi}{10}}{0.9}\right) & \frac{\pi}{10} < t < \frac{19\pi}{10} \\ 0 & t \geq \frac{19\pi}{10} \end{cases} \tag{9.4}$$

$$U(t) = \begin{cases} 0 & t \leq \frac{\pi}{10} \\ \frac{U_A}{2}\left[1 - \cos\left(\frac{t-\frac{\pi}{10}}{0.9}\right)\right] & \frac{\pi}{10} < t < \frac{19\pi}{10} \\ 0 & t \geq \frac{19\pi}{10} \end{cases} \tag{9.5}$$

where for the chosen problem $U_A = 6$ is the height of the impulse and the period as shown is necessarily 2π. Figure (9.7) plots the $L_2 - norm$ of the solution error over one period versus N_{ele} for elements with the specified polynomial degree. As can be seen, the same sawtooth pattern as was exhibited by the error in the higher-order SEMT solutions of the Euler equations is also exhibited by the error in the SEMT solutions of this ODE. The first-order and second-order SEMT error curves have average slopes of 0.99 and 2.02, respectively, between $N_{ele} = 4$ and $N_{ele} = 1024$. If only the N_{ele} numbers that are not multiples of 20 are considered, the higher-order SEMT error curves track the $N = 2$ curve pretty closely, i.e. their accuracy is reduced to second order. However, if only those N_{ele} numbers that are multiples of 20 are considered, the slope of the $N = 3$ curve is 3.01 between $N_{ele} = 20$ and 960, the slope of the $N = 4$ curve is 4.01 between $N_{ele} = 20$ and 480, and the slope of the $N = 5$ curve is 5.31 between $N_{ele} = 20$ and 120. For the ODE, as long as the second-order discontinuity occurs exactly at the endpoint of an element, the expected order of error convergence is observed. If, however, this discontinuity occurs within an element, the error

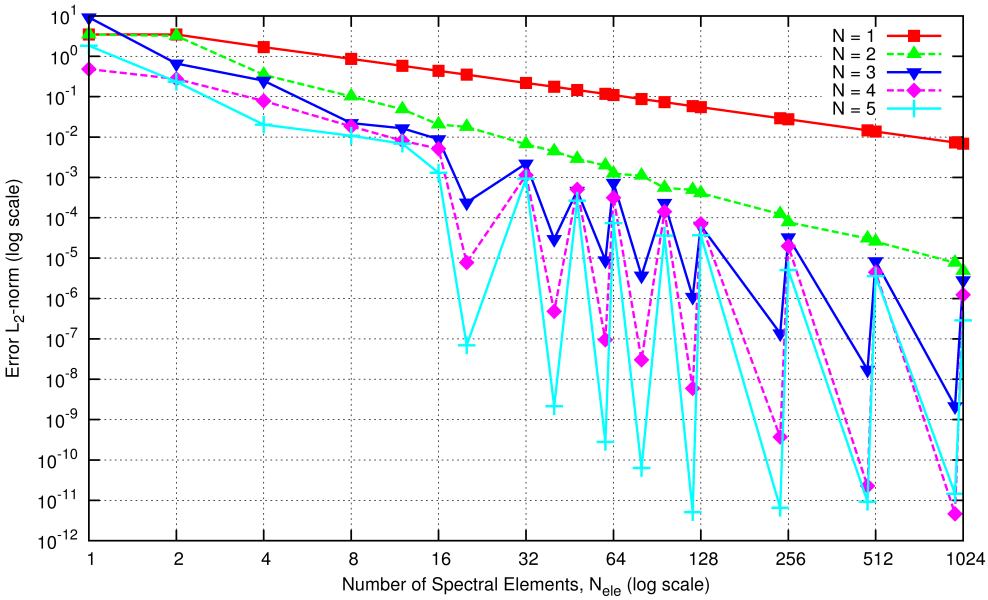


Figure 9.7: $L_2 - norm$ of the SEMT solution error for the ODE in equation (9.4) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

converges at close to second-order accuracy regardless of the degree of polynomials used within the spectral elements in time. Figure (9.8) plots the exact solution on the right axis and the instantaneous error on the left axis for the two solutions noted in the legend. The two solutions plotted have similar $L_2 - norm$s of error over the period considered. As can be seen, for the $N_{ele} = 20$ solution, the instantaneous error varies sinusoidally over the period and somewhat within each element. On the other hand, the error in the $N_{ele} = 1024$ solution oscillates at the second-derivative discontinuity, and then tracks the exact solution closely, except with a constant 3×10^{-7} offset, before it oscillates again at the second discontinuity, after which is has a small offset from zero. This solution also highlights that the oscillations that occur in the error of the SEMT Euler equations when N_{ele} is a multiple of 20 are unique to those solutions, and may very well be due to the second-order flow effects discussed above.

9.3 A Continuously Differentiable Airfoil Motion

The goal of the exploration of the SEMT solution of the Euler equations herein is to demonstrate the expected error convergence for higher-order elements. Thus far, this goal has

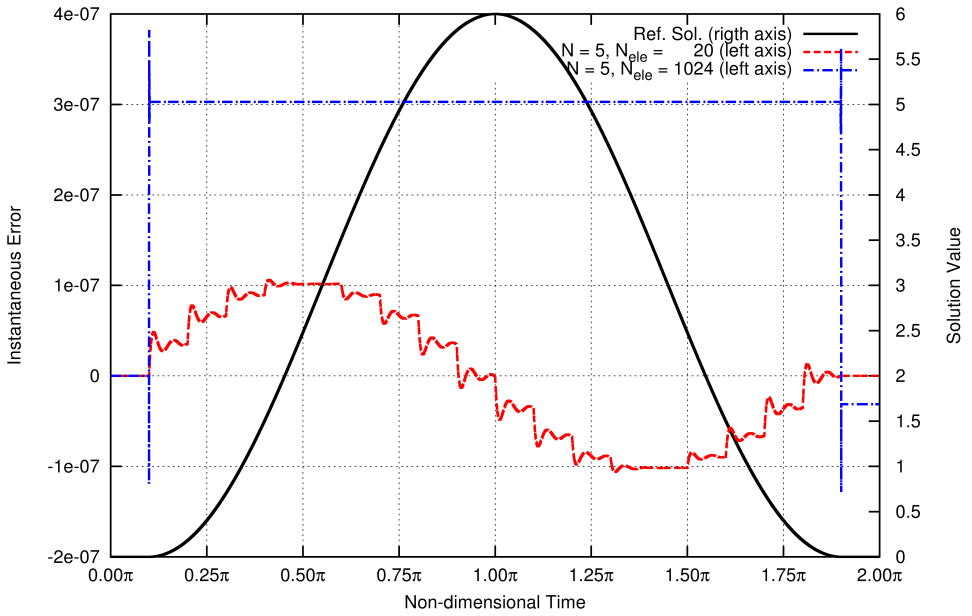


Figure 9.8: Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (9.4) and specified polynomial degree and number of spectral elements in time

not been achieved; however, the analysis up to this point has indicated that the solution of the NACA-0012 airfoil undergoing a continuously differentiable motion might produce the desired results.

For this second test case, a similar motion as was used for the first test case is prescribed. An airfoil starts at zero angle of attack and then pitches up to a prescribed angle and next pitches back down to zero, but without the beginning and ending intervals where it remains at zero angle of attack. This motion is given by the following equation:

$$\alpha(t) = \alpha_A \frac{1 - \cos\left(\omega t\right)}{2}. \tag{9.6}$$

Before this case is considered for the Euler equations, a similar ODE is solved to confirm the expected behavior. This ODE and its solution are given as follows:

$$\frac{dU}{dt} = U_A \frac{\sin t}{2} \tag{9.7}$$

$$U(t) = U_A \frac{1 - \cos t}{2}. \tag{9.8}$$

Here, $U_A = 6.0$ has the same value as in the previous ODE given by equation (9.4). Figure (9.9) plots the error curves, as has been done previously, for $N = 4, 5, 6, 8$, and 10. All curves

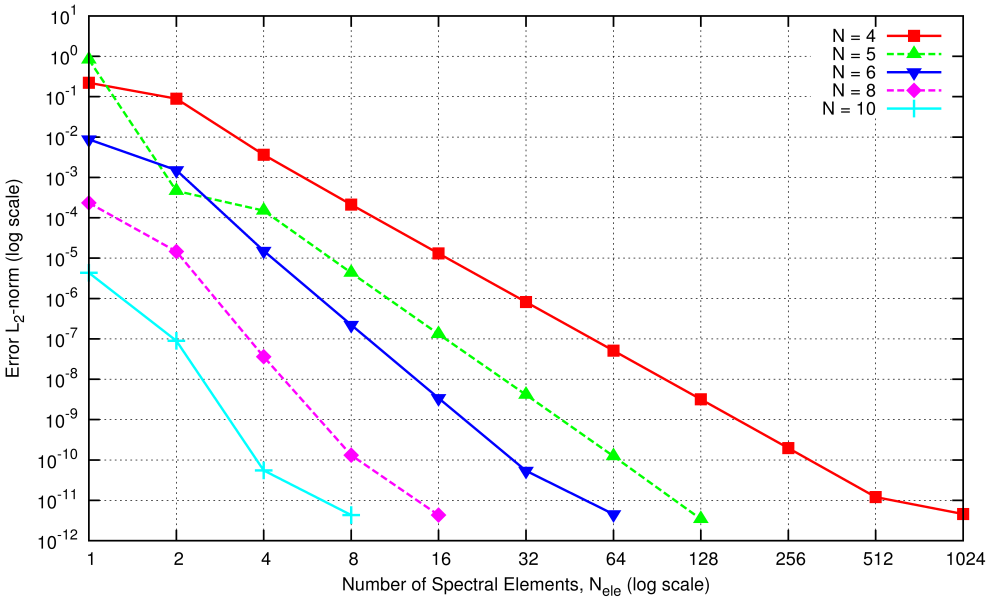


Figure 9.9: $L_2 - norm$ of the SEMT solution error for the ODE in equation (9.7) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

exhibit the expected slope in at least some limited region (the very limited region between 2 and 4 elements in the case of 10th order elements). Figures (9.10) and (9.11), which plot the instantaneous error over the period, show that the error both among the elements and within the elements themselves, is well behaved and conforms to expectations.

