

# A Block-Jacobi Time-Spectral Method For Incompressible Flow

by

Alton James Luder III

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Naval Architecture and Marine Engineering)  
in The University of Michigan  
2013

Doctoral Committee:

Assistant Professor Kevin Maki, Chair  
Professor Robert Beck  
Professor Robert Krasny  
Professor Eric Paterson, Virginia Polytechnic Institute and State University  
Professor Armin Troesch

# TABLE OF CONTENTS

<b>LIST OF FIGURES</b> . . . . .	iv
<b>LIST OF TABLES</b> . . . . .	vi
<b>LIST OF APPENDICES</b> . . . . .	vii
<b>LIST OF ABBREVIATIONS</b> . . . . .	viii
<b>CHAPTER</b>	
<b>I. Introduction</b> . . . . .	1
1.1 Time-Periodic Flows . . . . .	2
1.2 Thesis Contributions . . . . .	6
<b>II. Time-Spectral Algorithm</b> . . . . .	8
2.1 Time-Spectral Derivation . . . . .	8
2.2 Formation of the Fully Time-Coupled Matrix . . . . .	11
2.2.1 Fully Time-Coupled Matrix . . . . .	12
2.2.2 Linearization of Convection . . . . .	18
2.3 Block-Jacobi Algorithm . . . . .	18
<b>III. Scalar Transport and Burgers' Equation</b> . . . . .	28
3.1 Linear Advection-Diffusion . . . . .	28
3.1.1 Linear Advection . . . . .	30
3.1.2 Linear Diffusion . . . . .	31
3.1.3 Backward-Facing Step Test Case . . . . .	34
3.2 Burgers' Equation . . . . .	36
3.2.1 One-Dimensional Oscillating Inflow . . . . .	37
3.2.2 Backward-Facing Step with Burgers' Equation . . . . .	39
<b>IV. Incompressible Laminar Flow</b> . . . . .	46

4.1	Time-Spectral Algorithm for the Navier-Stokes Equations . . .	46
4.2	Backward-Facing Step . . . . .	50
4.3	Pitching Foil . . . . .	53
4.4	Plunging Foil . . . . .	59
<b>V. Incompressible Turbulent Flow . . . . .</b>		<b>64</b>
5.1	Time-Spectral Algorithm for the URANS Equations . . . . .	64
5.2	Pitching Foil . . . . .	67
5.3	Plunging Foil . . . . .	70
5.4	Propeller in Shear Flow . . . . .	74
<b>VI. Conclusions . . . . .</b>		<b>83</b>
6.1	Accomplishments . . . . .	84
6.2	Future Work . . . . .	85
<b>APPENDICES . . . . .</b>		<b>88</b>
A.1	Power-Sum Simplification . . . . .	89
B.1	Von Neumann Analysis . . . . .	93
<b>BIBLIOGRAPHY . . . . .</b>		<b>100</b>

## LIST OF FIGURES

### Figure

2.1	Geometry for Finite Difference Example Matrix . . . . .	14
2.2	Predicted versus Tested Under-Relaxation Values for Three Time-Levels . . . . .	24
3.1	Spectral Accuracy versus Second Order Accuracy for Linear Advection	31
3.2	Spectral Accuracy versus Second Order Accuracy for Linear Diffusion	33
3.3	Backward-Facing Step Geometry for Linear Advection-Diffusion . .	34
3.4	Streamlines for Advection Field of Backward-Facing Step Test Case	34
3.5	Linear Advection-Diffusion Results over Backward-Facing Step for Horizontal Line . . . . .	35
3.6	One Dimensional Burgers' Equation Simulation . . . . .	38
3.7	Convergence of the Fully Time-Coupled Matrix Due to Iterations Between Nonlinear Updates . . . . .	39
3.8	Backward-Facing Step Geometry for Burgers' Equation . . . . .	39
3.9	Spectral Accuracy versus First Order Accuracy over Backward-Facing Step with Burgers' Equation . . . . .	41
3.10	Spectral Accuracy versus Second Order Accuracy over Backward-Facing Step with Burgers' Equation . . . . .	42
3.11	Discretization versus Cost for Burgers' Equation . . . . .	43
3.12	Error at a Point versus Cost for Burgers' Equation . . . . .	45
4.1	Backward-Facing Step Geometry for the Navier-Stokes Equations .	51
4.2	Results for Convergence of $u$ at a Point over a Backward-Facing Step	52
4.3	Backward-Facing Step Work Comparison . . . . .	52
4.4	Foil Domain . . . . .	54
4.5	Force Polar For Laminar Pitching Foil . . . . .	55
4.6	$C_p$ Distribution For Laminar Pitching Foil at $T = 0$ . . . . .	56
4.7	Comparison of Forces as Function of Cost . . . . .	58
4.8	Force Polar For Laminar Plunging Foil . . . . .	60
4.9	$C_p$ Distribution For Laminar Plunging Foil at $\sin \omega t = 0$ . . . . .	61
4.10	Vorticity Near the Trailing Edge of a Pitching Airfoil . . . . .	61
4.11	Comparison Of Forces as Function of Processing Cost . . . . .	63
5.1	Drag versus Lift for Turbulent Pitching Foil . . . . .	69
5.2	$C_p$ Comparison for Turbulent Pitching Foil . . . . .	70

5.3	$C_p$ Comparison for Highly Resolved Time-Spectral and BDF Case . . . . .	72
5.4	$C_p$ Comparison for BDF Case with Varying Resolution . . . . .	73
5.5	Lift versus Drag Polar for Turbulent Plunging Foil . . . . .	74
5.6	Shear Flow at Inlet . . . . .	76
5.7	Force over a Cycle for Propeller Blade in Shear Flow . . . . .	78
5.8	$K_T$ versus $K_Q$ for Propeller Blade in Shear Flow . . . . .	79
5.9	Pressure On Propeller Surface at Time $\sin \omega t = 0$ . . . . .	80
5.10	Force Convergence Comparison of Propeller at Time $\sin \omega t = 0$ . . . . .	82

## LIST OF TABLES

### Table

3.1	Parameters for the Two-Dimensional Burgers' Equation Test Case .	40
4.1	Parameters for the Backward-Facing Step Test Case for the Navier-Stokes Equations . . . . .	51
4.2	Parameters for the Navier-Stokes Equations Pitching Airfoil Case .	54
5.1	Parameters for the Turbulent Pitching Airfoil Case . . . . .	68
5.2	Parameters for the Propeller in Shear Flow . . . . .	75

## LIST OF APPENDICES

### Appendix

A.	Power-Sum Simplification . . . . .	89
B.	Von Neumann Stability Analysis . . . . .	93

## LIST OF ABBREVIATIONS

- ALE** Arbitrary Lagrangian Eulerian
- BDF** Backward Difference Formula
- CFD** Computational Fluid Dynamics
- LES** Large Eddy Simulations
- OpenFOAM<sup>®</sup>** Open Field Operation And Manipulation
- PDE** Partial Differential Equation
- PISO** Pressure Implicit Split Operators
- SIMPLE** Semi-Implicit Method for Pressure-Linked Equations
- URANS** Unsteady Reynolds Averaged Navier-Stokes



# CHAPTER I

## Introduction

Fluid flow is significant for many areas of engineering interest. For this reason, accurate modeling of fluid flow can yield dramatic improvements in engineering application areas. These improvements can be from a better way to understand fluid flow a better way to analyze designs. A more thorough comprehension of fluid flow can lead to novel designs. For example, flapping wings often produce a Leading Edge Vortex (LEV) that can delay the onset of stall as compared to fixed wings (Shyy et al., 2008). This allows for a design with much higher lift than would be expected from steady analysis. Accurate portrayal of fluid flow is also important in design analysis. Accuracy of evaluation techniques is essential for helping designers choose between competing designs and making incremental improvements to existing designs. Unsteady fluid flow at high Reynolds number is particularly difficult to describe. High Reynolds number flows exhibit turbulence that can be challenging to model both experimentally and numerically. Numerical approaches are more and more preferred as a way to model fluid flow because computational resources are continually increasing. Computational Fluid Dynamics (CFD) in particular is a numerical approach that is popular because it provides full flow field information. Being rooted in the Navier-Stokes equations, CFD is a physics-based approach that continues to prove itself as a staple engineering tool. CFD does, however, tend to be computationally expen-

sive for unsteady computations. This is due to Courant number limitations with the most commonly available time-marching methods. For this reason, many engineering problems are cost prohibitive to solve using conventional time-marching with CFD. For fluid flow of a periodic nature, such as propellers in shear flow, this unsteadiness is of particular importance as fluctuations in the flow are repetitive and therefore have a recurring effect on performance. The Time-Spectral Method is an alternative to conventional time-marching methods that can exploit the time-periodic nature of these flows to allow an accurate and efficient time-discretization scheme. This thesis will introduce and demonstrate the Time-Spectral Method for periodic single-phase Reynolds' averaged turbulent incompressible flows.

## 1.1 Time-Periodic Flows

The Time-Spectral Method is a different numerical approximation for the time-derivative than more conventional time-marching techniques such as the Backward Difference Formula (BDF). Specifically, the Time-Spectral Method utilizes sinusoidal basis functions instead of polynomial basis functions for discretizing the time-derivative term of the Navier-Stokes Equations or other Partial Differential Equations (PDEs) of interest. As sinusoids are periodic, the Time-Spectral Method requires a periodic assumption. Due to this different formulation, the Time-Spectral Method can be faster and more accurate for the capture of periodic flows. The first way the Time-Spectral Method is faster is that it does not compute the transients and directly computes the periodic steady state. This saves computational time by only computing the desired end state. The second way the Time-Spectral Method saves computational time over time-marching CFD is to use fewer time levels within a period. Conventional time-marching methods must compute much higher frequency information for stability, thus spending time computing unwanted information. The Time-Spectral Method has spectral accuracy (Jameson, 2009), which allows dramat-

ically fewer time-levels per period for the same level of accuracy. This becomes especially important for high-fidelity models. The current Time-Spectral Method utilizes the same flux and viscosity terms already well refined for CFD applications, which are fast, accurate, and parallelizable. This allows extended CFD capability by the research of the time-derivative term alone, making the Time-Spectral Method a valuable computation tool.

Due to the assumption of periodicity, the Time-Spectral Method is only useful with flows that exhibit periodic behavior. These flows are extremely common as motors take energy and convert it into a rotational motion while turbines extract fluid energy into a rotational motion. Examples of rotational propulsion are propellers and waterjets. Propellers in particular can have very strong periodic effects as the blades rotate into and out of the viscous wake of a ship. Translational periodicity is important to understand and replicate wildlife propulsion, such as the swimming motion of a fish or the flapping wings of a bird. For industrial uses, turbines often convert fluid energy into rotational motion, which can then be converted into electrical energy. Examples of this are water turbines and wind turbines. Both types of turbines are in continual development for their ability to generate power using renewable resources. The Time-Spectral Method has use in a wide range of engineering fields.

The current Time-Spectral Method can be traced back to using sinusoidal functions for periodic flows linearized about a mean flow condition (Hall and Crawley, 1989). This required the periodic component of the flow to be small compared to the mean flow, which limits its applicability. Hall et al. (2002) introduced the harmonic balance method, which removed this limitation by the transformation of the time derivative and the spatial terms to the frequency domain. This method allowed the addition of viscous flux terms, which had not been used previously. This paper also reported that the computation of fluxes in the frequency domain was less efficient than first computing fluxes in the time domain then, subsequently, transforming them

to the frequency domain. As such, the fluxes had to be transformed to the frequency domain to take the time derivative, then the updated primitive variables had to be inversely transformed back to the time domain to find the fluxes for the next update. This method has been extended to multistage turbomachinery (Ekici and Hall, 2007), Horizontal Axis Wind Turbines (HAWT) (Jackson et al., 2011), and also has been used with helicopters in steady flight (Ekici et al., 2008).

Gopinath and Jameson (2005) used a transformation to implicitly put the harmonic balance equations back into the time domain. This saved the cost of transforming the fluxes and primitive variables between the time domain and frequency domain. In this formulation, the frequency domain is found implicitly via the derivation of the time derivative term (Canuto et al., 2007). The Time-Spectral Method has been used to simulate a pitching foil (Gopinath and Jameson, 2005), turbomachinery (van der Weide et al., 2005), vertical-axis wind turbine (Vassberg et al., 2005), multistage turbomachinery (Gopinath et al., 2007), and rotorcraft (Butsunorn and Jameson, 2008). The Time-Spectral Method has been extended to solve flows with unknown time periods via the Gradient Based Variable Time Period (GBVTP) approach (Gopinath and Jameson, 2006). The Gradient Based Variable Time Period has been used to solve flow past a circular cylinder. Yang and Mavriplis (2010) combined the Time Spectral Method with the BDF to look at flows with both periodic and non-periodic components. These computational methods were applied to helicopter maneuvering by Yang et al. (2011). Mavriplis et al. (2012) extended this capability to overlapping mesh for helicopter maneuvering simulations.

Much of the development of the Time-Spectral Method has included an investigation into stability. Gopinath and Jameson (2006) as well as van der Weide et al. (2005) reported an issue with even-odd decoupling when an even number of time intervals was used. For cases with relatively small time derivatives, this discrepancy did not destabilize the solution. For cases that had relatively larger time derivatives, however, the

even-odd decoupling introduced instabilities that caused the Time-Spectral algorithm to diverge. These investigations all used dual-time stepping, which is a method where a pseudo-time derivative is introduced so that the original governing equations are recovered at pseudo-time steady-state. Conventional steady-state numerical techniques can then be used to accelerate computation to pseudo-time steady-state. Gopinath and Jameson (2005) used a Von Neumann analysis of the pseudo-time term for their dual time-stepping technique. They reported that, for a given spatial operator, the addition of Time-Spectral physical time discretization added only an imaginary component to the amplification factor of pseudo-time marching. Thus, when marching to a pseudo-time steady-state, it would be faster to use an implicit pseudo-time method, which tends to have a larger region of stability than the previously used explicit pseudo-time method. Sicot et al. (2008) utilized a block-Jacobi technique to enable implicit pseudo-time derivatives for enhanced convergence to pseudo-time steady-state. Prior to this technique, only explicit pseudo-time marching schemes had been used. This technique greatly sped up computations for dual-time stepping schemes. Mavriplis and Yang (2011) followed this technique, but utilized a Newton scheme to speed up the block-Jacobi solving. Su and Yuan (2010) noticed that the addition of the Time-Spectral terms weakened diagonal dominance and, hence, negatively affected convergence and stability. They overcame this issue by the use of a multigrid preconditioned Generalized Minimum Residual (GMRES) linear system solver, which is less sensitive to diagonal dominance than the block-Jacobi methods used by Sicot et al. (2008).

Incompressible flow is mostly untried with the Time-Spectral Method. Currently, there are only three recent references on incompressible Time-Spectral/harmonic-balance methods. Jameson (2009) used artificial compressibility to simulate flapping wings. Because artificial compressibility made the equations essentially hyperbolic, he was able to use standard compressible flow methods in order to solve the resulting

equations. Antheaume and Corre (2011) followed Jameson (2009) in using artificial compressibility to obtain a divergence free flow. This method compared the Time-Spectral Method against BDF for a two-dimensional airfoil in pitch-heave motion using an Arbitrary Lagrangian Eulerian (ALE) formulation for grid motion. They only studied flows at a Reynolds number of 1100. Antheaume and Corre (2011) showed up to 5 times improvement in computational time over the backward difference formula (BDF). Welch et al. (2005) used the vorticity-transport/stream-function approach instead of relying on primitive variables. This approach was used to simulate two-dimensional jets in cross flow.

## 1.2 Thesis Contributions

The work of this thesis has extended computational methods to utilize the Time-Spectral Method for pressure projection methods via block-Jacobi solving with under-relaxation stabilization. While much of the earlier Time-Spectral work of this thesis was to understand the Time-Spectral Method by using the Linear Advection-Diffusion Equation and Burgers' Equation, the Time-Spectral Method has been utilized with the incompressible flow equations via pressure projection methods. The Time-Spectral Method has been extended to turbulent incompressible flow. As discussed above, a Time-Spectral Method has been used before with incompressible flow. However, the approaches taken before were not based on pressure projection methods such as PISO or SIMPLE. Pressure projection methods are the dominant form of solving the incompressible Navier-Stokes Equations, thus it is important to be able to use these methods for solving periodic problems with the Time-Spectral Method. The prior methods were based on either the artificial compressibility approach or the vorticity stream function approach. The vorticity stream function approach is generally limited to two dimensions, thus limiting its applicability. The artificial compressibility approach unnecessarily time-couples the pressure variable in a way

that is unphysical. Because of this, convergence can be slowed.

The block-Jacobi method is used to solve the fully time-coupled system of equations for the incompressible Navier-Stokes Equations. This is useful as traditional CFD solvers only expect to see one time-level at a time. The block-Jacobi method is discussed further in Section 2.3. Under-relaxation is used in order to provide diagonal dominance for the block-Jacobi linear-systems solver. Under-relaxation is a common stabilization method for pressure projection methods, but has not been used with the Time-Spectral Method before this thesis. This stands apart from the prior Time-Spectral implementations as dual-time stepping was previously used for stability. The success of using the Time-Spectral Method for incompressible flow led to the addition of turbulence modeling. Unsteady Reynolds Averaged Navier-Stokes (URANS) turbulence modeling is employed because it averages the velocity fields which smooths the solution in time. These contributions have extended the Time-Spectral Method to incompressible turbulent flows.