With the expected results of the SEMT numerical solution of the ODE analogous to the prescribed airfoil motion having been confirmed, the focus now transitions to the SEMT solution of the Euler equations for the motion described by equation (9.6) with $\alpha_A = 8.0^o$, for this case, $M_\infty = 0.25$, and $k_c = 0.001$, as previously. The value of α_A was increased to increase the maximum magnitude of C_L and hopefully with it the range of error produced in the solutions. It should be noted that $\alpha_A = 8.0^o$ is still well within the linear region of the lift curve for the NACA-0012 airfoil. Figure (9.12) plots the C_L error convergence curves for spectral elements of various polynomial degree. The slopes for 1st and 2nd order accurate schemes appear to have the appropriate values. For $N = 3$ the slope between $N_{ele} = 3$ and 64 is 3.80; for $N = 4$ the slope between $N_{ele} = 2$ and 12 is 5.07. Finally, for $N = 5$ the slope between $N_{ele} = 3$ and 8 is 5.44. As can be seen, the C_L error in these higher-order elements converges as expected over a limited range. However, as is also apparent from this figure,

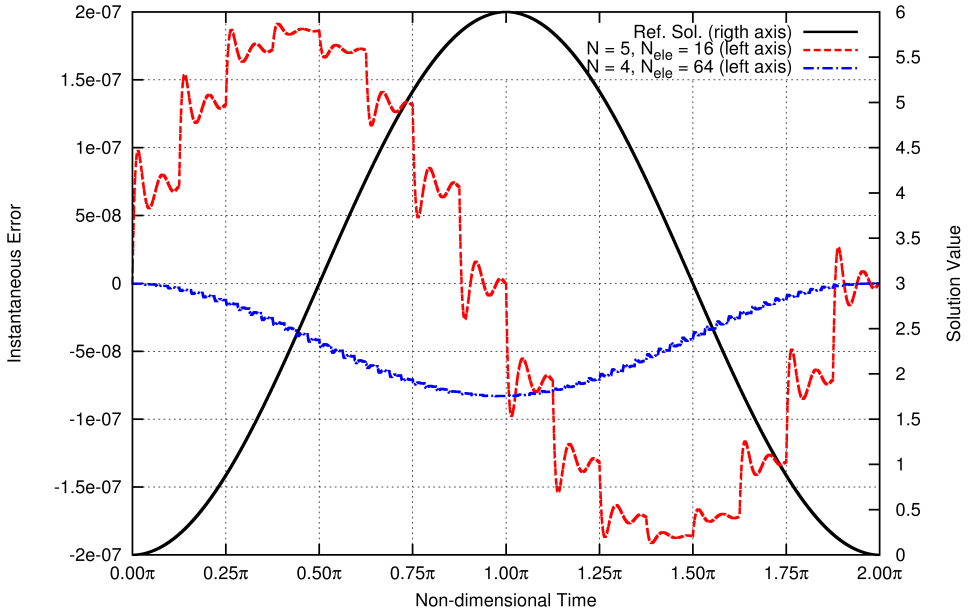


Figure 9.10: Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (9.7) and specified polynomial degree and number of spectral elements in time

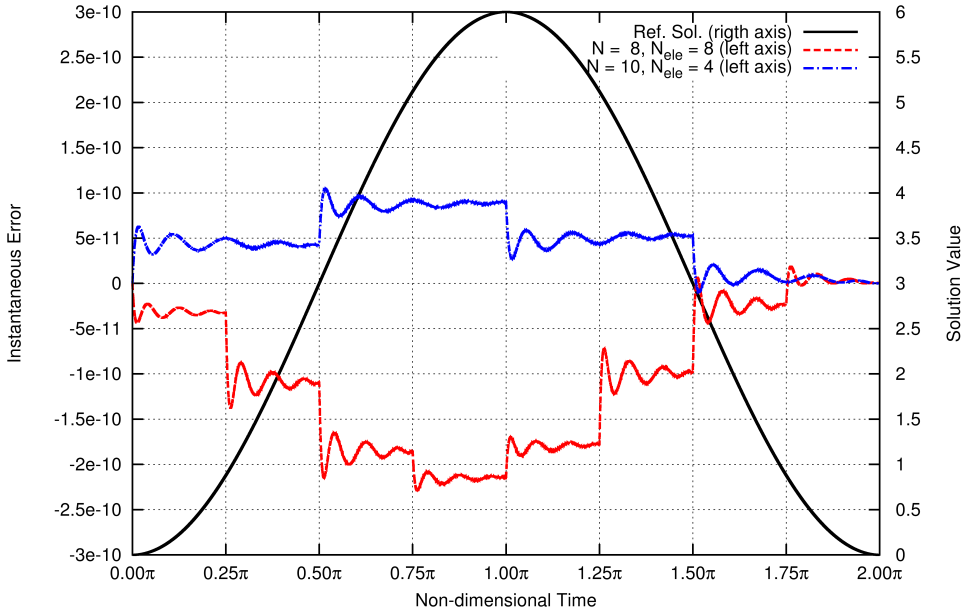


Figure 9.11: Instantaneous error of the SEMT solutions versus non-dimensional time for the ODE in equation (9.7) and specified polynomial degree and number of spectral elements in time

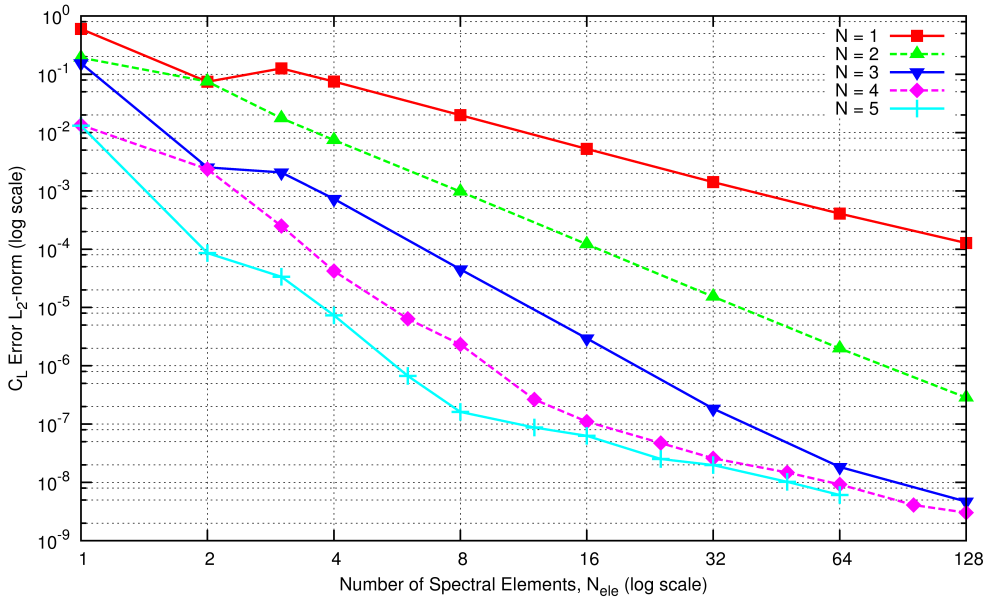


Figure 9.12: $L_2 - norm$ of the SEMT solution C_L error for the Euler equations with the airfoil motion prescribed in equation (9.6) with the given number of spectral elements in time (x-axis) and the specified polynomial degree

after $N_{ele} = 64$ for the $N = 3$ curve, after $N_{ele} = 12$ for the $N = 4$ curve, and after $N_{ele} = 8$ for the $N = 5$ curve, the slope of the C_L error convergence changes abruptly to around first-order accuracy. To ascertain why this change might occur, instantaneous C_L error plots are examined in Figures (9.13-9.15). The first of these figures shows two solutions with about the same C_L error $L_2 - norm$ over the period. The second figure is identical to the first except the range of the y-axis is changed so that the smaller variations in C_L error can be examined more closely. The last of these figures examines the error for a much more highly resolved solution than the fist two. Perhaps the first thing that is noticed in these figures is that two periods of the solution are plotted. As can be seen, there is a very large error in the first spectral element. This large error, on the order of 10^{-4} is caused because of a mis-specification of the initial condition. The steady state solution of the airfoil at zero angle of attack was used for the initial condition; however, even though the motion of the airfoil is slow enough to be considered "quasi-steady," it is not equivalent to an actual steady flow. As such, it is nearly impossible to specify the correct initial condition for such a motion.

Because of this initial error, Figure (9.12) uses the second period of motion for its results.