## CHAPTER II

# Time-Spectral Algorithm

The Time-Spectral Algorithm is derived by finding a time-derivative based on the discrete representation of a continuous function. This discrete time derivative represents an algebraic relationship between time derivatives at a time level and the values of that variable at other time levels. The time derivative term of the Time-Spectral Method has different implementation than the more common time marching methods such as BDF. In particular, the Time-Spectral Method requires coupling all the time levels. A fully time-coupled matrix is used to accomplish this coupling. The block-Jacobi algorithm is introduced in order to solve this fully time-coupled matrix. Under-relaxation is used to stabilize the block-Jacobi algorithm. Lastly, under-relaxation is compared to dual-time stepping, which is common in literature for compressible flow. These steps form the background for using the Time-Spectral Method with incompressible flow.

### 2.1 Time-Spectral Derivation

CFD methods are based around solving numerical approximations of fluid flow equations on a discretized field. For a time-periodic function it makes sense to represent the numerical approximation via Fourier basis functions as Fourier basis functions are periodic in nature. This periodicity in function  $\phi(t)$  is represented by the equa-



tion  $\phi(t) = \phi(t + T)$  where  $T$  is the fundamental period. As  $\phi$  will later be used for discrete analysis, only the discrete formulation of  $\phi$  will be considered. Doing this,  $\phi$  at the  $n$ -th discrete time level ( $t = n\Delta t$ ) can be expressed:

$$\phi(n\Delta t) \approx \phi^n = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \hat{\phi}_k e^{ik\frac{2\pi}{T}n\Delta t} \quad (2.1.1)$$

with the inverse discrete Fourier Transform

$$\hat{\phi}_k = \frac{1}{N} \sum_{n=0}^{N-1} \phi^n e^{-ik\frac{2\pi}{T}n\Delta t} \quad (2.1.2)$$

where  $\hat{\phi}_k$  is the  $k$ -th Fourier Coefficient and  $i$  is the imaginary number. Equations 2.1.1 and 2.1.2 are the discrete formulations of the continuous function  $\phi$  where  $N$  is the total number of time-steps and is assumed odd. This thesis does not consider when  $N$  is even due to the even-odd decoupling found by Gopinath and Jameson (2006). Note that  $n\Delta t = t$  so  $N\Delta t = T$ . Taking the time derivative of Equation 2.1.1,  $\partial\phi/\partial t$  is expressed as:

$$\frac{\partial\phi^n}{\partial t} = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} ik\frac{2\pi}{T} \hat{\phi}_k e^{ik\frac{2\pi}{T}n\Delta t} \quad (2.1.3)$$

Substituting the inverse discrete Fourier Transform from Equation 2.1.2 into the time derivative term yields:

$$\frac{\partial\phi^n}{\partial t} = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} ik\frac{2\pi}{TN} e^{ik\frac{2\pi}{T}n\Delta t} \sum_{m=0}^{N-1} \phi^m e^{-ik\frac{2\pi}{T}m\Delta t} \quad (2.1.4)$$

$$\frac{\partial\phi^n}{\partial t} = \frac{2\pi}{T} \sum_{m=0}^{N-1} \phi^m \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\frac{2\pi}{N}(n-m)} \quad (2.1.5)$$

Notice that  $\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} (ik/N) e^{ik\frac{2\pi}{N}(n-m)} = 0$  when  $m = n$ . Equation 2.1.5 is simplified further through the use of a power-sum simplification, as well as trigonometric identities. The details of this simplification are found in Appendix A. Resulting from this simplification is the following formula:

$$\frac{\partial \phi^n}{\partial t} = \frac{2\pi}{T} \sum_{m=0}^{N-1} \phi^m \frac{(-1)^{n-m}}{2} \operatorname{csc} \left( \frac{2\pi(n-m)}{2N} \right) \quad (2.1.6)$$

Each term in this summation can be split into a coefficient and the unknown at each time level. The coefficient is based on the distance between two time levels. As the last summation in Equation 2.1.5 equals 0 when  $m = n$ , the singularity of the csc term is avoided and the temporal relationships are expressed as:

$$\frac{\partial \phi^n}{\partial t} = \sum_{m=0}^{N-1} d_{n-m} \phi^m \quad (2.1.7a)$$

$$d_{n-m} = \begin{cases} \frac{2\pi}{T} \frac{1}{2} (-1)^{n-m} \operatorname{csc} \left( \frac{\pi(n-m)}{N} \right) & n \neq m \\ 0 & n = m \end{cases} \quad (2.1.7b)$$

The Time-Spectral Method can be used in CFD by the substitution of the Time-Spectral operator for the time-derivative in the governing equations. An example would help to show how the Time-Spectral operator forms relationships between time levels of a periodic problem. The PDE for  $\phi$  is expressed as a time derivative and remaining spatial terms ( $R(\phi)$ ), the equation for  $\phi$  can be written:

$$\frac{\partial \phi}{\partial t} + R(\phi) = 0 \quad (2.1.8)$$

For this example,  $\phi$  will be discretized with three time-levels per period, using  $\phi^n$

to represent  $\phi(n\Delta t)$ . The algebraic relationships at the current time level are each expressed:

$$-\frac{\pi}{T} \csc\left(\frac{\pi}{3}\right) \phi^{n-1} + \frac{\pi}{T} \csc\left(\frac{\pi}{3}\right) \phi^{n+1} + R(\phi^n) = 0 \quad (2.1.9)$$

where  $\phi^{n+3} = \phi^n = \phi^{n-3}$  due to the periodicity assumption. The Time-Spectral Method is compared to the standard second-order BDF, which has no periodicity assumption. For constant time step, the BDF formula is:

$$\frac{3\phi^n - 4\phi^{n-1} + \phi^{n-2}}{2\Delta t} + R(\phi^n) = 0 \quad (2.1.10)$$

In the Time-Spectral formulation,  $\Delta t$  is only present by the relationship  $T = N\Delta t$ . Also, the Time-Spectral Method looks both forward and backward in time, whereas the Backward Difference Formula only looks at the current time level and at previous time levels. The physical interpretation of this is that if there is a disturbance in the flow at a point just ahead of the current time level, the Time-Spectral formulation assumes that disturbance to have happened at all previous cycles. This follows from the periodic assumption and is why the physical requirement that future times cannot affect a past time may seem to be violated. It is worth noting that the Time-Spectral formulation has no term at the current time level while BDF does. This pole term is important for solution methods, particularly iterative algorithms and pressure projection methods. Certain iterative solution algorithms require diagonal dominance which requires a strong pole term. Pressure projection methods will sometimes use the pole term when updating the pressure field.

## 2.2 Formation of the Fully Time-Coupled Matrix

As the Time-Spectral formulation is a function of all time levels, the time-levels must be solved concurrently. Solving this system requires a matrix that is fully time-coupled. As the time-derivative and spatial-derivative terms are separate, spatial-

derivative terms can be discretized using readily available finite-difference, finite-element, or finite-volume techniques. These spatial terms may be specified for each term in the PDE of interest. Many specialized schemes exist for each term of the Navier-Stokes Equations, such as limiting schemes for advection. These schemes remain available for use with the Time-Spectral Method. For explicit time-marching schemes, the values of  $\phi$  used are at the previous time-level, and thus are available to be directly evaluated. For implicit numerical schemes, the resulting equations represent relationships between variables that are put into a linear system of equations. This linear system is then solved to find the discretized solution to the PDE of interest. Implicit schemes are common for incompressible flow due to pressure velocity coupling. PISO or SIMPLE are two common algorithms that maintain pressure velocity coupling for incompressible flow. Both of these algorithms require implicit matrix solves for both pressure and velocity. Following these algorithms, only implicit numerical schemes are used in this thesis.

### 2.2.1 Fully Time-Coupled Matrix

The fully time coupled matrix can be formed by simply combining the relationships formed by the time-derivative terms and the spatial-derivative terms into a linear system of equations. The Time-Spectral Method makes this easily visualizable as the spatial terms are exclusively at the current time level while Time-Spectral terms are exclusively at time levels other than the current time. This lack of a term at the current time level can be seen in the null coefficient of the Time-Spectral Method in Equation 2.1.7. The spatial system of equations at time  $n$  can be represented by the block matrix  $R^n$ , and the time derivatives can be represented by the block matrix  $P_{n-m}$ . For three time levels, a fully time-coupled matrix can be expressed in block

form as follows:

$$\begin{bmatrix} R^0 & P_{-1} & P_1 \\ P_1 & R^1 & P_{-1} \\ P_{-1} & P_1 & R^2 \end{bmatrix} \begin{Bmatrix} \phi^0 \\ \phi^1 \\ \phi^2 \end{Bmatrix} = \begin{Bmatrix} R_S^0 \\ R_S^1 \\ R_S^2 \end{Bmatrix} \quad (2.2.1)$$

Here  $R_S^n$  represents the source terms due to boundary conditions. From Equation 2.1.7 it can be shown that  $P_{n-m}$  is a diagonal matrix with  $d_{n-m}$  for each diagonal value. That is:

$$P_{n-m} = d_{n-m} \bar{I} \quad (2.2.2)$$

where  $\bar{I}$  is the identity matrix. The fully time coupled linear system of equations must satisfy standard linear algebra nonsingular matrix requirements.

A specific example is presented to clarify the fully time-coupled matrix. To simplify things, the analysis is kept to one dimension and basic discretization schemes. The PDE to be solved is the linear advection-diffusion equation which is:

$$\frac{\partial \phi}{\partial t} + c \frac{\partial \phi}{\partial x} - \nu \frac{\partial^2 \phi}{\partial x^2} = 0 \quad (2.2.3)$$

In this formulation,  $c$  and  $\nu$  are both considered constant and positive. This will first be looked at in a finite difference framework. This keeps the equations in differential form. As implicit schemes are being investigated, these will all be formulated for the current time level. This is represented by  $\phi^n$ , where  $n$  represents the  $n - th$  discrete time level. Similarly, the subscript  $i$  will be used to represent the spatial position. Convection is discretized with the upwind scheme and diffusion is discretized with the linear scheme as follows:

$$c \frac{\partial \phi}{\partial x} = c \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} \quad (2.2.4)$$

$$\nu \frac{\partial^2 \phi}{\partial x^2} = \nu \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{\Delta x^2} \quad (2.2.5)$$

For an interior point, these discretizations result in the following equation:

$$\frac{\partial \phi}{\partial t} + c \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} - \nu \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{\Delta x^2} \quad (2.2.6)$$

The time derivative is being left in continuous form as both the Time-Spectral Method and BDF will be used. For the boundary conditions, a fixed value boundary condition will be applied at the inlet while a zero-gradient condition will be applied at the outlet as follows:

$$\phi_0^n = A \sin(\omega n \Delta t) \quad (2.2.7)$$

$$\left. \frac{\partial \phi^n}{\partial x} \right|_{x=3\Delta x} = 0 \quad (2.2.8)$$

Having this information, a simple four point domain can be constructed to demonstrate the spatial matrix  $R^n$  from Equation 2.2.1 and then the fully time-coupled matrix. The domain is shown in Figure 2.1.

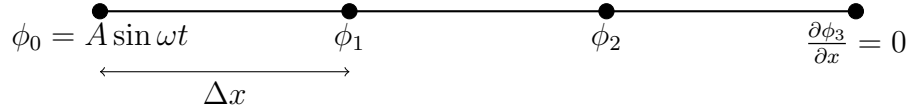


Figure 2.1: Geometry for Finite Difference Example Matrix

$$R^n \phi^n = \left( \frac{c}{\Delta x} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{\nu}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 2 & -2 \end{bmatrix} \right) \begin{Bmatrix} \phi_1^n \\ \phi_2^n \\ \phi_3^n \end{Bmatrix} \quad (2.2.9)$$

And the source term  $R_S^n$  is as follows:

$$R_S^n = - \begin{Bmatrix} \frac{-c}{\Delta x} A \sin(\omega n \Delta t) \\ 0 \\ 0 \end{Bmatrix} + \begin{Bmatrix} \frac{\nu}{\Delta x^2} A \sin(\omega n \Delta t) \\ 0 \\ 0 \end{Bmatrix} \quad (2.2.10)$$

This matrix and source term can be simplified by using  $\gamma = c/\Delta x$  and  $\zeta = \nu/\Delta x^2$ .

$$R^n = \left( \begin{bmatrix} \gamma + 2\zeta & -\zeta & 0 \\ -\gamma - \zeta & \gamma + 2\zeta & -\zeta \\ 0 & -2\zeta & 2\zeta \end{bmatrix} \right) \quad (2.2.11)$$

$$R_S^n = \left\{ \begin{array}{c} (\gamma + \zeta)A \sin(\omega n \Delta t) \\ 0 \\ 0 \end{array} \right\} \quad (2.2.12)$$

Now that spatial matrix block and source term are defined, and the Time-Spectral Method is defined from Equation 2.2.2, the fully time-coupled matrix is shown in Equation 2.2.13. This uses  $d_1 = \frac{-\pi}{T} \csc\left(\frac{\pi}{3}\right)$  for this example with 3 time levels.

Similarly, the BDF time-marching scheme can be shown in the same block matrix form. Here four time levels are used and initial conditions provide the extra time level. In order to start the algorithm, the backward Euler time stepping formula was used for the first time step. This is seen in Equation 2.2.14. The matrices in Equations 2.2.13 and 2.2.14 show very different time derivative properties. Firstly, the matrix from the Time-Spectral Method has no dependence on the initial conditions while the matrix with BDF time marching does. In the long time solution, this influence should become negligible. In the long time solution, however, the block matrix made from BDF time-marching must become much larger to account for many more time steps. Also the block matrix made from BDF time-marching is block lower triangular. This allows a block form of forward substitution to be used to solve this block matrix. As the Time-Spectral Matrix has a wider bandwidth, a more complicated solving algorithm must be implemented.

$$\begin{bmatrix}
\gamma+2\zeta & -\zeta & 0 & -d_1 & 0 & 0 & 0 & d_1 & 0 & 0 \\
-\gamma-\zeta & \gamma+2\zeta & -\zeta & 0 & -d_1 & 0 & 0 & 0 & d_1 & 0 \\
0 & -2\zeta & 2\zeta & 0 & 0 & -d_1 & 0 & 0 & 0 & d_1 \\
d_1 & 0 & 0 & \gamma+2\zeta & -\zeta & 0 & -d_1 & 0 & 0 & 0 \\
0 & d_1 & 0 & -\gamma-\zeta & \gamma+2\zeta & -\zeta & 0 & -d_1 & 0 & 0 \\
0 & 0 & d_1 & 0 & -2\zeta & 2\zeta & 0 & 0 & -d_1 & 0 \\
-d_1 & 0 & 0 & d_1 & 0 & 0 & \gamma+2\zeta & -\zeta & 0 & 0 \\
0 & -d_1 & 0 & 0 & d_1 & 0 & -\gamma-\zeta & \gamma+2\zeta & -\zeta & 0 \\
0 & 0 & -d_1 & 0 & 0 & d_1 & 0 & -2\zeta & 2\zeta & 0
\end{bmatrix}
\begin{bmatrix}
\phi_1^0 \\
\phi_2^0 \\
\phi_3^0 \\
\phi_1^1 \\
\phi_2^1 \\
\phi_3^1 \\
\phi_1^2 \\
\phi_2^2 \\
\phi_3^2
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
(\gamma+\zeta)A\sin(\omega\Delta t) \\
0 \\
0 \\
(\gamma+\zeta)A\sin(\omega 2\Delta t) \\
0 \\
0
\end{bmatrix}$$

(2.2.13)



$$\begin{bmatrix}
\frac{1}{\Delta t} + \gamma + 2\zeta & -\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\gamma - \zeta & \frac{1}{\Delta t} + \gamma + 2\zeta & -\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -2\zeta & \frac{1}{\Delta t} + 2\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{-4}{2\Delta t} & 0 & 0 & \frac{3}{2\Delta t} + \gamma + 2\zeta & -\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-4}{2\Delta t} & 0 & \frac{-c}{\Delta x} - \zeta & \frac{3}{2\Delta t} + \gamma + 2\zeta & -\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2\zeta & \frac{3}{2\Delta t} + 2\zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2\Delta t} & 0 & 0 & \frac{-4}{2\Delta t} & 0 & 0 & \frac{3}{2\Delta t} + \gamma + 2\zeta & -\zeta & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2\Delta t} & 0 & \frac{-4}{2\Delta t} & 0 & 0 & -\gamma - \zeta & \frac{3}{2\Delta t} + \gamma + 2\zeta & -\gamma & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2\Delta t} & 0 & 0 & 0 & -2\zeta & \frac{3}{2\Delta t} + 2\zeta & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\phi_1^1 \\
\phi_1^2 \\
\phi_1^3 \\
\phi_2^1 \\
\phi_2^2 \\
\phi_2^3 \\
\phi_3^1 \\
\phi_3^2 \\
\phi_3^3
\end{bmatrix}
=
\begin{bmatrix}
\frac{1}{\Delta t} \phi_1^0 + (\gamma + \zeta) A \sin(\omega \Delta t) \\
\frac{1}{\Delta t} \phi_2^0 \\
\frac{1}{\Delta t} \phi_3^0 \\
\frac{-1}{2\Delta t} \phi_1^0 + (\gamma + \zeta) A \sin(\omega 2\Delta t) \\
\frac{-1}{2\Delta t} \phi_2^0 \\
\frac{-1}{2\Delta t} \phi_3^0 \\
(\gamma + \zeta) A \sin(\omega 3\Delta t) \\
0 \\
0
\end{bmatrix}
\tag{2.2.14}$$

### 2.2.2 Linearization of Convection

The convection term in the Navier-Stokes Equations is particularly difficult because it is nonlinear. Care must therefore be taken when solving the Navier-Stokes Equations by utilizing linear systems techniques. For the Time-Spectral Method, the nonlinearity is treated by using Picard Iteration Techniques. That is, an initial guess is used to form the fully time-coupled linear system, the linear system is solved, and the new value is used as the next iterate. This is successively repeated until the resultant value and the previous iterate are sufficiently close to each other.