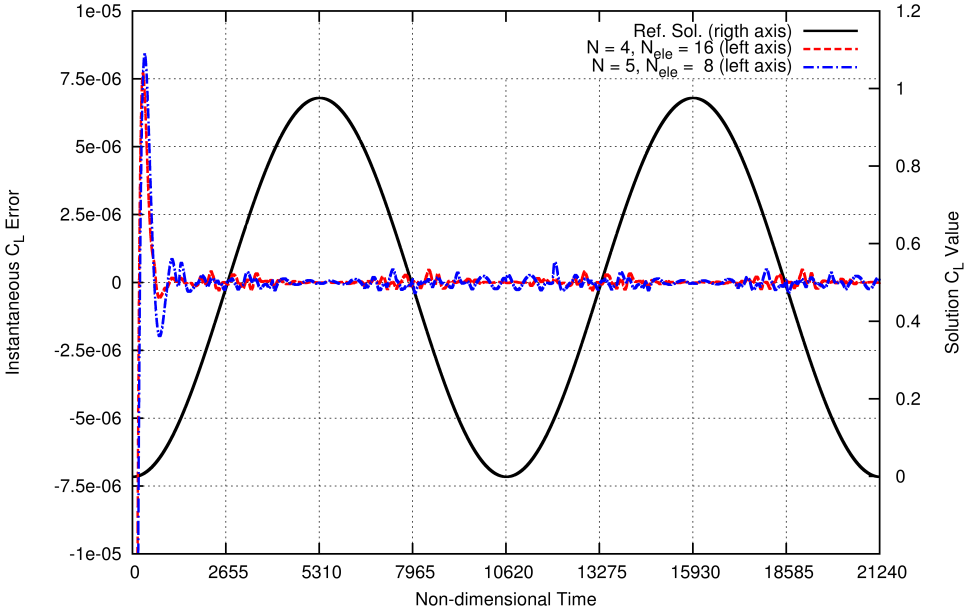


Figure 9.13: Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (9.6) and specified polynomial degree and number of spectral elements in time

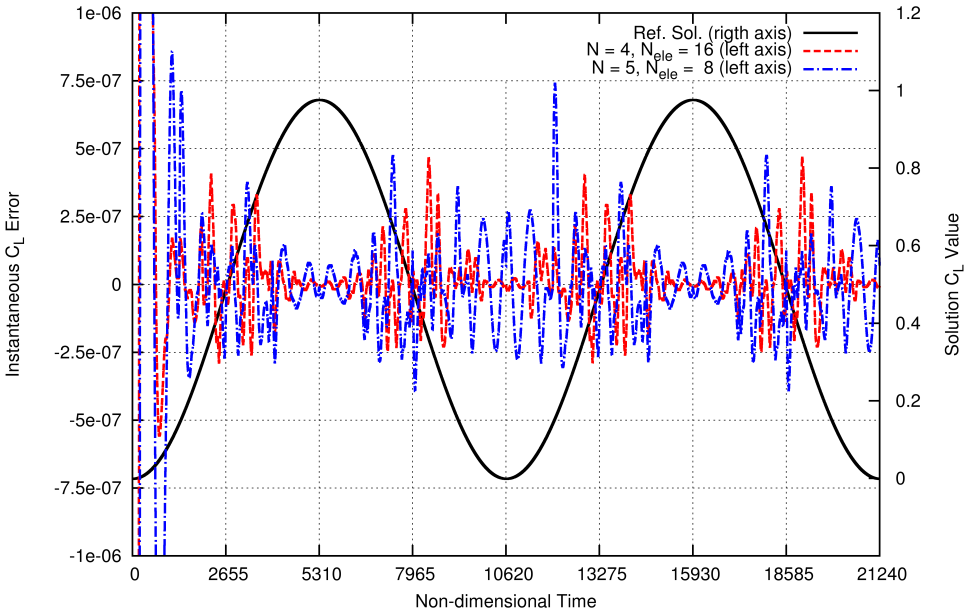


Figure 9.14: Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (9.6) and specified polynomial degree and number of spectral elements in time

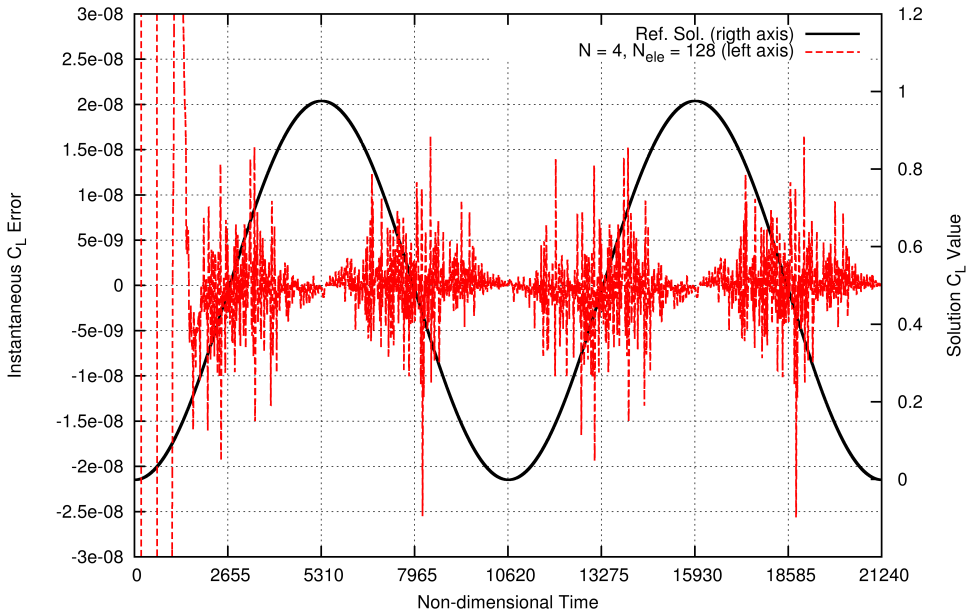


Figure 9.15: Instantaneous C_L error of the SEMT solutions versus non-dimensional time for the Euler equations with the airfoil motion prescribed in equation (9.6) and specified polynomial degree and number of spectral elements in time

Three possible explanations as to why the slope of the error convergence decreases after a certain point as the number of spectral elements in time increases are as follows:

1. Since the problem is solved hyperbolically, but is actually periodic in nature, transients exist that have not yet been dissipated in the second period

2. The initial oscillations introduce an offset to the solution as is seen in the solution of the ODE given in Figure (9.8)

3. The initial oscillations cause periodically repeating oscillations in the entire future time history of the solution

These possibilities will now be examined in the order presented. The first seems unlikely. When the instantaneous error is examined in Figure (9.15), it very clearly repeats in almost exactly the same pattern for the first and second periods (after the initial oscillations) and especially in the second half of the period. If transients were still dissipating, it would be expected that the pattern of the error in the second period would be dissimilar to that of the first.

The second possibility also seems unlikely. From all three of the instantaneous error plots, and especially those two with a narrower range, no clear offset is visible. The error appears to oscillate randomly around a mean of zero. It should be noted, however, that the C_L error that is being examined is error in an integrated quantity. Such offsets might instead manifest themselves as offsets in the pressure in individual cells surrounding the airfoil. Still, it seems intuitive that such pressure offsets would necessarily lead to a constant lift offset as well. For this reason, the second explanation is also discounted.

Finally, the third possibility seems most likely. As can be seen from all three instantaneous error plots, there are large oscillations in the C_L error, even after the initial interval. These oscillations have no corollary in the ODE error plots produced in this work. Additionally, the error oscillations repeat in the second period almost identically to the first. Certainly, it seems completely believable that an initial oscillation on the order of 10^{-4} could cause residual oscillations on the order of 10^{-8} even a long time later. For these reasons, the idea that the initial oscillations corrupt the rest of the solution seems most plausible.

Although it has not yet been implemented, purely-periodic coupling of the SEMT, as described in Chapter 2, would eliminate the potentially corrupting effect of the choice of initial condition on the periodic solution. Thus, such a solution to the problem described by equation (9.6) should be analyzed for accuracy in the future. It should be noted that all of the SEMT results have been converged to a residual tolerance of 1×10^{-14} using the GMRES-DC solver, as described above, applied to the backward-only coupling of the SEMT Euler equations.

9.4 Summary of the SEMT Results

The spectral element method in time shows promise for the time-accurate solution of the Euler equations. In this preliminary investigation, slow moving, low Mach number problems were considered for validation purposes. Even with these "easy" problems, some obvious areas for concern and improvement have been identified. Primarily, higher-order SEMT does not handle discontinuities well. This is not limited to discontinuities in the solution variables

only, but in their time derivatives as well. This issue must be addressed for higher-order SEMT to become competitive. Fortunately, experience with higher-order spatial methods already provides two paths for addressing this issue. First, h-refinement around discontinuities and order-reduction of elements containing those discontinuities provides the first path forward. An alternative path exists whereby using artificial viscosity in time to damp oscillations within elements containing discontinuities could also achieve the desired results. In the present work, the spectral-element method described falls into the family of continuous-Galerkin methods. The possibility exists that by allowing discontinuity in the solution variables themselves at element end points, the oscillations produced by derivative discontinuities within an element might decrease in amplitude. As such, discontinuous-Galerkin methods in time should also be attempted. Exploration of these possibilities remains work for the future.

Chapter 10

Conclusions and Future Work

10.1 Summary

This work has demonstrated that robust, efficient solutions of the unsteady Euler equations using the time-spectral method, quasi-periodic time-spectral method, and spectral element method in time can be found for a wide variety of problems of practical interest. These problems can either exclude or include the added complexity of aeroelastic response of the aerodynamic body in question.

In the course of this work, many improvements have been made to the implicit solver for the alternative temporal discretizations examined herein. When this work was begun BCGS-EX seemed like a good linear solver to solve flow-alone problems with the time-spectral method. Indeed for problems requiring only a few harmonics (up to the 4th harmonic, i.e. 9 time instances), BCGS-EX can be used to converge the time-spectral method more quickly than the time-implicit BDF2 method. However, it quickly became apparent that many problems of practical importance required more than 4 harmonics in order to achieve the desired accuracy. As such, the linear solver was improved to the BCGS-IM framework, which not only proved capable of producing solutions using many more time instances, but also was much faster and more robust than the BCGS-EX method. It was also found that BCGS-IM could converge problems with as many time instances as typically can be found in the literature [20–28]. However, the convergence of the BCGS-IM method was still found to

degrade as the number of time instances was further increased, mainly as a result of the small CFL number that was required to maintain diagonal dominance of the Jacobian. For this reason, and to allow for the use of both the second-order Jacobian and the implicit-coupling of the structural equations for aeroelastic cases, BCGS was abandoned as the linear solver in favor of the flexible variant of the Generalized Minimal Residual method (FGMRES) using BCGS as a preconditioner with an additional defect-correction step.

By incorporating FGMRES as the linear solver, the possibility of solving time-spectral problems with an arbitrarily large number of time instances has become possible. In the present work, multiple problems using as many as 127 time instances have been solved to machine precision. If there is an upper limit to the number of time instances that can be solved using the present solver framework, it has not yet been found.

The work with aeroelastic problems has lead to one seemingly very definitive conclusion: when different sets of disparate equations are solved simultaneously, the coupling between those sets of equations should be included at as many solution levels as possible. When aeroelastic coupling was updated only at each non-linear iteration, solution of the aeroelastic, time-spectral method could be successfully found only under a few, very precise conditions [32]. When that coupling was included in the FGMRES solver, but not in the BCGS preconditioner, most aeroelastic problems attempted could be solved in a reasonable amount of time. However, for some problems, such as those with strong gusts, equivalent to $> 5^o$ angle of attack change, convergence of the non-linear residual was very slow. Specifically, the non-linear flow residual and structural residual would alternately increase and decrease such that when the non-linear flow residual increased, the structural residual decreased, with the opposite also being true. By including the full aeroelastic coupling in both the defect-correction portion of the preconditioner, and in the BCGS algorithm itself, this convergence stagnation was overcome in the more difficult cases, while convergence of easier cases also improved as a welcome side-effect. It certainly seems to be case that the best results can be achieved when different sets of equations which must be solved simultaneously are coupled at all solution levels, i.e. in the non-linear residuals, in the FGMRES linear solver, in the defect-correction step, and in the BCGS preconditioner.

The time-spectral solver, in its current form, can solve many, if not most flow-alone and aeroelastic problems more efficiently than current, state-of-the-art time-implicit methods. This is especially true for problems:

1. That are weakly compressible (i.e. $M_\infty < 0.7$),

2. That have low reduced frequencies,

3. That have content in only the first few harmonics.

The AGARD test case no. 5, and its aeroelastic variant, which have been the subjects of much of the present work, have proven so challenging because they meet none of the above criteria. It is because of these challenges, however, that the time-spectral solver has been improved as much as it has in this work.

The difficulty of more quickly obtaining an accurate solution of the AGARD 5 test case using the time-spectral method rather than the time-implicit method has also been one of the main motivations of the recent expansion of this work to include the spectral-element method in time. By dividing the solution time domain into multiple elements and clustering these elements around areas of rapid change and large gradients, it is believed that it might be possible to obtain an accurate SEMT solution to this problem more quickly than with the time-implicit method. Expansion into SEMT was also motivated by the fact that the current FGMRES solver could be used to solve the SEMT Euler equations with no modifications. The preliminary work with the SEMT method demonstrates the potential efficiency gains that can be achieved by using this approach. Unfortunately, this preliminary work has also revealed some shortcomings which must be overcome for this method to be robust and competitive in all cases, especially in regards to problems with discontinuities of some variety.

Overall, the present work has led to the development of a robust, efficient implicit solver that has wide application to many different time discretizations of coupled CFD and computational structural dynamics (CSD) equations. Over the course of this work, solver efficiency has increased by more than an order of magnitude. The increased complexity of the problems that can be solved is not easily quantifiable. Any practical, periodic CFD/CSD

problem that can be imagined can be solved precisely with the aeroelastic time-spectral method as long as enough time instances are used. When this work began, the number of time instances which could be easily and quickly solved numbered in the teens. At the present, this number now exceeds one-hundred with no upper limit in sight. Indeed, much progress has been made to improve solution of the time-spectral and other high-order in time methods.

10.2 Advancements to the Field

- **Increasing the number of time instances that can be used in the time-spectral method**

This thesis does not represent the first use of the time-spectral method applied to the Euler equations [22] nor does it even represent the first use of the BDFTS method [81]. However, through the solver developed in this work, the time-spectral method has been extended to make use of, and problems have been solved with, as many as 127 time instances, which can resolve 63 harmonics of spectral content. This 127 time instances number does not represent a limit, but rather merely the highest number of time instances that have actually been attempted in this work. It is believed that the solver developed herein will allow the time-spectral method to be applied to arbitrarily many time instances. This is a significant accomplishment of this work.

- **Implicit aeroelastic, time-spectral solution**

While others have applied the harmonic balance and time-spectral methods to aeroelastic problems [82], this work represents the first application of these methods to the implicitly-coupled aeroelastic system. While the derivation of a complete aeroelastic Jacobian that includes the full-linear-coupling of the Euler equations and the structural equations is not groundbreaking, it has proven to be a novel approach that has allowed for the solution of much more complicated and difficult aeroelastic problems than was initially possible.

- **Quasi-periodic time-spectral flutter analysis**

The quasi-periodic time-spectral method was first developed only a year before this work was begun [81]. As such it was rather straightforward to explore a new application of this method: its application to the coupled fluid/structure aeroelastic equations. Although this extension initially seemed like it would be easily accomplished, the explicit coupling typically used for time-implicit aeroelastic calculations [3] proved not to be strong enough to allow for the BDFTS solution of aeroelastic flutter problems. It was this difficulty that lead to the development of the implicitly-coupled aeroelastic solver noted in the previous bullet. After much development, this work represents the first application of the BDFTS method to aeroelastic flutter analysis.

- **Application of SEMT to the Euler equations**

It is believed that this work represents this first application of the spectral-element method in time to the Euler equations, although the SEMT has been applied to various other ordinary and partial differential equations in the past [34]. Additionally, this work gives the first demonstration of the expected error convergence slopes using spectral elements with 1st through 5th degree polynomial bases applied to the Euler equations. This work has also demonstrated the need for *hp*-adaptation methods in time to be developed and applied to the SEMT Euler equations.

10.3 Future Work

- **Solver efficiency improvements**

The aeroelastic, time-spectral solver presented herein is more than an order of magnitude more efficient than the solver that was initially developed when this work was first undertaken [32]; however, it is still far from optimal. For an optimal solver, the wall-clock time to convergence should remain constant for time-spectral or SEMT solutions with any number of time instances using one CPU core per time instance (within a time element for SEMT). The current variant of the solver uses significantly more than the optimal amount of wall-clock time for large numbers of time instances when 1 core is used for each time

instance. This trend worsens as the freestream Mach number and reduced frequency of prescribed pitching increase, especially for transonic Mach numbers. However, despite the non-optimality of the solver, the time-spectral method can solve many, if not most, flow and aeroelastic problems in less wall-clock time than time-implicit methods, both because time-implicit methods require so many periods of solution to arrive at a purely periodic solution and because of the additional parallelism in time afforded by the time-spectral method and the spectral-element method in time.

When comparing time-spectral and time-implicit methods, the following should be kept in mind. The time-spectral method becomes stiff at a) high freestream Mach number because of the presence of shock waves, b) high reduced frequency because the various time instances become more tightly coupled, and c) large numbers of time instances because the frequency of the highest harmonic that can be resolved increases with the number of time instances and thereby decreases the pseudo-time step size. Exacerbating this situation, it is usually these high Mach number, high reduced frequency cases that require large numbers of time instances to obtain an accurate solution. On the other hand, time-implicit methods scale more slowly than linearly when additional time steps are added because a smaller time step tends to make the Jacobian more diagonally dominant and thus each time step easier to solve. These two aspects combine to demonstrate how challenging it is to develop a time-spectral solver that can outperform a time-implicit solver for problems requiring very large numbers of time instances and time steps per period, respectively. However, it has also been shown that, because the entire period is solved simultaneously in the time-spectral method, the convergence tolerance for TS need not be as strict as for BDF2, where error can accumulate across the many thousands of time steps that must be solved to attain a periodic solution. This finding may prove a key the benefit of the time-spectral method when compared to traditional time-implicit methods.

Despite the large solver efficiency gains that have been made as compared to when this work was begun, solver efficiency must be improved further still. First and foremost, a more efficient preconditioner, such as, linear-agglomerated multigrid should be implemented. While the expected efficiency gains from multigrid are large, other preconditioning tech-

niques that completely eliminate the need for diagonal dominance, such as incomplete LU factorization, should also be tested. Finally, these methods need not be used in isolation; a preconditioner that combines the two might prove to offer the efficiency gains that are being sought. Exploration of better preconditioning methods is where the pursuit of an optimal time-spectral solver should be focused.