Conventional time-marching techniques, such as BDF, may use Picard Iterations. More frequently, however, the initial guess for the convection value is based on the previous time-step with no correction step. This can, of course, have accuracy consequences, but may prove satisfactory for some problems, especially when small time steps are involved. These methods by which to linearize the nonlinear term are shown as follows.

$$\text{Time Spectral: } \phi^n \phi^n \approx {}_{\eta-1}\phi^n {}_{\eta}\phi^n \quad (2.2.15)$$

$$\text{BDF no iterations: } \phi^n \phi^n \approx \phi^{n-1} \phi^n \quad (2.2.16)$$

$$\text{BDF with iterations: } \phi^n \phi^n \approx {}_{\eta-1}\phi^n {}_{\eta}\phi^n ; {}_0\phi^n = \phi^{n-1} \quad (2.2.17)$$

Here  ${}_{\eta}\phi^n$  represents  $\phi$  at time level  $n$  and iteration level  $\eta$ .

## 2.3 Block-Jacobi Algorithm

Now that the fully time-coupled matrix is defined, an algorithm needs to be used to solve it. While there are many available linear system solvers, they are not readily available for use with the Time-Spectral Method inside standard CFD frameworks. This is because standard CFD libraries only expect to solve for one time level at

a time. This is because the only contribution from other time levels with standard time marching methods is in the source term. Were standard time marching to be represented in the fully time-coupled matrix form, it would be a block diagonal and block lower triangular matrix. The block diagonal would represent the current time level, while the block lower triangular would represent the contribution from previous time levels. This block lower triangular matrix is solved using forward substitution and a block solve at each time level, hence the lack of a need for time coupling in the linear systems solver. As the Time-Spectral Method has all of the time points coupled, it requires a time-segregated algorithm in order to be used with typical time-segregated CFD solvers. The block-Jacobi method fits this approach, which is why it was adopted by Sicot et al. (2008). The following steps describe the block-Jacobi algorithm (Quarteroni et al., 2000; Saad, 2003):

1. Guess Initial  $\left\{ \phi^n \right\}$  for each  $n$
2. For  $\lambda = 0, \dots, \Lambda - 1$  Nonlinear Updates
  - (a) For  $n = 0, \dots, N - 1$  time levels  
Form Block Matrix  $\left[ \lambda R^n \right]$  and source term  $\left\{ \lambda R_S^n \right\}$  from  $R(\left\{ \lambda \phi^n \right\})$
  - (b) For  $\eta = 0, \dots, H - 1$  Linear Iterations
    - i. For  $n = 0, \dots, N - 1$  time levels  
Solve

$$\left[ \lambda R^n \right] \left\{ \eta \phi^n \right\} = \left\{ \lambda R_S^n \right\} - \sum_{\substack{m=0 \\ m \neq n}}^{N-1} \left[ P_{n-m} \right] \left\{ \eta \phi^m \right\} \quad (2.3.1)$$

$\eta$  and  $\lambda$  are used as iteration counters. Note that the implemented algorithm in fact uses the best available guess of  $\phi$ , and thus is actually a block version of the Gauss-Seidel algorithm. Nevertheless, the phrase “block-Jacobi” will be used following the terminology of Sicot et al. (2008).

The standard Jacobi solver requires diagonal dominance to guarantee stability (Quarteroni et al., 2000, pg. 130). Although no sources could be found that verify this for the block-Jacobi solver, it is assumed that the block-Jacobi solver also requires diagonal dominance. This assumption has held in practice as matrices without a diagonal boost have failed to converge, while diagonally dominant matrices will converge. As a solution to making the system diagonally dominant, the fully time-coupled matrix can be under-relaxed. Diagonal dominance is normally a small problem as the time-derivative term typically adds a large component to the diagonal. This can be seen in Equation 2.1.10 as BDF has a term at the current time level which is proportional to  $1/\Delta t$  so that diagonal dominance increases as the time step gets smaller. The requirement for diagonal dominance is a larger problem with the Time-Spectral Method because it has no contribution to the diagonal term as can be seen in Equation 2.1.9. Given the problems cited by Jameson (2009) and Antheaume and Corre (2011) (and discussed in Chapter I) for incompressible flow, solving this system with the same methods used for compressible flow is insufficient. In standard time-marching methods, it is frequently necessary to use under-relaxation to stabilize the solution of incompressible flow. This condition is despite also using implicit time-marching schemes. Following this idea, under-relaxation is applied to the incompressible Time-Spectral Method. Implicit under-relaxation is applied to matrices, while explicit under-relaxation is applied to the pressure coupling variable.

The main idea of implicit under-relaxation is to boost diagonal dominance by increasing the diagonal of the matrix by a multiplicand, and then adding a source term to compensate for the alteration of the diagonal. After a sufficient number of iterations, the solution to the original equations is then recovered. For a general linear system of equations,  $A\phi = b$ ,  $A$  is decomposed into an upper and lower triangular matrix,  $F$  and  $E$  respectively, and a diagonal  $D$ . The basic formula is as follows:

$$A\phi = (F + D + E)\phi = b \quad (2.3.2)$$

Using  $\eta$  as an iteration counter, and  $\alpha$  as the under-relaxation factor, the under-relaxed system is as follows:

$$\frac{D}{\alpha} \eta \phi + (U + L)_{\eta} \phi = b + \frac{1 - \alpha}{\alpha} D_{\eta-1} \phi \quad (2.3.3)$$

It can be shown that if  ${}_{\eta} \phi \approx {}_{\eta-1} \phi$ , then the solution of Equation 2.3.3 is the solution to Equation 2.3.2. It should be noted that this results in the same equations when applied to time-coupled or time-segregated matrix as it is only applied to the diagonal elements and not the diagonal block.

When using under-relaxation to stabilize the system of equations, it is useful to know how much under-relaxation is needed. The requirement for diagonal dominance is that the pole term of each line in a matrix is larger in magnitude than, or equal in magnitude to, the sum of the magnitudes of all the other terms acting on that variable. That is:

$$|a_{ll}| \geq \sum_{m \neq l} |a_{lm}| \quad (2.3.4)$$

where  $a_{lm}$  is the coefficient of matrix  $A$  at the  $l$  row and  $m$  column. It should be noted that at least one line needs to have strict inequality. This inequality usually happens due to boundary conditions.

The Linear Advection-Diffusion Equation is the model PDE to analyze the amount of under-relaxation needed to stabilize the block-Jacobi method when using the Time-Spectral Method. The one-dimensional version of the Linear Advection-Diffusion Equation given in Equation 2.2.3 will be used to analyze diagonal dominance. Dis-

cretizing the one-dimensional equation yields the following three terms:

$$\text{Time-Spectral: } \frac{\partial \phi}{\partial t} = d_1 \phi_i^{n+1} + d_{-1} \phi_i^{n-1} \quad (2.3.5)$$

$$\text{Upwind: } c \frac{\partial \phi}{\partial x} = \frac{c}{\Delta x} (\phi_i^n - \phi_{i-1}^n) \quad (2.3.6)$$

$$\text{Linear: } \nu \frac{\partial^2 \phi}{\partial x^2} = \frac{\nu}{\Delta x^2} (\phi_{i-1}^n - 2\phi_i^n + \phi_{i+1}^n) \quad (2.3.7)$$

Of course other discretizations could be used, but these are used as representative discretizations. Also realize that the upwind discretization for advection assumes  $c > 0$ . The discretized equation appears as:

$$d_1 \phi_i^{n+1} + d_{-1} \phi_i^{n-1} + \frac{c}{\Delta x} (\phi_i^n - \phi_{i-1}^n) - \frac{\nu}{\Delta x^2} (\phi_{i-1}^n - 2\phi_i^n + \phi_{i+1}^n) = 0 \quad (2.3.8)$$

Note that in the fully time-coupled matrix, only the coefficients at the current space and time point contribute to the diagonal term. This formulation can now be broken up into the diagonal and off diagonal components as follows:

$$\left( \frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2} \right) \phi_i^n + d_1 \phi_i^{n+1} + d_{-1} \phi_i^{n-1} - \left( \frac{c}{\Delta x} + \frac{\nu}{\Delta x^2} \right) \phi_{i-1}^n - \frac{\nu}{\Delta x^2} \phi_{i+1}^n = 0 \quad (2.3.9)$$

Since the magnitude of each component is desired, Equation 2.3.9 can be viewed in light of Equation 2.3.4 as follows:

$$a_{ll} \phi_l + \sum_{m \neq l} a_{lm} \phi_m = 0 \quad (2.3.10)$$

$$|a_{ll}| = \left| \frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2} \right| = \frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2} \quad (2.3.11)$$

$$\sum_{m \neq l} |a_{lm}| = 2|d_1| + \left| \frac{c}{\Delta x} + \frac{\nu}{\Delta x^2} \right| + \left| \frac{\nu}{\Delta x^2} \right| = 2|d_1| + \frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2} \quad (2.3.12)$$

This relationship takes advantage of the relationship  $d_1 = -d_{-1}$  and that in the upwind formulation  $c > 0$ . These equations show that  $|a_{ll}| < \sum_{m \neq l} |a_{lm}|$  because of

the contribution from the Time-Spectral terms. By using under-relaxation, diagonal dominance can be recovered.

$$\frac{|a_{ll}|}{\alpha} \geq \sum_{m \neq l} |a_{lm}| \quad (2.3.13)$$

$$\alpha \leq \frac{|a_{ll}|}{\sum_{m \neq l} |a_{lm}|} \quad (2.3.14)$$

$$\alpha \leq \frac{\frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2}}{2|d_1| + \frac{c}{\Delta x} + \frac{2\nu}{\Delta x^2}} \quad (2.3.15)$$

$$\alpha \leq \frac{\frac{c\Delta t}{\Delta x} + \frac{2\nu\Delta t}{\Delta x^2}}{2\Delta t|d_1| + \frac{c\Delta t}{\Delta x} + \frac{2\nu\Delta t}{\Delta x^2}} \quad (2.3.16)$$

$$\alpha \leq \frac{\sigma + 2\beta}{2\Delta t|d_1| + \sigma + 2\beta} : \sigma = \frac{c\Delta t}{\Delta x} : \beta = \frac{\nu\Delta t}{\Delta x^2} \quad (2.3.17)$$

$$\alpha \leq \frac{(\sigma + 2\beta)/\delta}{1 + (\sigma + 2\beta)/\delta} : \delta = 2\Delta t|d_1| \quad (2.3.18)$$

$$\alpha \leq \frac{v}{1 + v} : v = \frac{\sigma + 2\beta}{\delta} = \frac{\sigma + 2\beta}{2\Delta t|d_1|} \quad (2.3.19)$$

Equation 2.3.19 shows a relationship between a dimensionless parameter  $v$  and the under-relaxation factor. Larger  $v$  places less restriction on  $\alpha$  to stabilize the solution. As such, larger Courant Number ( $\sigma$ ) and larger  $\beta$  actually help to stabilize the solution. This is in contrast to conventional time-marching methods where there is typically more stability as  $\sigma$  and  $\beta$  are reduced. As  $v$  is inversely correlated to cell size, the requirements on under-relaxation factor are actually reduced as the grid is refined. This is again different than the stability requirements of many polynomial based time-marching schemes where the stability requirements increase as grid size decreases due to Courant number restrictions. When moving into higher dimensional flow,  $v$  will have additional terms to account for spatial coupling in other directions. As these terms are similar to the one-dimensional spatial terms, they will contribute to the numerator of  $v$  and thus improve diagonal dominance of the fully time-coupled matrix. When looking at improved temporal resolution, however, only the off-diagonal terms are increased. For the same spatial discretization schemes, improving the temporal

scheme to five time levels would use:

$$\frac{\partial \phi}{\partial t} = d_2 \phi_i^{n+2} + d_1 \phi_i^{n+1} + d_{-1} \phi_i^{n-1} + d_{-2} \phi_i^{n-2} \quad (2.3.20)$$

Resulting in:

$$\alpha \leq \frac{\sigma + 2\beta}{2\Delta t(|d_1| + |d_2|) + \sigma + 2\beta} \quad (2.3.21)$$

This has the same effect as increasing the magnitude of  $\delta$  in  $v$ . This example shows how, as more time-levels are added, the magnitude of the off-diagonal terms increases, which necessitates a smaller  $\alpha$  for diagonal dominance.

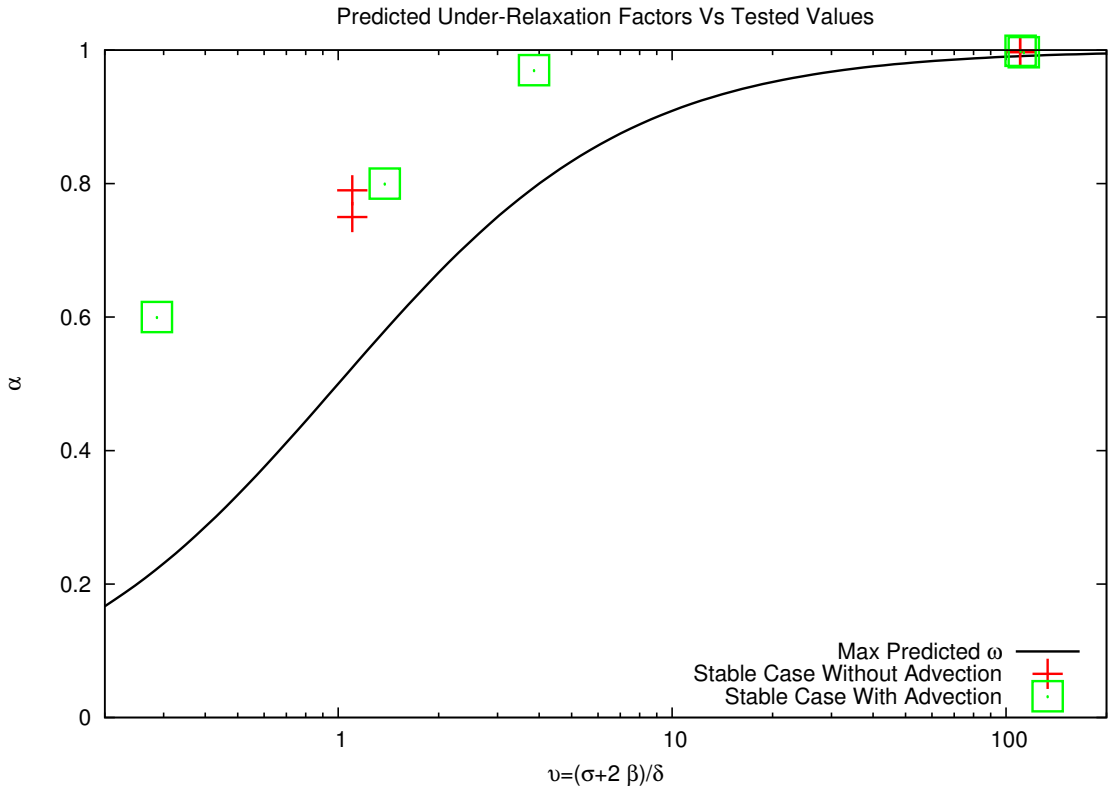


Figure 2.2: Predicted versus Tested Under-Relaxation Values for Three Time-Levels

In order to test the derivation of maximum  $\alpha$  given in Equation 2.3.19, numerical experiments were performed to find stable under-relaxation factors that were



close to the stability limit. Figure 2.2 displays the the predicted maximum under-relaxation versus the maximum tested under-relaxation values for linear-advection diffusion. What is clear is that this predicted value significantly under-predicts the required under-relaxation value, and thereby unnecessarily limits convergence. Figure 2.2 shows that tests both with and without advection have this property. This extra stabilizing effect could be due to boundary conditions, numerical errors, or the block solver.

Explicit under-relaxation is a similar iterative technique, but it only uses a fraction of the solution to update the previous value. Explicit under-relaxation does not help solve the matrix itself, but it can help improve stability in coupled equations. For instance, explicit under-relaxation is often used to limit the pressure update in an iterative incompressible Navier-Stokes algorithm. Explicit under-relaxation helps in these algorithms because the full pressure update will frequently introduce numerical oscillations that will destabilize the solution. The formulation for explicit under-relaxation is as follows:

$$\phi^{new} = \phi^{old} + \alpha(\phi^{computed} - \phi^{old}) \quad (2.3.22)$$

Both explicit and implicit under-relaxation help to stabilize the solution algorithm and thus are used in this thesis. Under-relaxation techniques are popular with segregated incompressible flow algorithms that use pressure projection due to the already iterative nature of these algorithms.