- **Flutter frequency determination for BDFTS flutter**

As was noted in the BDFTS flutter chapter, the actual frequency at which aeroelastic flutter occurs is close to the natural frequency of pitch, but is not, in fact, exactly that frequency. As such, in the BDFTS flutter analysis presented herein, the flutter frequency is found using the BDF2 method, and this flutter frequency is prescribed for the BDFTS solution. Obviously, it is pointless to solve flutter problems in this way in real world applications since it is necessary to find a BDF2 solution for all cases, before the BDFTS solution can be found. In that case, the BDFTS solution is additional, unnecessary work that gives no new information. As such, for the BDFTS method for flutter to become a suitable method for wide-scale use, a procedure must be developed such that the flutter frequency can be found as the BDFTS flutter solution proceeds.

Such a method might be similar to a method that Gopinath has developed to find the period of vortex shedding when using the purely periodic time-spectral method [28]. This method converges the solution part way using an initial guess of the shedding frequency, until convergence stagnates, and then uses the gradients of an objective function to update the frequency to a better value that allows the solution to converge further. The process is repeated until the desired level of convergence can be achieved. However, it is not clear that this procedure can be used for the BDFTS method as the combination of polynomial and spectral bases in the BDFTS method allow it to converge to machine zero even if the incorrect period is specified (the solution produced just shows poor agreement with the actual solution). On the other hand, it does seem likely that an optimization procedure of some type could be used, where, for a given number of time-spectral time instances, the error over the solution is minimized with the flutter frequency as the design variable. To develop and implement this optimization remains a topic for future work.

- **Efficiency quantification for BDFTS**

Even with the extension of BDFTS to aeroelastic problems, uncertainty remains regarding under what conditions the BDFTS method might be more efficient than the BDF2 method. Certainly, it has been shown that one of the main advantages of the time-spectral method over time-implicit methods is that no transients must be dissipated in real-time before a periodic solution can be found. This advantage does not exist for the quasi-periodic time-spectral method, which is usually applied to transient problems with strong periodic content. As such, in most cases, the same number of periods are to be computed regardless of whether the BDFTS method or the BDF2 method is used. Thus, for the BDFTS method to be more efficient than the BDF2 method, each period solved with the BDFTS method needs to be solved more quickly than that same period using the BDF2 method. More than likely, for this to be the case, each period must have content in a small number of harmonics, for instance seven. However, perhaps the periods of solution in the BDFTS method need not be converged to as strict a tolerance as the time steps if the BDF2 method were used, as has been shown for the time-spectral method. Examination of these efficiencies and the issues that affect them remains for future work.

- **Deforming meshes**

All of the problems in this thesis that have required the mesh to move have used a rigidly moving mesh. In order for the time-spectral aeroelastic method as outlined herein to be extended to three-dimensional aeroelastic problems, a deforming mesh capability must be implemented. The Euler equations in Chapter 2 have already been presented in an arbitrary-Lagrangian-Eulerian form that allows cell volume to change as the mesh deforms. To retain the conservative nature of the scheme, a geometric conservation law must be satisfied [83–88]. Regardless of what method is chosen to deform the mesh, a new set of equations describing this mesh deformation must also be solved. To retain the efficiency of the solver, these mesh deformation equations would need to be fully-coupled into the solver: in FGMRES, the defect-correction step, and the preconditioner, instead of only coupling mesh deformation at the non-linear iteration level. Such an implicitly-coupled system would yield a much larger

Jacobian that took into account how the deforming mesh affects the flow variables, which in turn affect the structural equations, which then affect the mesh deformation, etc. The development of a mesh deformation capability, which is vital for this work to be extended into three dimensions, remains work for the future.

- **The time-spectral method for overset mesh problems**

The use of overset meshes for the solution of problems with both rotating and fixed bodies is common. When overset methods are used, the solution variables in cells of the background mesh that fall inside of the aerodynamic body at a give time, i.e. blanked cells, have no set value. For time-implicit methods, these blanked cells cause no problems as the time step size is small enough that blanked cells can acquire values while they are in the interpolation region of the flow before they re-enter the solution region of the flow. On the other hand, if the spectral basis used in the time-spectral method is fully retained, all cells must have values for all time instances. The author of this thesis has previously examined applying the time-spectral method to problems solved using overset meshes [29]. The values of the conserved variables in blanked cells are generated through the solution of Poisson's equation in these cells using the surrounding real valued cells as boundary conditions for each time instance individually. Leffell et al. take a different approach and use barycentric rational interpolants to span the time-domains between blanked cells, requiring no values be assigned to these cells when they are blanked [89]. Although both these methods allow for the solution of time-spectral problems on overset meshes, they both have drawbacks as acknowledged by their authors [29, 89]. Thus, exploration of better solution methods for time-spectral problems on overset meshes, perhaps by using non-uniform Fourier transforms, should be pursued in future work.

- **Adjoint based *hp*-refinement in time**

In the present work, when a time-spectral or BDFTS solution is found, the number of time instances used for the solution is an initial input specified by the user. The number of time instances is increased until, either the solution has changed negligibly as a result of

the additional time instances, or until it matches a reference solution significantly well. For practical problems, reference solutions are not possessed *a priori*, indeed what need would there be to run a CFD solution if the answer were already known? As such, the former procedure is used for practical problems. This procedure requires the engineer to check the converged solutions to see if they are similar enough to stop adding time instances. Instead, such a procedure could by automated by using the adjoint sensitivities to automatically add time instances (p-refinement) when the uncertainty in some objective perhaps involving C_L, C_D, and/or C_m is below a certain threshold.

When applied to the BDFTS method, adjoint based p-refinement has even more promise. Suppose the complete time history of the aeroelastic response of a body to a transient gust, where the airfoil starts out at rest a long time before the gust and returns to rest a long time after the gust passes, is desired. This problem could be solved with the purely periodic time-spectral method with a period sufficiently long that the airfoil returns to rest before the gust hits again; however, as the temporal width of the gust (found by dividing the spatial gust width by the freestream velocity) relative to the length of the period is decreased, increasingly more harmonics are needed to resolve the gust. As such, it can be imagined that the problem could quickly become intractable. Another approach, however, would be to use the a BDFTS time step size (BDFTS period) equal to the temporal length of the gust. The time step containing the gust would have content in many harmonics whereas the time steps very far before and after that gust would only have content in the constant mode. Adjoint based p-refinement could be used to ensure that each period (time step) of the solution uses enough time instances to resolve the harmonic content within that period, with different periods using different numbers of time instances. In such a way, each period would use the optimal number of time instances necessary for a given level of accuracy.

For the TS and BDFTS methods, adjoint based p-refinement would be a nice addition, but hardly constitutes a necessity. On the other hand, for the SEMT method, hp-refinement should prove to be invaluable. The SEMT results presented herein demonstrate that poor error convergence can arise when a high-order time element spans a discontinuity in time, even if that discontinuity exists in a derivative of the functional variables being sought.

As mentioned previously, a good way to prevent this degradation in error convergence is to add elements where they are needed most instead of uniformly, which is exactly what h-refinement would do. Further, p-refinement could be used within elements to increase the accuracy within large elements that only contain smooth variations of the functional variables. Finally, both methods can be combined for hp-refinement whereby the the precise distribution of element size and order can be determined to minimize the temporal error at the lowest computational cost.

- **Artificial viscosity in time**

As mentioned previously, an alternative method to eliminate spurious oscillations resulting from discontinuities within spectral elements in time is through the addition of an artificial viscosity term in time. In fact, there are third and forth methods to achieve these same goals, namely slope limiters [90–93] and total variation bounded (TVB) finite-element discretizations [94]. However, these two additional "fixes" to limit oscillations in finite elements are not suggested as they both have significant drawbacks. Slope limiters can give unnecessary reductions in accuracy in smooth regions and can prevent convergence to steady state. TVB discretizations contain problem dependent coefficients which can be difficult to estimate. Burbeau et al. [95] has developed TVB without these coefficients, but this work is still in its infancy and needs more validation for real applications before it is considered.

For the reasons stated above, the implementation of artificial viscosity to stabilize SEMT and damp spurious oscillations is suggested. This approach is well established in the literature [70, 96–100]. Such an artificial viscosity term would take the form of a second derivative in time and would tend to take what would otherwise be a discontinuity and diffuse or smear it out over a finite region. In such a way, a non-smooth event becomes computationally smooth. Investigation of artificial viscosity techniques in time should also be pursued.

Although not addressed previously in this work, spectral methods can become prone to aliasing errors, whereby content at higher frequencies, which cannot be resolved with the given number of time instances, corrupts the solution at the lower frequencies that can be resolved [101]. Aliasing errors can both degrade convergence of spectral methods and lead to

convergence to non-physical solutions [101]. One method for addressing aliasing error is to resolve fifty-percent more frequencies than are actually desired, and then to drop the content from the highest one-third of the computed frequencies. This would necessarily increase the number of time instances which must be used by the same factor of $\frac{3}{2}$, e.g. if the first 20 harmonics were desired to be resolved, 30 harmonics would need to be computed: the number of time instances needed would increase from 41 to 61. Such an increase in the number of time instances would make the problem more stiff and could thereby slow convergence as well. Another, perhaps more elegant method to eliminate aliasing errors is temporal artificial viscosity [102–106], as mentioned above for SEMT. Thus, temporal artificial viscosity should be explored in the context of the time-spectral method and BDFTS as well.

- **Cell-discontinuous time-spectral method**

For all of the problems presented in this thesis, it was assumed that all cells in the spatial domain required exactly the same harmonic resolution, e.g. if $N = 63$ time instances were required to resolve C_D to the required precision in a single cell in the spatial domain, all cells in the domain are required also to use 63 time instances. For cells very far away from the aerodynamic body in question, it seems intuitive that less harmonic resolution would be required. Indeed the freestream boundary conditions prescribed have zero harmonic content.