For compressible flow, however, dual-time stepping is usually used in lieu of under-relaxation. Aside from this thesis, the Time-Spectral Method and its predecessors are formulated either for compressible flow or to use compressible flow techniques. As such, instead of under-relaxation, the previous methods frequently used a dual-time stepping approach to boost diagonal dominance and stabilize the matrix (Sicot et al.,

2008; Gopinath and Jameson, 2005; Sicot et al., 2012; Butsumtorn and Jameson, 2008; Su and Yuan, 2010). This is also called pseudo time marching (Hall et al., 2002; McMullen et al., 2002; Nadarajah et al., 2003; Ekici and Hall, 2007; Kumar and Murthy, 2007). The dual-time stepping approach adds a pseudo time term which is marched to steady state such that it no longer affects the solution. This is described by the following PDE:

$$\frac{\partial \phi}{\partial \tau} + \frac{\partial \phi}{\partial t} + R(\phi) = 0 \quad (2.3.23)$$

using  $R(\phi)$  to represent all the spatial terms. For the Time-Spectral matrix discretized as  $A$ , this becomes:

$$\frac{\partial \phi}{\partial \tau} + A\phi = b \quad (2.3.24)$$

Using  ${}_{\eta}\phi$  to represent  $\phi$  at the  $\eta$ th pseudo time level, and a backward Euler pseudo time discretization, Equation 2.3.24 becomes:

$$\frac{{}_{\eta}\phi - {}_{\eta-1}\phi}{\Delta\tau} + A_{\eta}\phi = b \quad (2.3.25)$$

Equation 2.3.25 makes it clear that once  $\phi$  is at pseudo time steady state, the pseudo time derivative makes no contribution since  ${}_{\eta}\phi = {}_{\eta-1}\phi$  at pseudo time steady state. As the pseudo time term makes a contribution to the diagonal, this system can be represented as:

$$\left( \frac{1}{\Delta\tau} + D \right) {}_{\eta}\phi + (F + E)_{\eta}\phi = b + \frac{1}{\Delta\tau} {}_{\eta-1}\phi \quad (2.3.26)$$

Equation 2.3.26 shows a relationship between dual time stepping and under-relaxation as both make contributions to the diagonal. Under-relaxation makes a multiplicative contribution, whereas dual time stepping makes an additive contribution.

Comparing this relationship directly shows that

$$\alpha = \frac{D\Delta\tau}{1 + D\Delta\tau} \quad (2.3.27)$$

This shows that the relationship between the under-relaxation factor and the pseudo time step size is dependent on the diagonal terms, and thus is non-uniform over the matrix.

For dual-time stepping, Hall et al. (2002) talks about instability that eventually lead to nonconvergence for more fine time discretizations. Von Neumann stability analysis shows that dual-time stepping is unstable for the time-spectral method. This is shown in Appendix B on a system without either pseudo time stabilization or under-relaxation. Hall et al. (2002) points out that “unlike most CFD instabilities, which tend to involve short wavelength disturbances, the instability here is associated with the longest wavelengths.” Like Hall et al. (2002), this work has found that the “boundaries provide a stabilizing influence”. For this reason, Von Neumann stability analysis is not especially relevant to Time-Spectral simulations with coarse time discretization. For the more fine time discretizations, however, Von Neumann analysis does help explain how some low frequency perturbations receive insufficient smoothing.

The Time-Spectral Method is derived from a continuous function. This method is shown to create a system of time-coupled equations that are represented by a fully time-coupled matrix. The block-Jacobi algorithm is introduced in order to solve this fully time-coupled matrix in a time-segregated manner. Under-relaxation is introduced as a way to increase diagonal dominance for the block-Jacobi solver and an estimate for maximum allowable under-relaxation factor is presented. This under-relaxation is compared to dual-time stepping, which is the method presented in literature for increasing diagonal dominance of the fully time-coupled matrix. Overall, these practices lay the groundwork for using the Time-Spectral Method to solve PDEs presented in the following chapters.

## CHAPTER III

# Scalar Transport and Burgers' Equation

### 3.1 Linear Advection-Diffusion

The Linear Advection-Diffusion Equation represents scalar transport through a field. It is also called the Scalar Transport Equation for this reason. The Linear Advection-Diffusion Equation could represent transport of temperature gradients, a fluid-species, or other transport. The Linear Advection-Diffusion Equation bears many similarities to the full Navier-Stokes Equations. Because of this, the Linear Advection-Diffusion Equation is frequently used as a model equation in testing codes that solve the full Navier-Stokes equations. The advantage of the Scalar Transport Equation is that it is linear and uncoupled. The Scalar Transport Equation also is simplified in that it does not contain the pressure coupling term of the Navier-Stokes Equations. The linearity of the Linear Advection-Diffusion Equation also makes it much easier to analyze than the Navier-Stokes Equations for properties such as error analysis or stability analysis. Exact solutions exist for the Linear Advection-Diffusion Equation. This gives test problems available to test code or discretization schemes. Discretized problems can therefore be solved and analyzed by linear algebra techniques.

The Linear Advection-Diffusion Equation is:

$$\frac{\partial \phi}{\partial t} + \vec{u} \cdot \nabla \phi - \nu \nabla^2 \phi = 0 \quad (3.1.1)$$

This equation represents simple scalar ( $\phi$ ) transport in a fluid where  $\vec{u}$  is the fluid field and  $\nu$  is the diffusion coefficient of the scalar variable. The one-dimensional representation of this PDE is:

$$\frac{\partial \phi}{\partial t} + c \frac{\partial \phi}{\partial x} - \nu \frac{\partial^2 \phi}{\partial x^2} = 0 \quad (3.1.2)$$

where  $c$  is the propagation velocity.

This one-dimensional equation was used to test the initial Time-Spectral algorithm in MATLAB<sup>®</sup>. What the MATLAB<sup>®</sup> solver did show was that for the Time-Spectral Method in 1D, a direct Gaussian solve was substantially faster than block-Jacobi iterations. This was expected as Khoo et al. (2003) shows that iterative solvers tend to perform more poorly than a direct solve until higher dimensions are used. This illustrates that the block-Jacobi is not necessarily the best available method to solve the fully-time coupled equations, but rather one method available when the equations must be cut by time level. The rest of the results presented in this thesis, however, were made using a solver based on the Open Field Operation And Manipulation (OpenFOAM<sup>®</sup>) CFD library, which was limited to the block-Jacobi linear systems solver for the Time-Spectral Method. OpenFOAM<sup>®</sup> was used because it is an industrial CFD library capable of handling very complex three dimensional geometry using arbitrary polyhedral discretization (Weller et al., 1998; Jasak et al., 1999; Jasak, 2009). OpenFOAM<sup>®</sup>, however, only expects to solve a linear system of equations at a single time-level like most CFD libraries. For this and other reasons, OpenFOAM<sup>®</sup> is a good surrogate for other industrial CFD codes where only one time-level can be solved at a time.

### 3.1.1 Linear Advection

The one-dimensional linear advection-diffusion equation has exact solutions available. For pure sine wave advection with  $c > 0$ ,  $\nu = 0$ , and  $\phi(x, 0) = A \sin(x)$ , the solution is  $\phi(x, t) = A \sin(x - ct)$  which means the inflow boundary condition is  $\phi(0, t) = A \sin(-ct)$ . This case was used to verify the time-spectral method and to get an idea of what under-relaxation factors are reasonable. It was found that time-spectral methods could be 100 times faster with a properly chosen under-relaxation factor. For this problem, the results are dependent on the under-relaxation factor. The 100 fold speedup happened around  $\alpha = 0.98$  with  $\alpha = 1.0$  being unstable and  $\alpha = 0.9$  being only about as fast as BDF time marching.

Figure 3.1 shows results for a similar problem. Here the boundary conditions are  $\phi(0, t) = \sin 2\pi t + 0.2 \sin 14\pi t$ , which corresponds to linear combination of a long wave and a short wave. The error presented in Figure 3.1 is based on the difference from the exact solution. The  $L_2$  error is defined as:

$$L_2 = \sqrt{\frac{1}{n} \sum_i (\phi_i - \phi_{i,exact})^2} \quad (3.1.3)$$

where  $\phi_i$  is the value of  $\phi$  at  $x = i\Delta x$ . Even for a perfectly time-resolved solution, there is some error that is present due to the spatial discretization scheme (second order upwind biased), and spatial discretization (100 points); both of which were kept constant for these comparisons. As expected, the Time-Spectral Method will only resolve the longer wave when the number of time levels is greater than required sampling frequency of the shorter wave. This can be seen in the  $L_2$  error being constant for Time-Spectral Method until the algorithm can resolve the shorter wave, at which point the  $L_2$  error is again constant. The under-relaxation factor used in these computations was: 0.98 for 3, 5, and 9 time levels, 0.97 for 17 time levels, and 0.85 for 33 time levels. The BDF solutions, however, are much more dissipative, which

is why it takes an order of magnitude more time-levels in order to obtain the same  $L_2$  error. This comparison, therefore, shows an improved accuracy of the Time-Spectral Method over BDF.

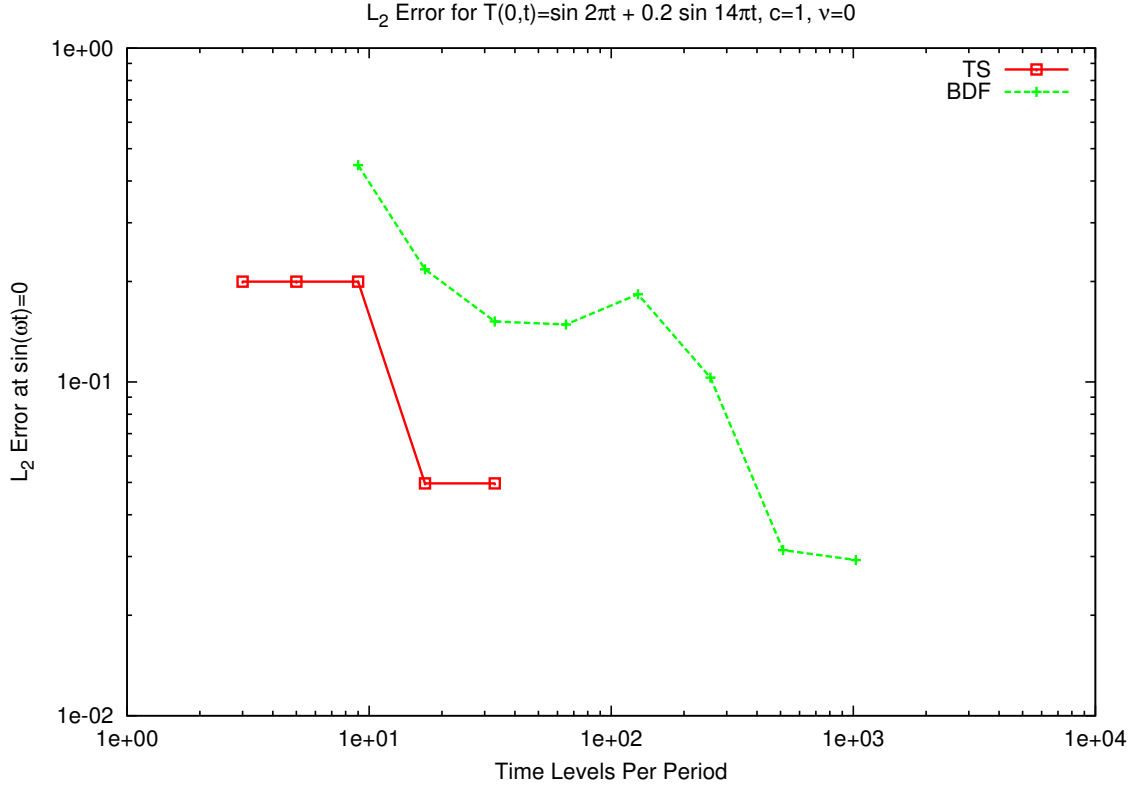


Figure 3.1: Spectral Accuracy versus Second Order Accuracy for Linear Advection

### 3.1.2 Linear Diffusion

Another exact solution of the one-dimensional Linear Advection-Diffusion Equation is the solution of fluid above an infinite oscillating plate. This problem is often called Stokes' Second Problem (White, 2005) as it first appeared in Stokes (1851). This problem is derived from simplifications of the Navier-Stokes Equations, but results in a one-dimensional diffusion problem since it assumes an infinite domain so  $\partial/\partial x = \partial/\partial z = 0$  and the problem simplifies to  $u = u(y, t)$ .  $u$  is the x component of

velocity, but this equation could also easily apply to a temperature diffusion with a sinusoidally oscillating temperature at the wall and equilibrium at the far field. The flow is driven by a wall at  $y = 0$  which oscillates at  $u(0, t) = U_0 \cos(\omega t)$ . In the far field there is no flow which means  $u(\infty, t) = 0$ . Because of the infinite domain assumption, there is no mean cross flow resulting in  $c = 0$ . The resulting equation is:

$$\frac{\partial u}{\partial t} - \nu \frac{\partial^2 u}{\partial y^2} = 0 \quad (3.1.4)$$

The solution of this pure diffusion problem is:

$$u = U_0 e^{-\eta} \cos(\omega t - \eta) \quad (3.1.5a)$$

$$\eta = y \sqrt{\frac{\omega}{2\nu}} \quad (3.1.5b)$$



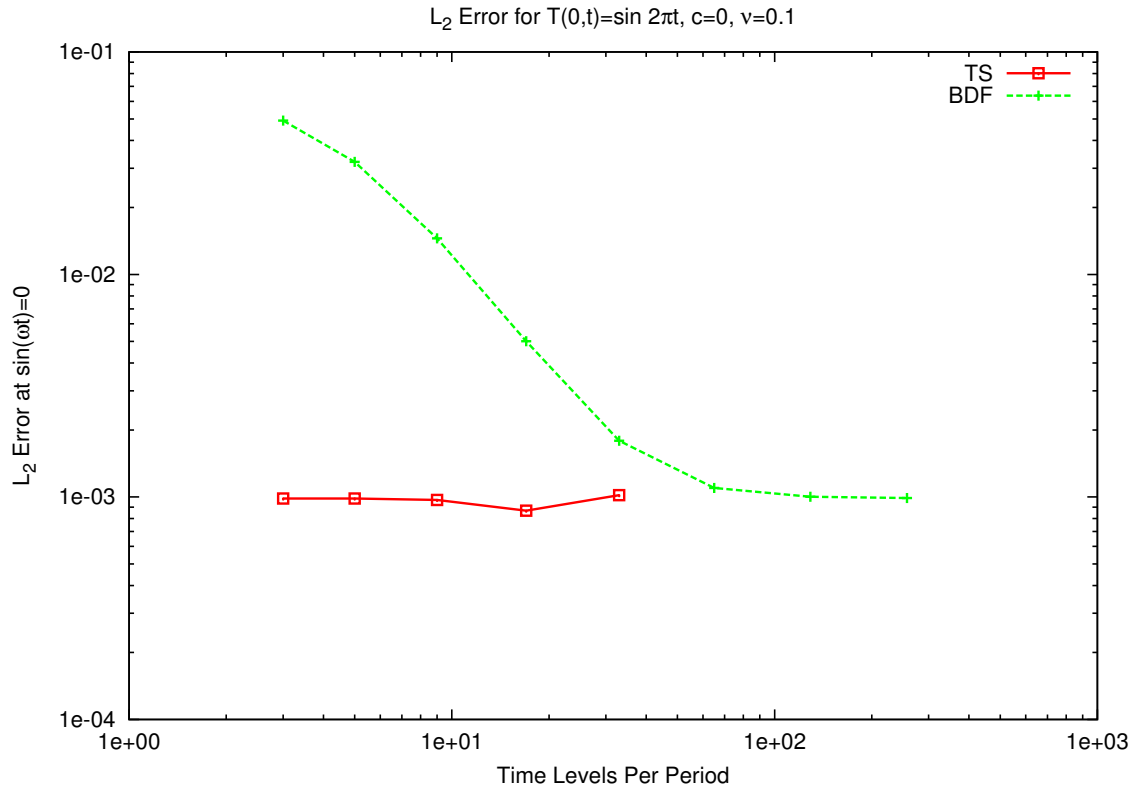


Figure 3.2: Spectral Accuracy versus Second Order Accuracy for Linear Diffusion

As linear advection tested the advection part of the Linear Advection-Diffusion Equation, linear diffusion tested the diffusion part of the problem. Figure 3.2 shows results for linear diffusion over a broad range of time discretizations. What is again very apparent from this graph is the advantage of spectral accuracy over mere 2nd order accuracy. The Time-Spectral Method achieves high convergence with only 3 time-levels while the Backward Difference Formula does not hit a similar convergence until 129 time-levels. It should be noted that the  $L_2$  error is based on the exact solution, so neither method can achieve an  $L_2$  error of zero due to spatial discretization error.

### 3.1.3 Backward-Facing Step Test Case

As the Linear Advection-Diffusion solver was implemented in 2D and 3D, it makes a simple testing ground for more complicated problems. The next case is to test the Time-Spectral Method with a spatially varying velocity field so advection is not constant. The backward-facing step test case provides this non-constant advection field. It also has the additional benefits of being a standard test case and simple geometry. The advection field is specified so as to closely match experimental results (Pitz and Daily, 1983). This advection field is shown in Figure 3.4. The inflow is set to  $\phi = 1 + 0.3 \cos(\omega t)$ . The outflow is set to  $\partial\phi/\partial n = 0$ . The diffusion coefficient of  $\phi$  is  $\nu = 10^{-4}$ . Figure 3.3 is a picture of the domain.

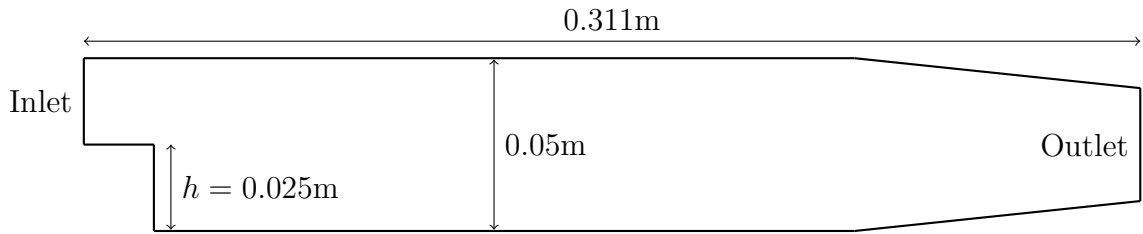


Figure 3.3: Backward-Facing Step Geometry for Linear Advection-Diffusion

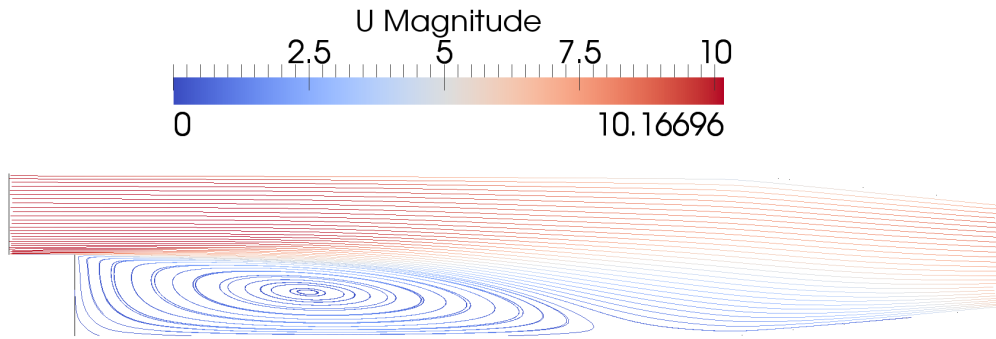


Figure 3.4: Streamlines for Advection Field of Backward-Facing Step Test Case

Figure 3.5 shows the values of  $\phi$  along a horizontal line 20% of the distance between

the step and the top wall. The time marching method is shown with 1000 time steps per cycle, whereas the time-spectral method is shown with only 5 time steps per cycle. The results show good agreement. The Euler implicit method has values that tend slightly more towards the average  $\phi$  value, showing how its more dissipative than the Time-Spectral Method. The Time-Spectral results are very crisp and less diffusive due to spectral accuracy being much higher fidelity than the first-order accuracy of the Euler implicit method.

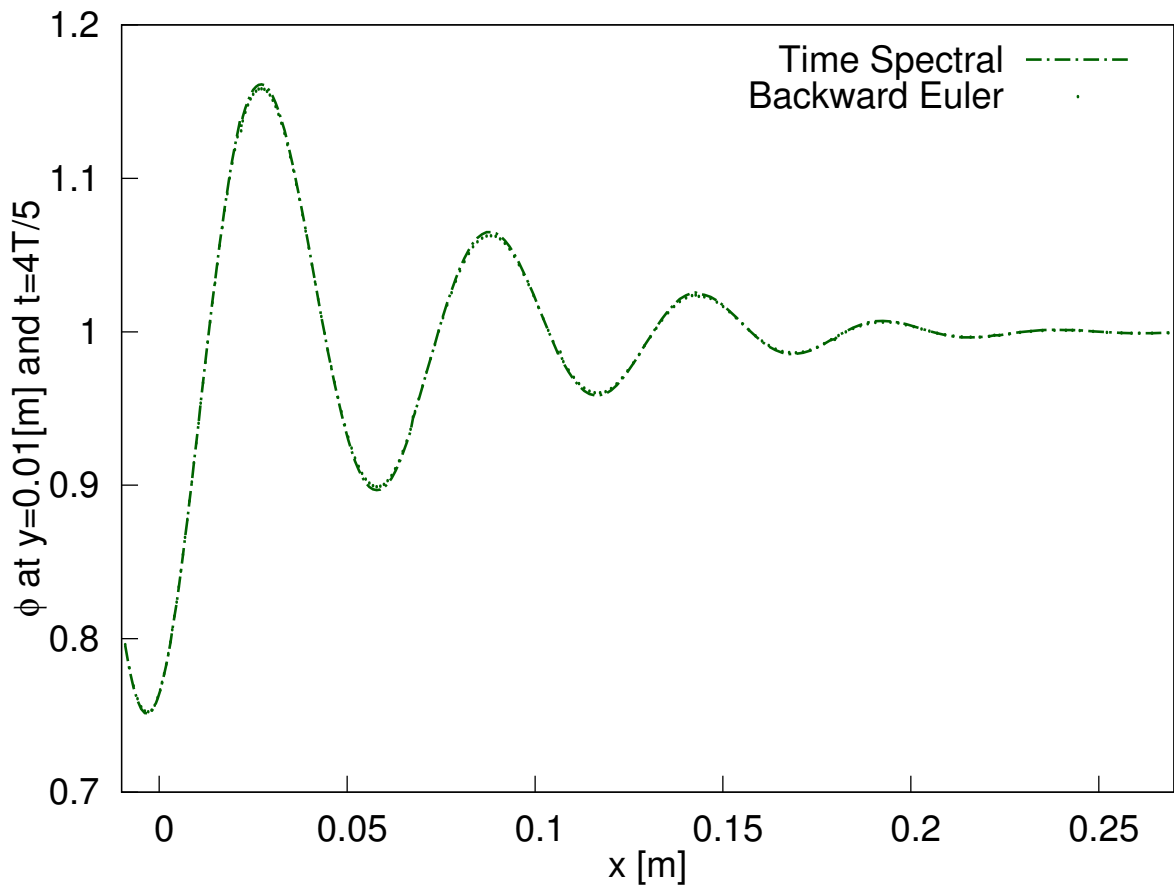


Figure 3.5: Linear Advection-Diffusion Results over Backward-Facing Step for Horizontal Line

### 3.2 Burgers' Equation

The viscous Burgers' Equation is a common nonlinear model equation for CFD techniques. The equation is:

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = 0 \quad (3.2.1)$$

When used in higher dimensions, the following equation is used:

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot \left( \frac{1}{2} \vec{u} \vec{u} \right) - \nu \nabla^2 \vec{u} = 0 \quad (3.2.2)$$

where  $\vec{u} \vec{u}$  is the vector outer product.

Being a nonlinear equation, Burgers' Equation allows examination of how much nonlinearity affects the solution, particularly, how the nonlinearity destabilizes the solution. This examination is essential because the primary difference between Burgers' Equation and the Navier-Stokes Equations is the pressure term. Furthermore, Burgers' Equation allows isolation of the nonlinear term as the nonlinear term is the only change from the Linear Advection-Diffusion Equation that was also tested. As mentioned in Section 2.2.2, the linearization of the nonlinear term can be done in several ways. Standard time-marching methods typically guess the value of  $\vec{u}$  from the previous time level. The Time-Spectral Method guesses the value of  $\vec{u}$  from the current time level, but the previous iteration. That is,  $\nabla \cdot (\vec{u} \vec{u})_{TS} \approx \nabla \cdot (\bar{\eta}_{-1} \vec{u}^n \bar{\eta}^n)$  while  $\nabla \cdot (\vec{u} \vec{u})_{BDF} \approx \nabla \cdot (\bar{u}^{n-1} \vec{u}^n)$ . This is done because the change in  $\vec{u}$  from time-level to time-level is typically greater with time-spectral methods than standard time-marching methods because of the larger time step size.

It should be noted that the method of linearization used here differs from the method in Section 2.2.2. The method in Section 2.2.2 prescribes solving the fully time-coupled matrix, then updating for nonlinearities and resolving. It was found that

it is faster to update for nonlinearities during each block-Jacobi sweep. By updating during every block-Jacobi sweep, the nonlinearities are more tightly coupled with the solution. Updating nonlinearities during each block-Jacobi sweep has the additional benefit that only one matrix block needs to be in memory at a time, thus keeping memory costs down.

### 3.2.1 One-Dimensional Oscillating Inflow

The first test case was a one-dimensional version of the oscillating inflow problem. The inlet was set to  $u = 1 + 0.1 \cos(\omega t)$  and the outlet was set to  $\partial u / \partial x = 0$  with unity initial conditions. This grid contained 100 spatial points over a distance of 1m. Figure 3.6 shows the solution as a function of time at a point that is halfway through the domain. Figure 3.6 compares the Time-Spectral Method with 5 time-steps (called TS5) to a very highly resolved (500 time levels per period) second-order Backward Difference Formula (BDF). The time-spectral method compares favorably to the BDF results. This indicates that the Time-Spectral Method is useful for nonlinear problems, as is expected from prior results in literature (see Chapter I). It also indicates that using under-relaxation instead of dual time stepping to stabilize the fully time-coupled matrix is indeed an effective tool for nonlinear problems as well. A preliminary timing analysis showed that for this case the choice of under-relaxation factor made a large difference in how quickly the Time-Spectral Method converged. It had a factor of five faster convergence with an under-relaxation factor change of 0.9 to 0.98. The reason for this large change of convergence is that an ideal under-relaxation factor is 1.0, and the deviation from 1.0 is what affects convergence speed.

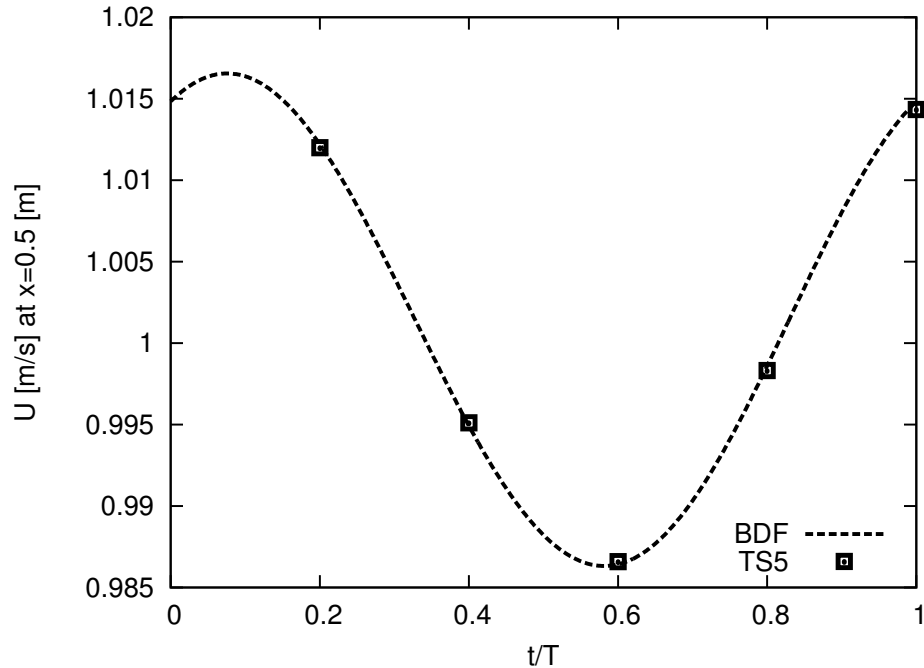


Figure 3.6: One Dimensional Burgers' Equation Simulation

This one dimensional case was retested using a varying number of iterations over the fully time-coupled matrix to understand how better linear updates could improve the speed to convergence. The results of this analysis are shown in Figure 3.7. Here work is the number of solves over the fully time-coupled matrix and the dependent variable is the velocity halfway through the domain. Each iteration is a sweep over the fully time-coupled matrix without updating for nonlinearities. Then nonlinearities are updated and the fully time-coupled matrix is iterated over until periodic steady state convergence. For example, with 2 iterations, the fully time-coupled matrix was formed, then every time level was solved once, then every time level was solved again, with the only change being the coupling due to time-discretization. After these 2 iterations, the fully time-coupled matrix was updated for nonlinearities. Then this process was repeated until convergence. The results from Figure 3.7 show that updating for nonlinearities at each iteration produces the fastest convergence to periodic steady state. Using a second linear iteration is about 20% slower, with any

more iterations slowing down the computation nearly proportionately to the iterations above the second. Other benefits of updating for nonlinearities at each iteration are simplified implementation and reduced memory usage.

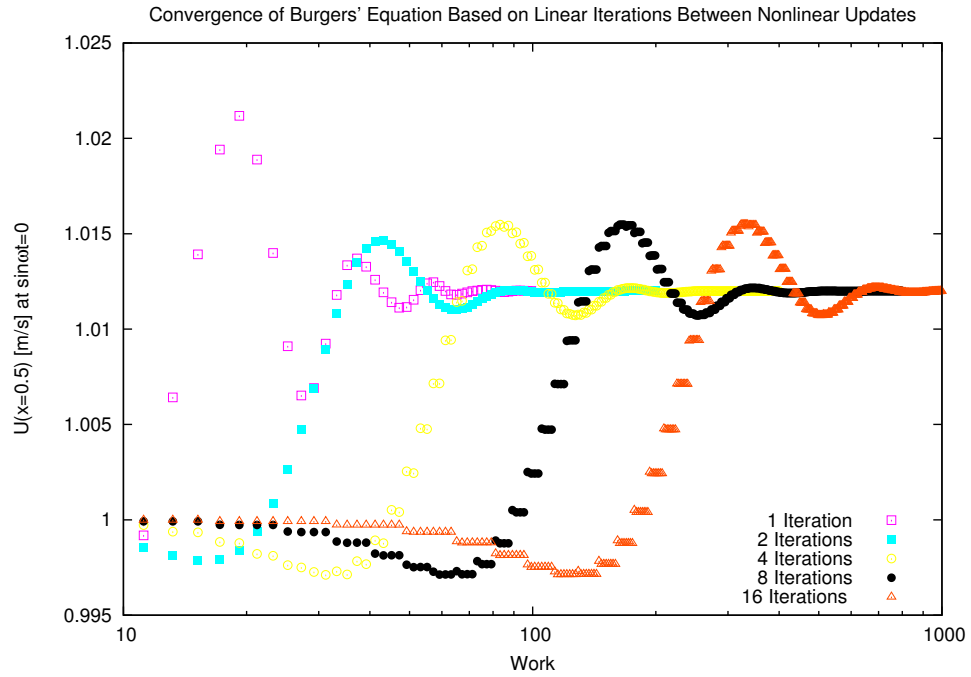


Figure 3.7: Convergence of the Fully Time-Coupled Matrix Due to Iterations Between Nonlinear Updates

### 3.2.2 Backward-Facing Step with Burgers' Equation

Moving on from the one-dimensional case, the next problem is to simulate two-dimensions. This was done for the backward-facing step, and the geometry is shown in Figure 3.8. This test case had 12,000 cells over the domain. The boundary conditions and other parameters of this test case are given in Table 3.1.

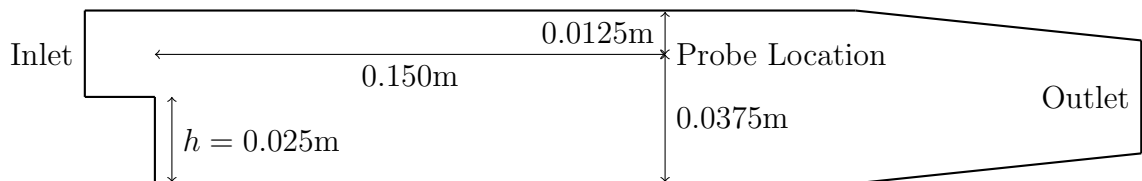


Figure 3.8: Backward-Facing Step Geometry for Burgers' Equation

Table 3.1: Parameters for the Two-Dimensional Burgers' Equation Test Case

$$\text{Inlet : } \vec{u} = U_0(1 + \sigma \sin(\omega t))\hat{i}$$

$$\text{Outlet : } \partial\vec{u}/\partial\hat{n} = 0$$

$$\text{Walls : } \vec{u} = 0$$

$$Re = U_0 h / \nu = 250$$

$$k = \omega h / U_0 = 2.62$$

$$\sigma = 0.1$$

Results are shown in Figure 3.9 for a line with constant y-value, being located midway along the inlet region. Here the first order Euler implicit method is used instead of the second order BDF because the second order BDF was unstable with large time steps. Only five time levels per period are used for each method. Here TS5 designates the Time-Spectral Method with five time levels per period and Euler5 designates the Euler implicit method with five time levels per period. Also, for stability with coarse time discretizations, the upwind spatial discretization scheme was used. This is a first order scheme. Both the Euler implicit method and the Time-Spectral Method produced graphically identical results at small  $\Delta t$ . What is clear from Figure 3.9 is that standard time marching schemes are much too dispersive for a coarse time discretization. The peaks of the waves from the inlet are slowed down so much that the sixth peak from the Euler method is closely matching the fifth peak of the highly resolved solution. The Time-Spectral Method, on the other hand, very closely resembles the highly resolved solution even at such a coarse discretization.