It seems logical that a more efficient time-spectral discretization would allow each cell to have a different number of time instances, as required to resolve whatever objective to the required degree of precision. Although this idea was arrived at independently in the present work, it turns out that this same concept has already been pursued for the non-linear frequency domain and harmonic balance methods [107–111]. When adjacent cells have different numbers of time instances, i.e. are spectrally discontinuous, the state variables can be spectrally interpolated to the number of time instances of the adjacent cell before the fluxes are calculated. Concern exists as to whether such a scheme could be made to be discretely conservative, so conservation would need to be investigated. In fact, this discretization as described by Mosahebi and Nadarajah for the non-linear frequency domain method [110, 111] is stated to be non-conservative. However, it can be imagined that if the harmonic content in all cells were sufficiently refined such that content in the highest

harmonic that can be resolved in a given cell were negligible, the magnitude of the non-conservative effects of such a scheme would themselves also be negligible.

With such a scheme, an initial solution with a small number of time instances, for instance five time instances, could be found and then adjoint based p-refinement in time could be used to increase the number of harmonics in cells where such refinement is necessary. Although conservation may preclude the feasibility of such a scheme, which seems unlikely, investigation should be pursued until a definitive answer is found; after all, overset methods are not discretely conservative, but are still in wide and common use because the non-conservative terms can be made to be negligible.

- **Spatial-temporal diagonal-block inversion**

The current framework assumes that the spatial discretization contributes more to the overall stiffness of the problem than the temporal discretization, i.e. $\| \lambda \| > V^n k'$ in equation (4.3) for all but the very largest cells. However, as has been noted previously, as the reduced frequency or the number of time instances used increases, the temporal discretization becomes stiffer. As a consequence, the temporal term restricts the pseudo-time step size as given by equations (4.3) and (4.4) for all but the very smallest cells in these instances. One method to mitigate this temporal stiffness might be to implement an approximate-factorization algorithm, as used by others as the linear solver for time-spectral discretizations [89, 112], as a preconditioner within FGMRES.

Another way in which these effects might be mitigated if the spatial-temporal diagonal blocks were inverted in the preconditioner instead of the spatial diagonal blocks for each cell and time instance alone. In other words, in the current implementation, only the $[4 \times 4]$ diagonal blocks of the spatial Jacobian are inverted directly in the preconditioner; whereas, for the suggested implementation, the $\Big[(N \times 4) \times (N \times 4)\Big]$ spatial-temporal diagonal blocks would be inverted directly in the preconditioner (where N is the number of time instances). For example, if three time instances were used, the spatial-temporal diagonal blocks would each be $[12 \times 12]$ with the $[4 \times 4]$ spatial diagonal blocks of the three time instances along the diagonal and the time-spectral coupling coefficients forming $[4 \times 4]$ off-diagonal blocks which

are themselves each diagonal. By making this change, it is believe that stronger temporal coupling is retained which might allow for larger pseudo-time step sizes to be used.

It must be cautioned that the current time-instance distribution, where each time instance spawns its own MPI process, is not easily conducive to an efficient implementation of spatial-temporal diagonal-block inversion. As such, an efficient implementation of this idea could be more easily made if all time instances of each spatial cell were placed on a single CPU and the discretization instead parallelized in space alone. It should be noted that distributing time instances in this way does not remove the advantage of TS in allowing parallelization in time; instead, it recasts this additional parallelization back into the spatial domain by allowing greater spatial parallelization before its benefits are exhausted. Further, if different numbers of time instances were to be used in different spatial cells, as has been suggested above, such a distribution of time instances would also be necessary.

- **Monolithic-time and periodic SEMT**

As mentioned previously, only the backward-only coupling of the SEMT discretization has been implemented and tested. Not only was this coupling method easiest to implement, it also could use the existing 1st-order BDFTS solver efficiently without modification because, within each element, the SEMT matrix is dense. Although, the 1st-order BDFTS solver could be used for the monolithic-time approach (with the SEMT matrix), and the periodic TS solver could be used for periodic SEMT, such use would be inefficient. The full SEMT matrix for either of these two latter couplings is not dense, but block-diagonal. As such, these matrices contain many zeroes, which would be included if the existing TS and BSDFTS solvers were used, since they assume dense matrices. In the future, these two additional coupling methods for SEMT should be implemented with their own specific solvers to retain the efficiency these methods afford.

- **Discontinuous-Galerkin Method in Time (DGMT)**

It has been noted that the SEMT method, as outlined in this thesis, is a continuous Galerkin method in time. Discontinuous-Galerkin methods in time might offer better error

convergence than the SEMT method [46]. Additionally, the additional degree of freedom allowed by the discontinuous nature of the end points of each time element in the DGMT might cause oscillations in elements that contain discontinuities to be less severe. Because of the breadth of literature on DGMT [45, 46, 113–117], SEMT seemed like the more novel choice to pursue; however, DGMT should still be explored in the future, if nothing else, so confidence can be had of the choice of finite element.

- **Fully-Implicit Runge-Kutta (FIRK)**

Another set of methods for discretizing time is the fully-implicit Runge-Kutta methods (FIRK). There are many different FIRK methods, all based on the quadrature rules of Gauss, Lobatto, and Radau [118]. The various FIRK discretizations have different forms and orders based on what quadrature rule is used and how many stages are used. Many, but not all, FIRK methods are collocation methods [67, 118–120]. Like finite element methods, FIRK methods can experience reduced order of accuracy in non-smooth regions [118]. FIRK methods represent a similar process to finite element methods in time in that there are two different matrices, one that modifies the conserved variables and another that modifies the residuals, similar to equation (2.69). However, the difference arises because the matrix that modifies the residuals is usually dense or lower triangular, whereas the matrix that modifies the conserved variables is usually diagonal or identity [67, 118–120]. For linear problems, specific FIRK schemes can be shown to be equivalent to SEMT or DGMT methods; however, for general non-linear problems, such as those considered in this work, this difference makes the FIRK methods not amenable to the current solver as outlined in this thesis, and therefore their implementation was left for future work.

References

[1] J. J. Alonso and A. Jameson, "Fully-implicit time-marching aeroelastic solutions," ser. AIAA Paper 1994-0056, Jan. 1994.

[2] K. Willcox and J. Peraire, "Aeroelastic computations in the time domain using unstructured meshes," *International Journal for Numerical Methods in Engineering*, vol. 40-13, 1997.

[3] K. Mani and D. J. Mavriplis, "Adjoint-based sensitivity formulation for fully coupled unsteady aeroelasticity problems," *AIAA Journal*, vol. 47, no. 8, pp. 1902–1915, 2009.

[4] A. K. Gopinath, "Efficient Fourier-based algorithms for time-periodic unsteady problems," Ph.D. dissertation, Stanford University, 2007.

[5] D. J. Mavriplis and Z. Yang, "Time spectral method for periodic and quasi-periodic unsteady computations on unstructured meshes," *Mathematical Modelling of Natural Phenomena*, vol. 6, no. 3, pp. 213–236, 2011.

[6] Z. Yang, D. J. Mavriplis, and J. Sitaraman, "Prediction of helicopter maneuver loads using BDF/time spectral method on unstructured meshes," ser. AIAA Paper 2011-1122, June 2011.

[7] J. M. Verdon and J. R. Casper, "A linearized unsteady aerodynamic analysis for transonic cascades," *Journal of Fluid Mechanics*, vol. 149, pp. 403–429, 1983.

[8] K. C. Hall, W. S. Clark, and C. B. Lorence, "A linearized Euler analysis of unsteady transonic flows in turbomachinery," *ASME Journal of Turbomachinery*, vol. 116, no. 3, pp. 477–488, 1994.

[9] J. J. Adamczyk, "Model equations for simulating flows in multistage turbomachinery," ser. ASME Paper (85-GT-226), 1988.

[10] L. He, "I. Modelling issues for computation of unsteady turbomachinery flows; II. Time-marching calculations for blade row interaction and flutter," ser. VKI Lecture Series, von Karman Institute for Fluid Dynamics, Belgium, 1996, 1988.

[11] W. Ning and L. He, "Computation of unsteady flows around oscillating blades using linear and nonlinear harmonic Euler methods," *ASME Journal of Turbomachinery*, vol. 120, no. 3, pp. 508–514, 1998.

[12] L. He and W. Ning, "Efficient approach for analysis of unsteady viscous flows in turbomachines," *AIAA Journal*, vol. 36, no. 11, pp. 2005–2012, 2005.

[13] L. He, "Three-dimensional unsteady Navier-Stokes analysis of unsteady stator-rotor interaction in axial-flow turbines," *IMechE Journal of Power and Energy*, vol. 214, no. 1, pp. 13–22, 2000.

[14] K. C. Hall and E. F. Crawley, "Calculation of unsteady flows in turbomachinery using the linearized Euler equations," *AIAA Journal*, vol. 27, no. 6, pp. 777–787, 1989.

[15] K. C. Hall, C. B. Lorence, and W. S. Clark, "Nonreflecting boundary conditions for linearized unsteady aerodynamic calculations," ser. AIAA Paper 1993-0882, Jan. 1993.

[16] K. C. Hall and P. D. Silkowski, "The influence of neighboring blade rows on the unsteady aerodynamic response of cascades," *Journal of Turbomachinery*, vol. 119, no. 1, pp. 85–93, 1997.