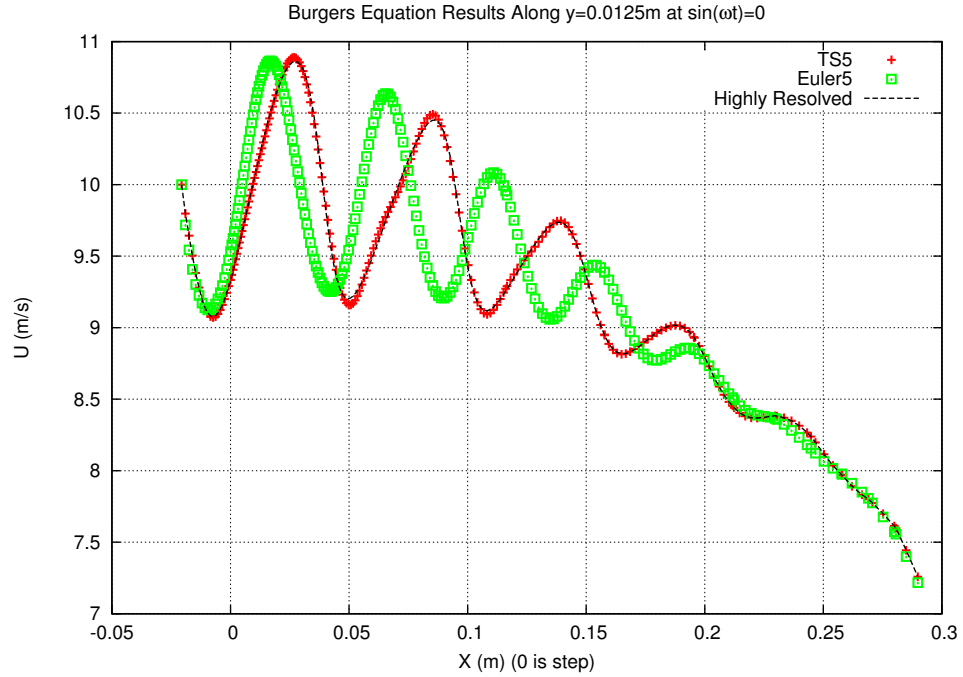


Figure 3.9: Spectral Accuracy versus First Order Accuracy over Backward-Facing Step with Burgers' Equation

For realistic engineering problems, however, first order accuracy is usually inadequate. Therefore, this backward-facing step was redone with a second order upwind biased scheme for spatial discretization and second order BDF for time discretization (for comparison to Time-Spectral Method). Figure 3.10 shows the results of these computations along the same constant  $y$ -value. At the largest stable time steps using the BDF scheme, the results indicate severe oscillations and over predictions of  $\vec{u}$  near the peaks. This behavior may be expected as the second order upwind biased scheme that was used is not a limited scheme, and thus will admit these errors. These oscillations, however, are much less pronounced with finer temporal resolution, so that the results are converging to the same values as the Time-Spectral Method. The Time-Spectral simulation with a much larger time step size, however, produces results very much in line with the BDF's results having 100x as many time steps.

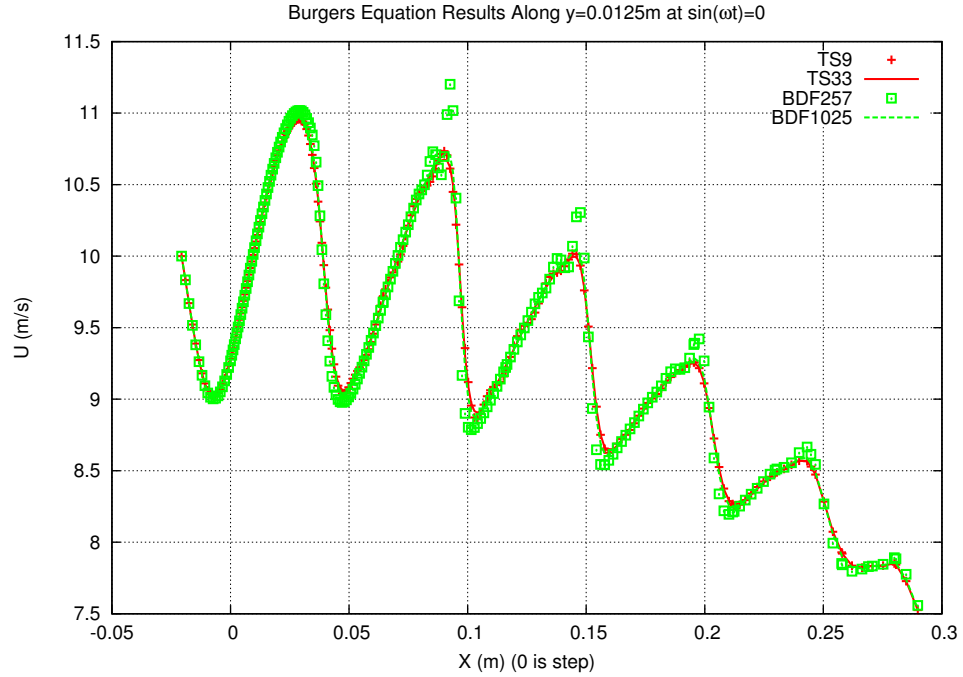


Figure 3.10: Spectral Accuracy versus Second Order Accuracy over Backward-Facing Step with Burgers' Equation

For the same problem, solutions were computed over a range of time-discretizations for both Time-Spectral Method and BDF. In order to evaluate these succinctly, only the probe location referenced in Figure 3.8 will be used for evaluation. Firstly, each discretization was run until the simulation was well converged to a periodic steady state. Next each discretization was evaluated to determine where it was within 0.05% of its periodic steady state value at the time value of  $\sin \omega t = 0$ . The amount of work to get to this 0.05% error is plotted in Figure 3.11. As each discretization is being compared against itself, this figure shows how cost increases as discretization changes. In Figure 3.11 work is  $N * P_{0.05\%}$  where  $N$  is the number of time levels per period, and  $P_{0.05\%}$  is the number of periods to converge to within 0.05% of the fully converged solution. For the Time-Spectral simulations, periods are iterations over the fully time-coupled matrix as OpenFOAM<sup>®</sup>'s time-marching mechanism was used to iterate over time levels. These results indicate that for Time-Spectral simulations,

the cost increases dramatically as more time levels are added. This is partly due to having to solve the much larger fully time-coupled matrix. Another significant slowdown for Time-Spectral Method is that as more time levels are added, there is decreased diagonal dominance, so a smaller under-relaxation factor must be used for stabilizing the fully time-coupled matrix. For reference, the under-relaxation factors used were 0.9, 0.85, 0.7, 0.6, and 0.4 for 3, 5, 9, 17, and 33 time-levels per period respectively. For BDF simulations, the cost for the cheapest simulation is very high, but there is a slower increase in cost as more time-levels are used. This indicates that at these finer discretizations nonlinearities are better taken care of by using the estimate of  $\vec{u}$  at the prior time level.

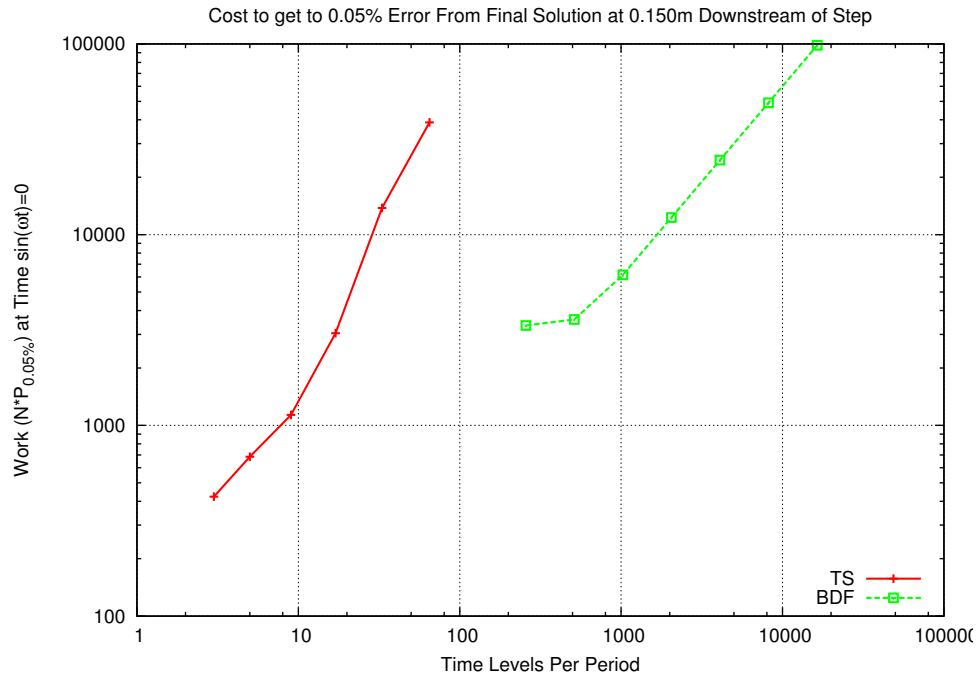


Figure 3.11: Discretization versus Cost for Burgers' Equation

As Figure 3.11 is only comparing cost to convergence, it is not very useful in determining cost to a given level of accuracy. Figure 3.12 shows cost versus error for the same simulations. The value used as truth is the result from the Time-Spectral Method with 65 time-levels per period. This was chosen as both the Time-Spectral

method and BDF were converging to the same value, but BDF was converging much more slowly. Nevertheless, both the Time-Spectral Method and the BDF simulations were within 0.02% of each other at 17 and 16,385 time levels per period for the Time-Spectral Method and BDF respectively. The value of the simulation at 0.05% convergence was used for comparison as it represents where each time discretization was approximately converged to its periodic steady state. Figure 3.12 clearly shows that spectral accuracy is desirable for high-accuracy non-linear simulations in order to reduce the number of time levels required per period. Furthermore, Figure 3.12 shows that the Time-Spectral Method only gives about a 2x reduction in computational cost when poor accuracy is desired. For higher accuracy simulations, however, the Time-Spectral Method is showing over an order of magnitude time savings. Additionally, the slope of the curve for Time-Spectral Method is much steeper than the slope for BDF, until the 0.05% limit is reached. This indicates that as higher and higher accuracy is required, the Time-Spectral Method will perform faster than BDF in comparison.

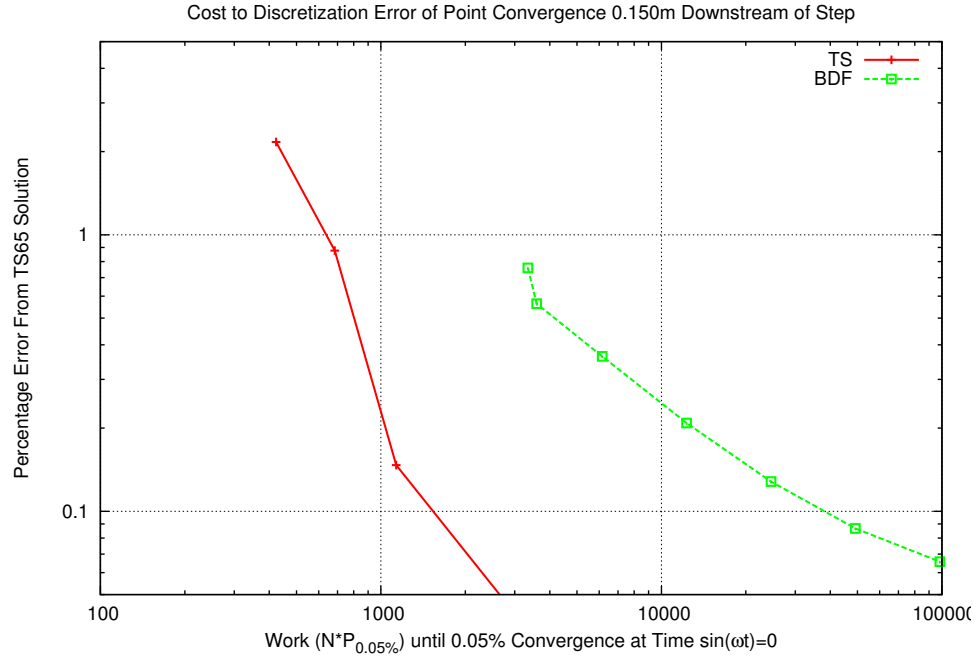


Figure 3.12: Error at a Point versus Cost for Burgers' Equation

Time-Spectral Method is examined for both the Linear Advection-Diffusion Equation and Burgers' Equation. These computations verify the implementation of the Time-Spectral Method for extending to more complex work. Spectral accuracy is shown to produce dramatically higher fidelity than conventional time-marching techniques. The Time-Spectral Method is capable of simulating pure advection, pure diffusion, and combined cases with the Linear Advection-Diffusion Equation. The Time-Spectral Method shows great promise in handling nonlinearities as shown by the results using Burgers' equation. The Time-Spectral Method not only had much greater fidelity than the conventional Backward Difference Formula, but also shows much greater stability, being able to stably solve the backward-facing step case with a second order scheme with dramatically fewer time levels than the Backward Difference Formula. A cost comparison demonstrates that the Time-Spectral Method can use much less computational power than conventional time marching for a given error.

## CHAPTER IV

# Incompressible Laminar Flow

### 4.1 Time-Spectral Algorithm for the Navier-Stokes Equations

Burgers' Equation demonstrated the application of the Time-Spectral Method for a nonlinear problem using the block-Jacobi algorithm with under-relaxation. After using Burgers' Equation to gain understanding of the time-spectral method, the method was extended to an incompressible Navier-Stokes Equations solver. The Navier-Stokes Equations are given in Equation 4.1.1 for incompressible flow of a Newtonian fluid.

$$\rho \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla p + \mu \nabla^2 \vec{u} + \vec{F}_b \quad (4.1.1)$$

Here,  $\rho$  is the constant fluid density,  $\vec{u}$  is the fluid velocity,  $p$  is the pressure,  $\mu$  is the fluid viscosity, and  $\vec{F}_b$  represents any body forces. The incompressible Navier-Stokes Equations is completed by the continuity equation, which is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{u} = 0 \quad (4.1.2)$$

This reduces to  $\nabla \cdot \vec{u} = 0$  for constant  $\rho$ , meaning that velocity is divergence free for incompressible flow. This necessitates a method of pressure coupling in order to

maintain divergence free velocity.

As this algorithm is implemented in OpenFOAM<sup>®</sup>, there are a few implementation specific details. OpenFOAM<sup>®</sup> is a finite-volume implementation that uses Gauss's Theorem for discretization of the governing equations. This results in the divergence free condition being implemented on the boundaries of each cell. Using Gauss's Theorem,

$$\int_V (\nabla \cdot \vec{u}) dV = \int_S (\vec{u} \cdot \hat{n}) dS = 0 \quad (4.1.3)$$

Discretizing this relationship, and assuming a constant velocity on the face, results in:

$$\int_S (\vec{u} \cdot \hat{n}) dS \approx \sum_f u_f A_f = 0 \quad (4.1.4)$$

where the subscript  $f$  represents the cell faces.  $u_f$  is  $\vec{u}_f \cdot \hat{n}_f$ , which is also known as the face flux.  $A_f$  is the area of each face. This implementation results in two versions of  $\vec{u}$ , the cell-centered  $\vec{u}$  and the face flux. As the solution is advanced, the face-flux is maintained with the divergence free condition, but, due to interpolation error, the cell-centered  $\vec{u}$  does not maintain  $\nabla \cdot \vec{u}$  to the same tolerance.