[17] P. D. Silkowski and K. C. Hall, "A coupled mode analysis of unsteady multistage flows in turbomachinery," *Journal of Turbomachinery*, vol. 120, no. 3, pp. 410–421, 1998.

[18] K. C. Hall and K. Ekici, "Multistage coupling for unsteady flows in turbomachinery," *AIAA Journal*, vol. 43, no. 3, pp. 624–632, 2005.

[19] K. C. Hall, J. P. Thomas, and W. S. Clark, "Computation of unsteady nonlinear flows in cascades using a harmonic balance technique," *AIAA Journal*, vol. 40, no. 5, pp. 879–886, 2002.

[20] M. McMullen, A. Jameson, and J. J. Alonso, "Acceleration of convergence to a periodic steady state in turbomachineary flows," ser. AIAA Paper 2001-0152, Jan. 2001.

[21] ——, "Application of a non-linear frequency domain solver to the Euler and Navier-Stokes equations," ser. AIAA Paper 2002-0120, Jan. 2002.

[22] A. K. Gopinath and A. Jameson, "Time spectral method for periodic unsteady computations over two- and three- dimensional bodies," ser. AIAA Paper 2005-1220, Jan. 2005.

[23] E. van der Weide, A. K. Gopinath, and A. Jameson, "Turbomachineary applications with the time spectral method," ser. AIAA Paper 2005-4905, June 2005.

[24] K.-H. Lee, J. J. Alonso, and E. van der Weide, "Mesh adaptation criteria for unsteady periodic flows using a discrete adjoint time-spectral formulation," ser. AIAA Paper 2006-0692, Jan. 2006.

[25] S. Sankaran, A. Gopinath, E. V. D. Weide, C. Tomlin, and A. Jameson, "Aerodynamics and flight control of flapping wing flight vehicles: A preliminary computational study," ser. AIAA 2005-0841, Jan. 2005.

[26] S. Choi, M. Potsdam, K. Lee, G. Iaccarino, and J. J. Alonso, "Helicopter rotor design using a time-spectral and adjoint-based method," ser. AIAA 2008-5810, Sep. 2008.

[27] S. Choi and A. Datta, "CFD prediction of rotor loads using time-spectral method and exact fluid-structure interface," ser. AIAA 2008-7325, Aug. 2008.

[28] A. K. Gopinath and A. Jameson, "Application of the time spectral method to periodic unsteady vortex shedding," ser. AIAA Paper 2006-0449, Jan. 2005.

[29] D. J. Mavriplis, Z. Yang, and N. L. Mundis, "Extensions of time spectral methods for practical rotorcraft problems," ser. AIAA Paper 2012-0423, Jan. 2012.

[30] J. G. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge U. P., 2006.

[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[32] N. L. Mundis and D. J. Mavriplis, "Quasi-periodic time spectral method for aeroelastic flutter analysis," ser. AIAA Paper 2013-0638, Jan. 2013.

[33] N. L. Mundis, D. J. Mavriplis, and J. Sitaraman, "Quasi-periodic time-spectral methods for flutter and gust response," ser. 69th Forum of the American Helicopter Society, AHS International, Alexandria, VA, May 2013.

[34] M. H. Kurdi and P. S. Beran, "Spectral element method in time for rapidly actuated systems," *Journal of Computational Physics*, vol. 227, no. 3, pp. 1809–1835, 2008.

[35] A. Ben-Tal, P. Bar-Yoseph, and H. Flashner, "Space-time spectral element method for optimal slewing of a flexible beam," *International Journal of Numerical Methods in Engineering*, vol. 39, no. 18, pp. 3101–3121, 1996.

[36] P. Beran, C. Petit, D. Millman, and J. Halley, "Uncertainty quantification of limit-cycle oscillations," *Journal of Computational Physics*, vol. 217, no. 1, pp. 217–247, 2006.

[37] M. Borri, "Helicopter rotor dynamics by finite element time approximation," *Computers Mathematics with Applications*, vol. 12A, no. 1, pp. 149–160, 1986.

[38] J. Oden, "A general theory of finite elements II. Applications," *International Journal of Numerical Methods in Engineering*, vol. 1, no. 3, pp. 247–259, 1969.

[39] C. Desai, J. Oden, and L. Johnson, "Evaluation and analyses of some finite element and finite difference procedures for time dependent problems," AEWES-Misc-Paper-S-75-7, Tech. Rep. Accession No.: ADA009739, 1975.

[40] I. Fried, "Finite element analysis of time-dependent phenomena," *AIAA Journal*, vol. 7, no. 6, pp. 1170–1173, 1969.

[41] J. Argyris and D. Scharpf, "Finite elements in time and space," *Nuclear Engineering and Design*, vol. 10, no. 4, pp. 456–464, 1969.

[42] K. Palaniappan, P. Beran, and A. Jameson, "Optimal control of lcos in aerostructural systems," ser. AIAA Paper 2006-1621, May 2006.

[43] S. Nadarajah and A. Jameson, "Optimum shape design for unsteady flows with time-accurate continuous and discrete adjoint methods," *AIAA Journal*, vol. 45, no. 7, pp. 1478–1491, 2007.

[44] R. Riff and M. Baruch, "Time finite element discretization of hamilton's law of varying action," *AIAA Journal*, vol. 22, no. 9, pp. 1310–1318, 1984.

[45] D. Aharoni and P. Bar-Yoseph, "Mixed finite element formulations in the time domain for solution of dynamics problems," *computational Mechanics*, vol. 9, no. 5, pp. 359–374, 1992.

[46] P. Bar-Yoseph, D. Fisher, and O. Gottlieb, "Spectral element methods for nonlinear temporal dynamical systems," *Computational Mechanics*, vol. 18, no. 4, pp. 302–313, 1996.

[47] T. Tezduyar, M. Behr, and J. Liou, "New strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial domain space-time procedure. I. The concept and the preliminary numerical tests," *Computer Methods in Applied Mechanics and Engineering*, vol. 94, no. 3, pp. 339–351, 1992.

[48] T. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space-time finite element techniques for computation of fluid-structure interactions," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 17-18, pp. 2002–2027, 2006.

[49] J. Pontaza and J. Reddy, "Space-time coupled spectral/hp least-squares finite element formulation for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 197, no. 2, pp. 418–459, 2004.

[50] B. Bell and K. Surana, "A space-time coupled p-version least squares finite element formulation for unsteady two-dimensional Navier-Stokes equations," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2593–2618, 1996.

[51] C. Klaij, J. van der Vegt, and H. van der Ven, "Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 217, no. 2, pp. 589–611, 2006.

[52] ——, "Pseudo time-stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 219, no. 2, pp. 622–643, 2006.

[53] J. van der Vegt and H. van der Ven, "Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressibel flows. I. general formulation," *Journal of Computational Physics*, vol. 182, no. 2, pp. 546–585, 2002.

[54] P. Bar-Yoseph and E. Moses, "Space-time spectral element methods for unsteady convection diffusion problems," *International Journal of Numerical Methods for Heat and Fuild Flow*, vol. 7, no. 2-3, pp. 215–235, 1997.

[55] P. Bar-Yoseph, E. Moses, U. Zrahia, and A. Yarin, "Space-time spectral element methods for one-dimensional nonlinear advection-diffusion problems," *Journal of Computational Physics*, vol. 119, no. 1, pp. 62–74, 1995.

[56] G. Hulbert and T. Hughes, "Space-time finite element methods for second-order hyperbolic equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 84, no. 3, pp. 327–348, 1990.

[57] E. Fehlberg, "Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with step size control," NASA Technical Report, Tech. Rep., 1968.

[58] N. Newmark, "A method of computation for structural dynamics," *ASCE Journal of the Engineering Mechanics Division*, vol. 5, no. EM3, pp. 67–94, 2002.

[59] H. Hilber, T. Hughes, and R. Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering and Structural Dynamics*, vol. 5, pp. 283–292, 1977.

[60] J. Yao, A. Jameson, J. Alonso, and F. Liu, "Development and validation of a massively parallel flow solver for turbomachinery flows," *Journal of Propulsion and Power*, vol. 17, no. 3, pp. 659–668, 2001.

[61] B. Cockburn and C. Shu, "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, vol. 16, no. 3, pp. 173–261, 2001.

[62] M. Giles, "Stability analysis of a Galerkin/Runge-Kutta Navier-Stokes discretisation on unstructured tetrahedral grids," *Journal of Computational Physics*, vol. 132, no. 2, pp. 201–214, 1997.

[63] N. Nikitin, "Third-order-accurate semi-implicit Runge-Kutta scheme for incompressible Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 51, no. 2, pp. 221–233, 2006.

[64] D. J. Mavriplis, "Unstructured mesh discretizations and solvers for computational aerodynamics," *AIAA Journal*, vol. 46, no. 6, pp. 1281–1298, 2008.

[65] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes," ser. AIAA Paper 1981-1259, 1981.

[66] D. J. Mavriplis, "Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes," *AIAA Journal*, vol. 26, no. 7, pp. 824–831, 1988.

[67] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Wiley, 2003.

[68] D. Gottlieb and S. A. Orszag, "Numerical analysis of spectral methods: Theory and applications," ser. CBMS-26, Regional Conference Series in Applied Mathematics, SIAM,Philadelphia, PA, 1977.

[69] D. Burnett, *Finite Element Analysis: from Concepts to Applications*. Addison-Wesley, 1988.

[70] B. T. Helenbrook, "Artificial compressibility preconditioning for incompressible flows under all conditions," ser. AIAA Paper 2006-0689, 2006.