The method used to compute the incompressible Navier-Stokes Equations solution with Time-Spectral Method is a novel implementation. The previous implementations of incompressible Navier-Stokes solvers use artificial compressibility (Jameson, 2009; Antheaume and Corre, 2011) for pressure coupling. This solution uses a projection method (Jasak, 1996; Issa, 1985) for pressure coupling. This projection method utilizes delayed discretization (Jasak, 1996; Rhie and Chow, 1983) to avoid checkerboard oscillations in the solution of the pressure variable. Under-relaxation is used on both velocity and pressure with implicit under-relaxation on velocity and explicit under-relaxation on pressure. See Section 2.3 for the discussion on under-relaxation. In this implementation, the block matrix is not solved completely, but only a single iteration is used before the matrix is updated for nonlinearities. More iterations could

be used, but results from Burgers' Equation showed fastest convergence when using only one iteration. Additionally, updating for nonlinearities at each time-step has not been shown to introduce new instabilities. The following algorithm describes the implementation, using  $\eta$  as an iteration counter, and the superscript  $*$  to denote intermediate values:

1. Set  $\eta = 0$
2. Guess  ${}_0\bar{u}^{\eta}$  and  ${}_0p^{\eta}$
3. Interpolate  ${}_0\bar{u}^{\eta}$  to faces to obtain face flux  ${}_0\bar{u}_f^{\eta}$
4. While  $Residual(\vec{u}, p) > Tolerance$ 
  - (a)  $\eta = \eta + 1$
  - (b) For  $n = 0 \dots N - 1$ 
    - i. Form Block Matrix and Source Term,  $\left[ \mathbf{R} \right]$  and  $\left\{ \mathbf{S} \right\}$  respectively.

$$\nabla \cdot {}_{\eta-1}u_f^n \bar{u}^{n*} - \nu \nabla^2 \bar{u}^{n*} = -\nabla {}_{\eta-1}p^n - \sum_{\substack{m=0 \\ m \neq n}}^{N-1} P_{(n-m)\eta-1} \bar{u}^m \quad (4.1.5)$$

$$\left[ \mathbf{R} \right] \left\{ \bar{u}^{n*} \right\} = \left\{ \mathbf{S} \right\} - \left\{ \nabla {}_{\eta-1}p^n \right\} \quad (4.1.6)$$

- ii. Apply Implicit Under-Relaxation to  $\left[ \mathbf{R} \right]$
- iii. Solve Equation 4.1.6 for  $\left\{ \bar{u}^{n*} \right\}$
- iv. Split  $\left[ \mathbf{R} \right] = \left[ a_p - \mathbf{H} \right]$  where  $\left[ a_p \right]$  is the central coefficients (diagonal terms) and  $\left[ \mathbf{H} \right]$  is the neighbor terms (off diagonal terms) so Equation 4.1.6 becomes

$$\left[ \mathbf{a}_p \right] \left\{ \bar{u}^{n*} \right\} = \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} + \left\{ \mathbf{S} \right\} - \left\{ \nabla {}_{\eta-1}p^n \right\} \quad (4.1.7)$$



v. Compute intermediate velocity field disregarding pressure

$$\left\{ \bar{u}^{n**} \right\} = \left[ a_p \right]^{-1} \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} + \left[ a_p \right]^{-1} \left\{ \mathbf{S} \right\} \quad (4.1.8)$$

vi. Interpolate  $\left\{ \bar{u}^{n**} \right\}$  to cell faces to obtain  $u_f^{n**}$

$$u_f^{n**} = \left( \left[ \mathbf{a}_p \right]^{-1} \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} \right)_f + \left( \left[ \mathbf{a}_p \right]^{-1} \left\{ \mathbf{S} \right\} \right)_f \quad (4.1.9)$$

vii. Derive equation for intermediate pressure from Equation 4.1.7

$$\left\{ \bar{u}^{n*} \right\} = \left[ \mathbf{a}_p \right]^{-1} \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} + \left[ \mathbf{a}_p \right]^{-1} \left\{ \mathbf{S} \right\} - \left[ \mathbf{a}_p \right]^{-1} \left\{ \nabla p^{n*} \right\} \quad (4.1.10)$$

$$\left[ \mathbf{a}_p \right]^{-1} \left\{ \nabla p^{n*} \right\} = \left[ \mathbf{a}_p \right]^{-1} \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} + \left[ \mathbf{a}_p \right]^{-1} \left\{ \mathbf{S} \right\} - \left\{ \bar{u}^{n*} \right\} \quad (4.1.11)$$

viii. Take divergence of Equation 4.1.11 on the faces

$$\begin{aligned} & \nabla \cdot \left( \left[ \mathbf{a}_p \right]^{-1} \left\{ \nabla p^{n*} \right\} \right)_f \\ &= \nabla \cdot \left( \left[ \mathbf{a}_p \right]^{-1} \left[ \mathbf{H} \right] \left\{ \bar{u}^{n*} \right\} \right)_f + \nabla \cdot \left( \left[ \mathbf{a}_p \right]^{-1} \left\{ \mathbf{S} \right\} \right)_f \\ & \quad - \nabla \cdot \left\{ u_f^{n*} \right\} \end{aligned} \quad (4.1.12)$$

$$\nabla \cdot \left( \left[ \mathbf{a}_p \right]^{-1} \left\{ \nabla p^{n*} \right\} \right)_f = \nabla \cdot \left\{ u_f^{n**} \right\} - \nabla \cdot \left\{ u_f^{n*} \right\} \quad (4.1.13)$$

ix. Solve Equation 4.1.13 for  $p^{n*}$  assuming  $\nabla \cdot \left\{ u_f^{n*} \right\} = 0$

x. Update face fluxes  ${}_{\eta} \bar{u}_f^n$  for  $p^{n*}$  to obtain divergence free face flux field

- xi. Explicitly Under-Relax  ${}_{\eta}p^n = {}_{\eta-1}p^n + \alpha_p (p^{n*} - {}_{\eta-1}p^n)$  where  $\alpha_p$  is the under-relaxation factor on pressure
- xii. Obtain updated  ${}_{\eta}\bar{u}^n$  for  ${}_{\eta}p^n$

$$\left\{ {}_{\eta}\bar{u}^n \right\} = \left\{ \bar{u}^{n**} \right\} - \left[ a_p \right]^{-1} \left\{ \nabla_{\eta} p^n \right\} \quad (4.1.14)$$

This method is a pressure projection method which bears direct similarity to conventional PISO and SIMPLE schemes. The differences are in the time derivative operator. The first difference is that all time-levels must be iterated over via a block-Jacobi approach. The next is the method of inclusion of the time derivatives in the source term. The Time-Spectral Method's summation is included in the source term whereas the BDF method has values in the source term and another value in the central coefficient.

## 4.2 Backward-Facing Step

The first test case for the incompressible Navier-Stokes solver is a backward-facing step with pulsating inflow. Figure 4.1 shows the geometry. The value at one point is used to observe the performance of the Time-Spectral Method. That location is shown in Figure 4.1. Table 4.1 shows the essential parameters. In particular, the reduced frequency  $\omega_{red}$  is moderately high which indicates a strong temporal dependence. The results presented in Figure 4.2 are shown for a second order upwind biased scheme. Note that only the time level corresponding to  $\cos(\omega t) = 1$  is plotted in Figure 4.2. This is done to ensure comparison at the same time level.

Two important facets of Time-Spectral Method can be seen in Figure 4.2. The first characteristic is improved accuracy. This can be seen in the plots (a)-(d), which show much better convergence of the Time-Spectral results over the BDF results, despite smaller changes in time-step size. The second characteristic of the Time-

Spectral Method shown in Figure 4.2 is how the Time-Spectral Method does not have monotonic convergence. This is shown by the wide spread of point values in the earlier computations of Figures 4.2(a) and 4.2(c).

Besides better accuracy, the other important characteristic of Time-Spectral Method is speed. Figure 4.3 shows that Time-Spectral Method can be 5 times faster in point convergence when comparing to a given level of accuracy. Here speed is measured by block matrices solved at a given time level. Notably, this is a very large difference. As typical errors are  $10^{-4}$  to  $10^{-6}$  for the entire flow field, this comparison is likely to be even more in favor of Time-Spectral Method for realistic simulations as better accuracy means it will take less computation from these values to reach lower point-to-point differences.

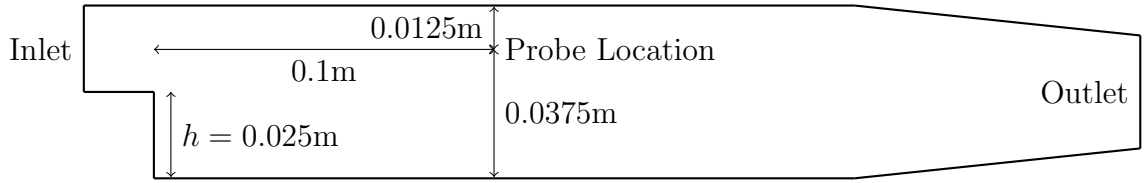


Figure 4.1: Backward-Facing Step Geometry for the Navier-Stokes Equations

Table 4.1: Parameters for the Backward-Facing Step Test Case for the Navier-Stokes Equations

$$\begin{aligned}
 \text{Inlet :} \quad & \vec{u} = U_0(1 + \sigma \sin(\omega t))\hat{i} \quad \partial p / \partial \hat{n} = 0 \\
 \text{Outlet :} \quad & \partial \vec{u} / \partial \hat{n} = 0 \quad p = 0 \\
 \text{Walls :} \quad & \vec{u} = 0 \quad \partial p / \partial \hat{n} = 0 \\
 \text{Re} = U_0 h / \nu = 250 \quad & \omega_{red} = \omega h / U_0 = 2.62 \quad \sigma = 0.1
 \end{aligned}$$

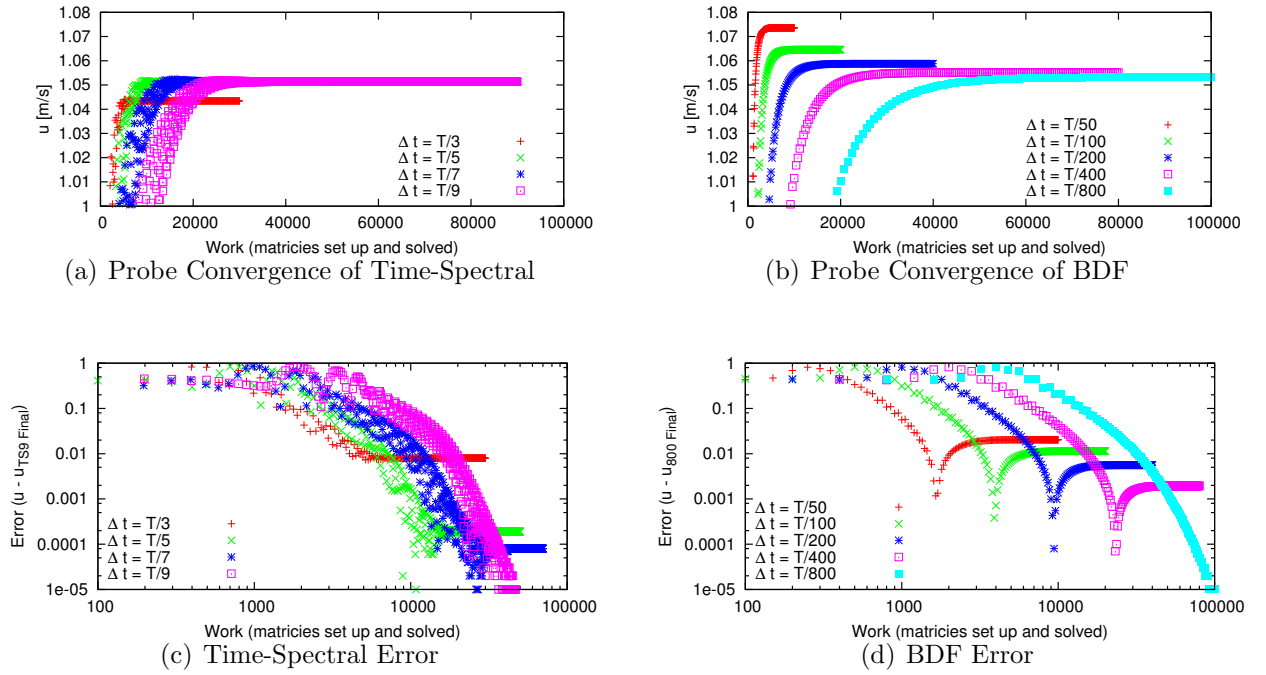


Figure 4.2: Results for Convergence of  $u$  at a Point over a Backward-Facing Step



Figure 4.3: Backward-Facing Step Work Comparison

### 4.3 Pitching Foil

Propellers and other rotating foils are a very common type of periodic flow of engineering. As such, the next flow to be investigated is a pitching foil. This pitching foil, along with the plunging foil discussed in Section 4.4, serve as two-dimensional surrogates of rotating foils, which can be subject to different flows at different points in their rotation. Pitching and plugging foils are also especially relevant to flapping wings which are becoming increasingly relevant with the emergence of Micro Air Vehicles as a research field. (Shyy et al., 2008)

In order to obtain airfoil movement, solid body mesh motion is employed using an ALE approach. This requires the specification of the velocity on the foil to be the same as the mesh motion. An exact mesh flux is used as opposed to the built in mesh motion that uses a first order approximation of mesh flux. This exact mesh flux is necessary as the approximate mesh flux becomes very inaccurate with the large time steps used with the Time-Spectral Method. Additionally, the approximate mesh flux is overly inaccurate far away from the center of rotation because the mesh flux is proportional to the distance from the center of rotation. For this reason, it is helpful to use the exact mesh flux for the BDF simulations as well. The boundary conditions are as specified in Table 4.2. The grid motion is  $\vec{U}_{Grid} = (\vec{\theta}\omega \cos(\omega t)) \times (\vec{r} - \vec{r}_0)$ , that is, a prescribed pitching motion. The maximum pitch angle is  $3^\circ$  and the center of rotation is the quarter-chord.

Table 4.2: Parameters for the Navier-Stokes Equations Pitching Airfoil Case

$$\begin{aligned}
 \text{Inlet :} & \quad \vec{u} = \vec{U}_\infty \quad \partial p / \partial \hat{n} = 0 \\
 \text{Outlet :} & \quad \partial \vec{u} / \partial \hat{n} = 0 \quad p = 0 \\
 \text{Top and Bottom :} & \quad \partial \vec{u} / \partial \hat{n} = 0 \quad \partial p / \partial \hat{n} = 0 \\
 \text{Walls :} & \quad \vec{u} = \vec{U}_{Body} \quad \partial p / \partial \hat{n} = 0 \\
 Re = U_0 c / \nu = 1050 & \quad \omega_{red} = \frac{2\pi}{T} \frac{c}{2U_\infty} = 3.0
 \end{aligned}$$

Airfoil Geometry: NACA0012

The mesh is two-dimensional with 13,000 cells. The farfield boundaries are very close to the foil in order to reduce the computational domain. The boundaries extend 5 chords in front, 5 chords above and below, and 10 chords behind. Figure 4.4 shows the foil relative to the domain along with an example flow field at  $\sin \omega t = 0$ .

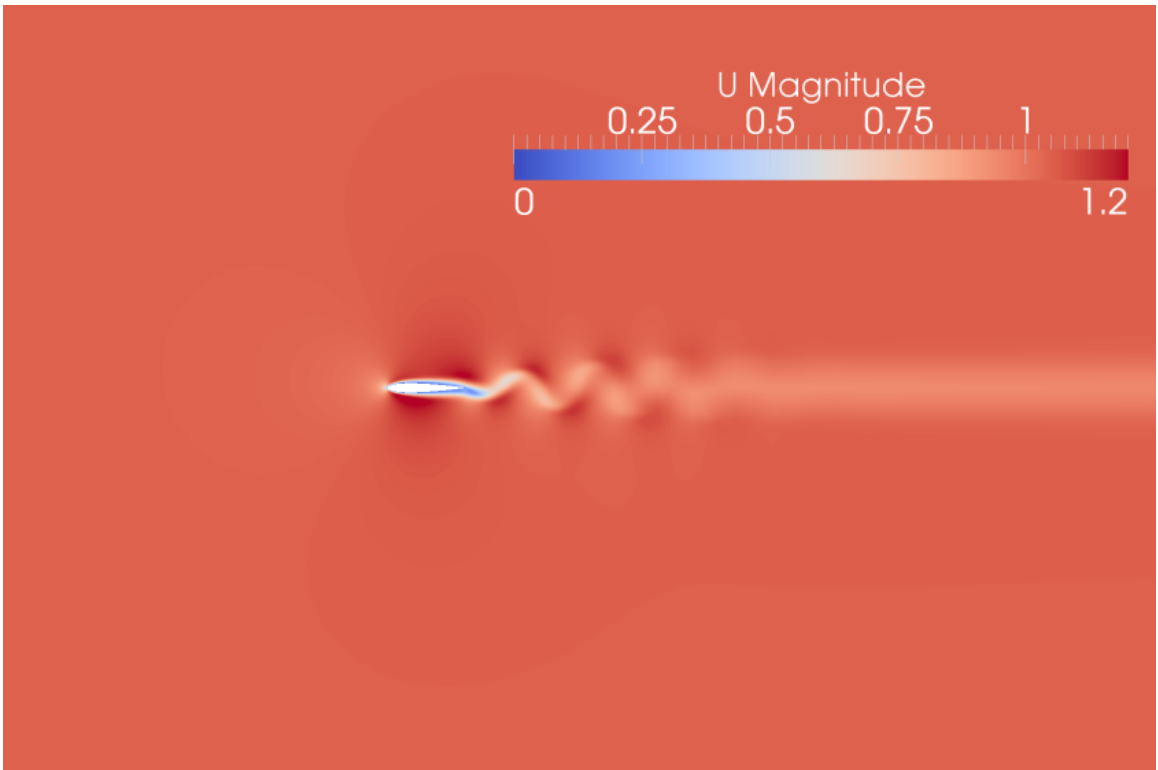


Figure 4.4: Foil Domain

The force polar diagram is shown in Figure 4.5. Note that TS9 is being used to denote Time-Spectral Method with nine time-levels per period. The BDF results are computed using 1000 time levels per period. The results indicate that drag has a mean offset, while lift is symmetric about 0. More importantly, Time-Spectral Method and BDF match up well indicating that the Time-Spectral Method is capable of matching the current standard.