[71] B. T. Helenbrook and G. W. Cowles, "Preconditioning for dual-time-stepping simulations of the shallow water equations including Coriolis and bed friction effects," *Journal of Computational Physics*, vol. 227, no. 9, pp. 4425–4440, 2008.

[72] F. Sicot, G. Puigt, and M. Montagnac, "Block-Jacobi implicit algorithm for the time spectral method," *AIAA Journal*, vol. 46, no. 12, pp. 3080–3089, 2008.

[73] J. Sitaraman, "CFD based unsteady aerodynamic modeling for rotor aeroelastic analysis," Ph.D. dissertation, University of Maryland, 2003.

[74] R. Singh and J. Baeder, "Generalized moveing gust response using CFD with application to airfoil-vortex interaction," ser. AIAA Paper 1997-2208, 1997.

[75] A. Zaide and D. Raveh, "Numerical simulation and reduced-order modeling of airfoil gust response," *AIAA Journal*, vol. 44, pp. 1826–1834, 2006.

[76] D. Jones, C. Wales, and A. Gaitonde, "Simulation of airfoil gust responses using prescribed velocities," ser. IFASD, 2011.

[77] C. Wales, D. Jones, and A. Gaitonde, "Reduced order modelling for aeroelastic aerofoil response to a gust," ser. AIAA Paper 2013-0790, 2013.

[78] AGARD, "Compendium of unsteady aerodynamic measurements," Tech. Rep. No. 702, 1982.

[79] K. Isogai, "Transonic dip mechanism of flutter of a sweptback wing: Part II," *AIAA Journal*, vol. 17, pp. 1240–1242, 1981.

[80] B. Szabó and I. Babuška, *Finite Element Analysis*. John Wiley & Sons, 1991.

[81] Z. Yang and D. J. Mavriplis, "Time spectral method for quasi-periodic unsteady computation on unstructured meshes." ser. AIAA Paper 2010-5034, June 2010.

[82] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Modeling viscous transonic limit-cycle oscillation behavior using a harmonic balance approach," *Journal of Aircraft*, vol. 41, no. 6, pp. 1266–1274, 2004.

[83] P. D. Thomas and C. K. Lombard, "Geometric conservation law and its application to flow computations on moving grids," *AIAA Journal*, vol. 17, no. 10, pp. 1030–1037, 1979.

[84] M. Lesoinne and C. Farhat, "Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations," *Computer Methods in Applied Mechanics and Engineering*, vol. 134, no. 1-2, pp. 71–90, 1996.

[85] B. Koobus and C. Farhat, "Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 170, no. 1-2, pp. 103–129, 1999.

[86] H. Guillard and C. Farhat, "On the significance of the geometric conservation law for flow computations on moving meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 11-12, pp. 1467–1482, 2000.

[87] C. Farhat, P. Geuzaine, and C. Crandmont, "The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids," *Journal of Computational Physics*, vol. 174, no. 2, pp. 669–694, 2001.

[88] 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*, vol. 213, no. 2, pp. 557–573, 2006.

[89] J. I. Leffell, S. M. Murman, and T. H. Pulliam, "An extension of the time-spectral method to overset solvers," ser. AIAA Paper 2013-0637, Jan. 2013.

[90] B. Cockburn, S. Hou, and C.-W. Shu, "The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case," *Mathematics of Computation*, vol. 54, pp. 545–581, 1990.

[91] B. Cockburn and C.-W. Shu, "The Runge-Kutta discontinuous Galerkin method for conservation laws, v: Multidimensional systems," *Journal of Computational Physics*, vol. 141, no. 2, p. 199âĂŞ224, 1998.

[92] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, *Discontinuous Galerkin Methods. Theory, Computation and Applications*, ser. Lecture Notes in Computational Science and Engineering. Springer-Verlag, 2000, vol. 11.

[93] D. S. Kershaw, M. K. Prasad, M. J. Shaw, and J. L. Milovich, "3D unstructured mesh ale hydrodynamics with the upwind continuous finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 158, no. 1, pp. 81–116, 1995.

[94] B. Cockburn and C.-W. Shu, "TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws.ii: General framework," *Mathematics of Computation*, vol. 22, pp. 411–435, 1989.

[95] A. Burbeau, P. Sagaut, and C.-H. Bruneau, "A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods," *Journal of Computational Physics*, vol. 169, no. 1, pp. 111–150, 2001.

[96] T. J. Barth, *Numerical Methods for Gasdynamic Systems on Unstructured Meshes*, ser. Lecture Notes in Computational Science and Engineering. Springer-Verlag, 1998, vol. 5.

[97] C. E. Baumann, "An hp-adaptive discontinuous finite element method for computational fluid dynamics," Ph.D. dissertation, University of Texas at Austin, 1997.

[98] B. Cockburn and P. A. Gremaud, "Error estimates for finite element methods for nonlinear conservation laws," *SIAM Journal on Numerical Analysis*, vol. 33, no. 2, pp. 522–554, 1996.

[99] J. Jaffre, C. Johnson, and A. Szepessy, "Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws," *Mathematical Models and Methods in Applied Sciences*, vol. 5, no. 3, pp. 367–386, 1995.

[100] M. J. Brazell and B. T. Helenbrook, "$p = 2$ continuous finite elements on tetrahedra with local mass matrix inversion to solve the preconditioned compressible NavierâĂŞStokes equations," *Computers & Fluids*, vol. 88, pp. 643–652, 2013.

[101] D. Gottlieb and J. S. Hesthaven, "Spectral methods for hyperbolic problems," *Journal of Computational and Applied Mathematics*, vol. 128, no. 1-2, pp. 84–131, 2001.

[102] E. Tadmor, "Convergence of spectral methods for nonlinear conservation laws," *SIAM Journal of Numerical Analysis*, vol. 26, no. 1, pp. 30–44, 1989.

[103] A. Gelb and E. Tadmor, "Enhanced spectral viscosity approximations for conservation laws," *Applied Numerical Mathematics*, vol. 33, no. 1, pp. 3–21, 2000.

[104] Y. Maday and E. Tadmor, "Analysis of the spectral vanishing viscosity method for periodic conservation laws," *SIAM Journal on Numerical Analysis*, vol. 26, no. 4, pp. 854–870, 1989.

[105] C. Xu and R. Pasquetti, "Stabilized spectral element computations of high reynolds number incompressible flows," *Journal of Computational Physics*, vol. 196, no. 2, pp. 680–704, 2004.

[106] H. Huang and K. Ekici, "Stabilization of the high-dimensional harmonic balance solvers using a temporal spectral viscosity operator," ser. AIAA Paper 2013-0354, 2013.

[107] R. C. Maple, "Multigrid for adaptive harmonic balance CFD," ser. 16th AIAA Computational Fluid Conference, June 2003.

[108] R. C. Maple, P. I. King, and M. E. Oxley, "Adaptive harmonic balance solutions to Euler's equations," *AIAA Journal*, vol. 41, no. 9, pp. 1705–1714, 2003.

[109] R. C. Maple, P. I. King, P. D. Orkwis, and J. M. Wolff, "Adaptive harmonic balance method for time-periodic flows," *Journal of Computational Physics*, vol. 193, no. 2, pp. 620–641, 2004.

[110] A. Mosahebi and S. K. Nadarajah, "An adaptive non-linear frequency domain method for viscous periodic steady state flows," ser. AIAA Paper 2010-1267, Jan. 2010.

[111] ——, "An implicit adaptive non-linear frequency domain method (pNLFD) for viscous periodic steady state flows on deformable grids," ser. AIAA Paper 2011-0775, Jan. 2011.

[112] J. P. Thomas, C. Custer, E. H. Dowell, K. C. Hall, and C. Corre, "Compact implementation strategy for a harmonic balance method within implicit flow solvers," *AIAA Journal*, vol. 51, no. 6, pp. 1374–1381, 2013.

[113] P. Bar-Yoseph, "Time finite element methods for initial value problems," *Applied Numerical Mathematics*, vol. 33, no. 1-4, pp. 435–445, 2000.

[114] P. Bar-Yoseph and D. Elata, "An efficient L_2 Galerkin finite element method for multidimensional nonlinear hyperbolic systems," *International Journal of Numerical Methods in Engineering*, vol. 29, no. 6, pp. 1229–1245, 1990.

[115] P. Bar-Yoseph, D. Elata, and M. Israeli, "On the generalized L_2 Galerkin finite element method for linear hyperbolic equations," *International Journal of Numerical Methods in Engineering*, vol. 36, no. 4, pp. 679–694, 1993.

[116] G. M. Hulbert, "Time finite element methods for structural dynamics," *International Journal of Numerical Methods in Engineering*, vol. 33, no. 2, pp. 307–331, 1992.

[117] X. D. Li and N.-E. Wiberg, "Structural dynamic analysis by a time-discontinuous Galerkin finite element method," *International Journal of Numerical Methods in Engineering*, vol. 39, no. 12, pp. 2131–2152, 1996.

[118] M. Calvo, S. González-Pinto, and J. I. Montijano, "On the iterative solution of the algebraic equations in fully implicit Runge-Kutta methods," *Numerical Algorithms*, vol. 23, pp. 97–113, 2000.

[119] J. Butcher, "Implicit Runge-Kutta processes," *Mathematics of Computation*, vol. 18, pp. 50–64, 1964.

[120] J. C. Butcher, "On the implementation of implicit Runge-Kutta methods," *BIT Numerical Mathematics*, vol. 16, pp. 237–240, 1976.