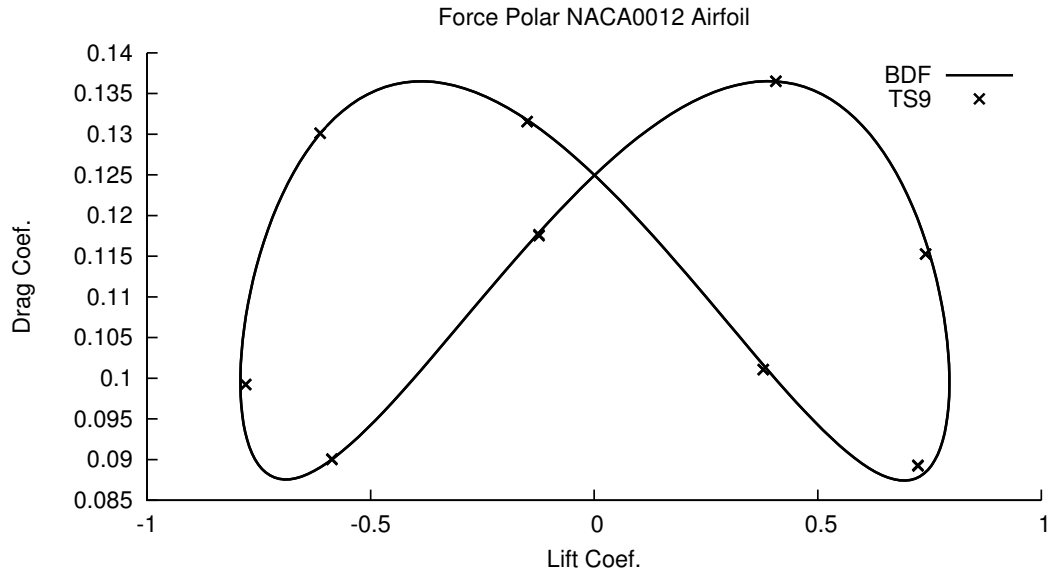


Figure 4.5: Force Polar For Laminar Pitching Foil

Figure 4.6 shows the  $C_P = (P - P_\infty)/(\frac{1}{2}\rho U_\infty^2)$  distribution over the airfoil. The time level is  $\sin(\omega t) = 0$  which corresponds to  $\alpha = 0$ , but  $\alpha$  is decreasing which is why the integral of pressure is nonzero. Both methods are very comparable at this time point. The differences in the plots are on the lower surface near what may be mild flow separation. The details of this separation are discussed in Lian and Shyy (2007). The differences in computations are in the form of checkerboarding, which is a numerical problem from co-located primitive variable storage. The Time-Spectral solution exhibits a more smooth solution than BDF. It would typically be expected that the more dissipative method, i.e. BDF, would smooth the solution more than the more accurate method, i.e. the Time-Spectral Method. Shen et al.

(2001) discusses this problem of checkerboarding from co-located variable storage and shows how using small time-steps may exaggerate the problem with Rhie-Chow interpolation (Rhie and Chow, 1983) when using standard time-marching methods. As the Time-Spectral Method does not have this problem, it is especially valuable for high accuracy computations where BDF can not be used due to this checker boarding problem.

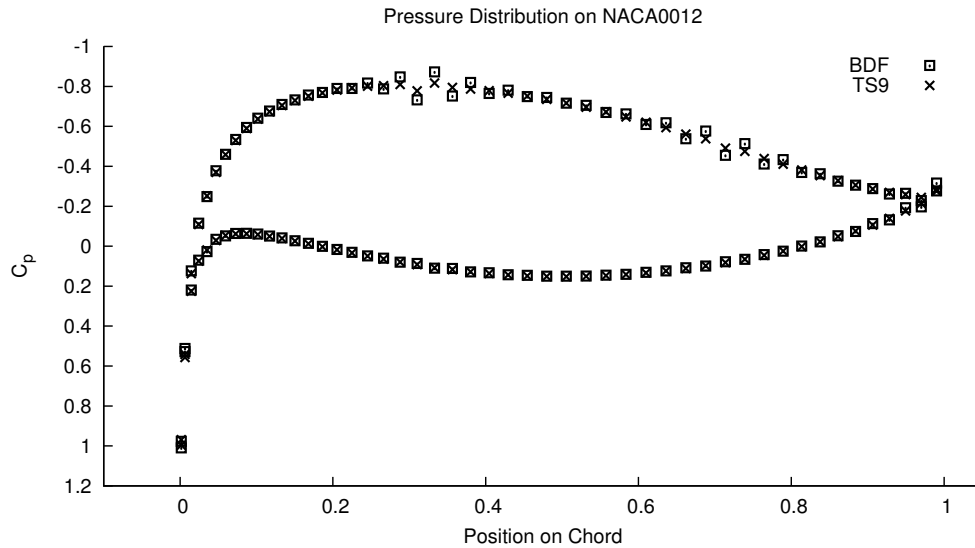


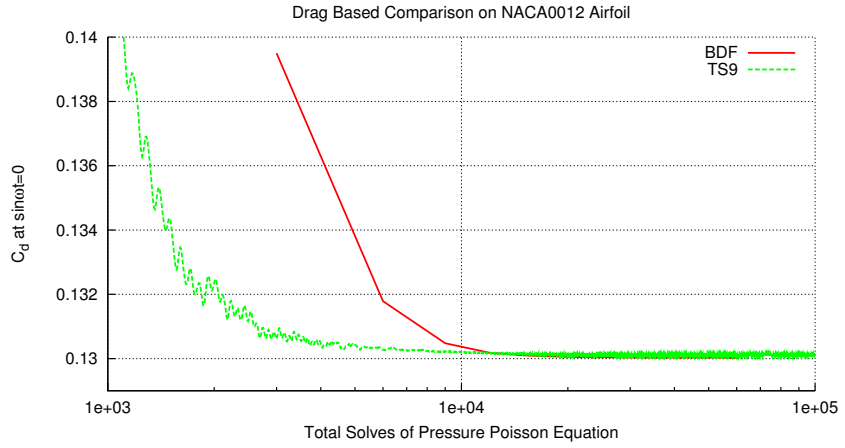
Figure 4.6:  $C_P$  Distribution For Laminar Pitching Foil at  $T = 0$

Figure 4.7 shows the force history of the pitching foil as a function of block solves. Specifically, Pressure Poisson Equation solves are used for computational cost as the Pressure Poisson Equation takes 1 to 2 orders of magnitude longer to solve than the velocity equations. Additionally, the number of solves of the velocity equations is directly proportional to the number of solves of the Pressure Poisson Equation. For the BDF computations, computational cost is directly proportional to length of simulation time to reach periodic steady state. For the Time-Spectral Method, the computational cost is proportional to the number of fully time coupled matrix sweeps. For both methods, the computational cost is also proportional to the number of time-steps per period. The BDF simulation used 1,000 constant time-steps per

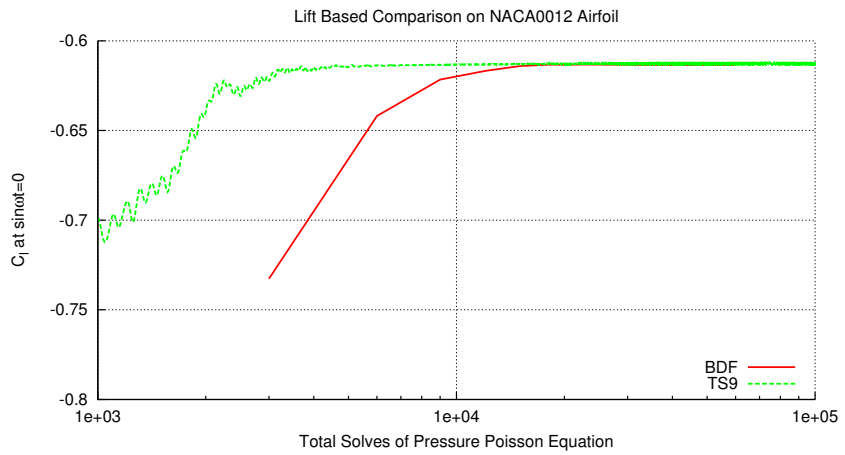


period which corresponds to a maximum Courant Number of 0.1 to 0.4 depending on location in the cycle. It should be noted that 1,000 time-steps per period can be considered very coarse as Ko and McCroskey (1997) used 10,000 time steps per cycle to simulate a similar pitching airfoil. The simulation from both BDF and Ko and McCroskey (1997) used three sub-iterations per time-step in order to reduce linearization and factorization error.

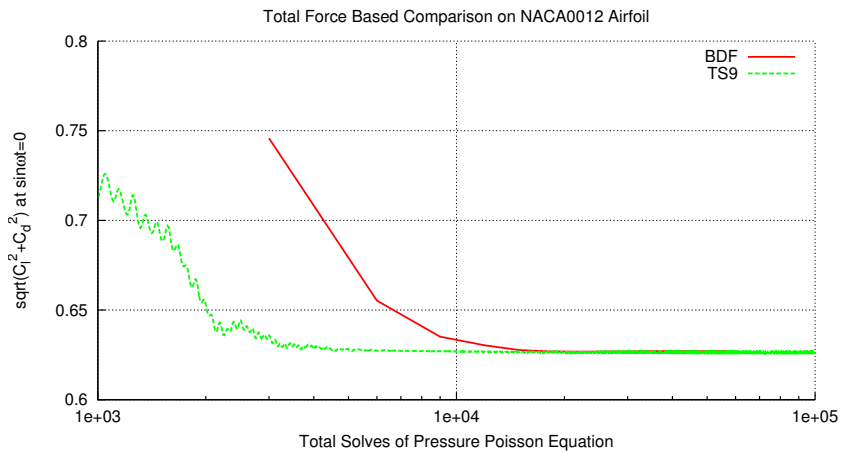
All three figures in Figure 4.7 show the trend of the Time-Spectral Method very quickly finding the periodic steady state value. Also both the Time-Spectral Method and BDF are approaching the same values, which suggests consistency in approximation. Seen most clearly in Figure 4.7(b), the TS9 simulation is effectively converged by 3,000 solves of the Pressure Poisson Equation. All three subfigures show how BDF is lagging behind the Time-Spectral Method. The peaks in force in Subfigures 4.7(b) and 4.7(c) show that the BDF simulation doesn't converge until about 15,000 cycles. This gives the Time-Spectral simulation a speedup of five times.



(a) Drag as Function of Cost



(b) Lift as Function of Cost



(c) Total Force as Function of Cost

Figure 4.7: Comparison of Forces as Function of Cost

## 4.4 Plunging Foil

In order to test the foil further in two dimensions, plunging motion is investigated. The same grid is used for the plunging foil as is used for the pitching foil described in Section 4.3. Also, the boundary conditions given in Table 4.2 are maintained. As the motion is translational, the grid velocity is described by  $\vec{U}_{Grid} = \vec{A} \cos(\omega t)$ . For this test case,  $\omega_{red} = \frac{2\pi}{T} \frac{c}{2U_\infty} = 1.0$ . Reynolds number is set to 1000.

Figure 4.8 shows the relationship between lift and drag for the plunging foil. Notably, during part of the cycle, drag shows a forward force which is expected for plunging foils as the plunging motion can create a velocity that induces an angle of attack such that the lift vector is pointed in the forward direction, thus offsetting the drag. This phenomena is also studied for sailing vessel appendages in unsteady motions (Milgram, 1998). Figure 4.8 also shows that BDF and Time-Spectral match up very well, although not as well as the pitching foil results in Figure 4.5. Figure 4.8 also shows a fairly wide range of under-relaxation parameters for the time-spectral method that converge to the same result. This demonstrates the under-relaxation parameter has little effect on the final solution, as would be expected.

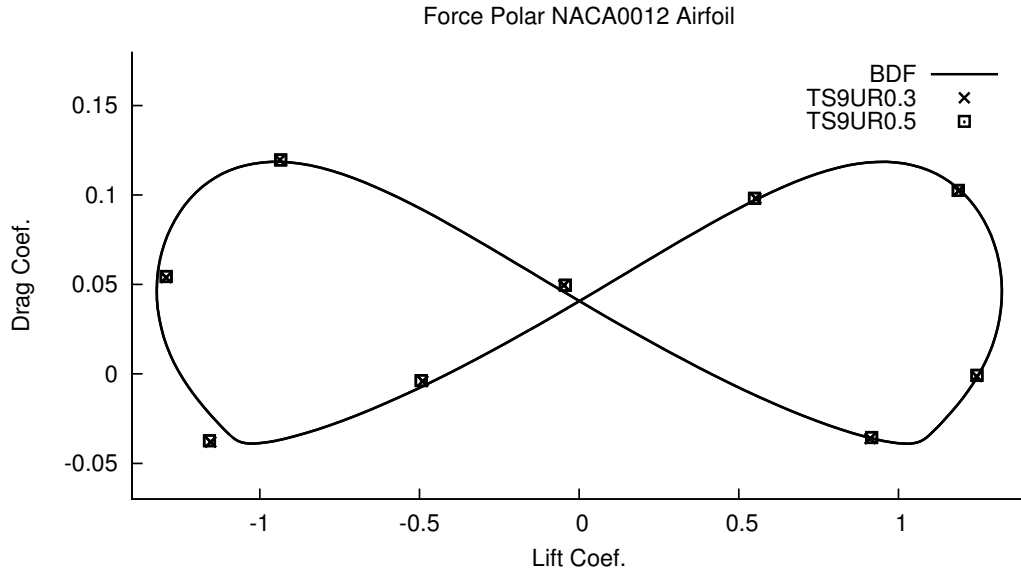


Figure 4.8: Force Polar For Laminar Plunging Foil

Figure 4.9 shows the pressure distribution along the airfoil. The stagnation point is offset from the front of the foil. This is due to the angle of attack induced from the foil motion as well as the forward velocity. Like the pitching foil case, BDF and Time-Spectral match up very well. There is a slight discrepancy between them, however, at the trailing edge. This is examined in Figure 4.10 where the vorticity of each method is shown near the trailing edge. There is a small but apparent change in vorticity magnitude between Time-Spectral and BDF on the top surface just before the trailing edge. Namely, in the BDF simulation shown in Figure 4.10(b), there is a stronger vortex than in the Time-Spectral simulation in Figure 4.10(a). This vortex is higher off the trailing edge and as such, it draws vorticity off of the foil, which decreases the pressure at the trailing edge as shown in Figure 4.9. This difference is due to high frequency content which is not resolved by TS9. This high frequency content is resolved by a TS17 simulation that is not shown.

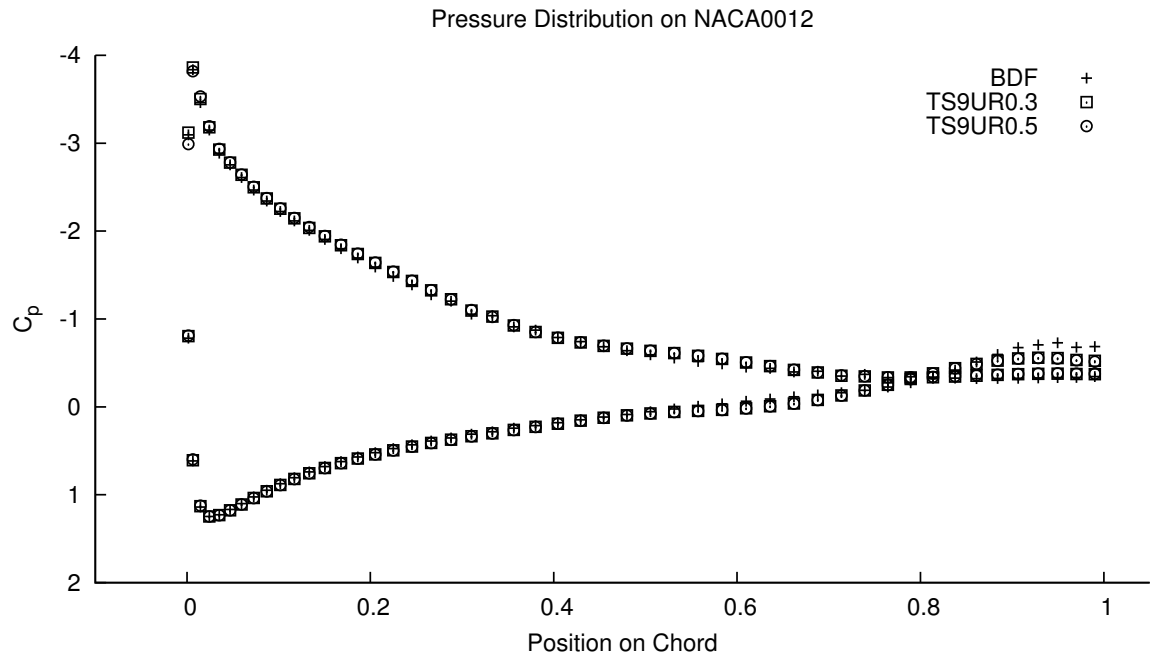


Figure 4.9:  $C_P$  Distribution For Laminar Plunging Foil at  $\sin \omega t = 0$

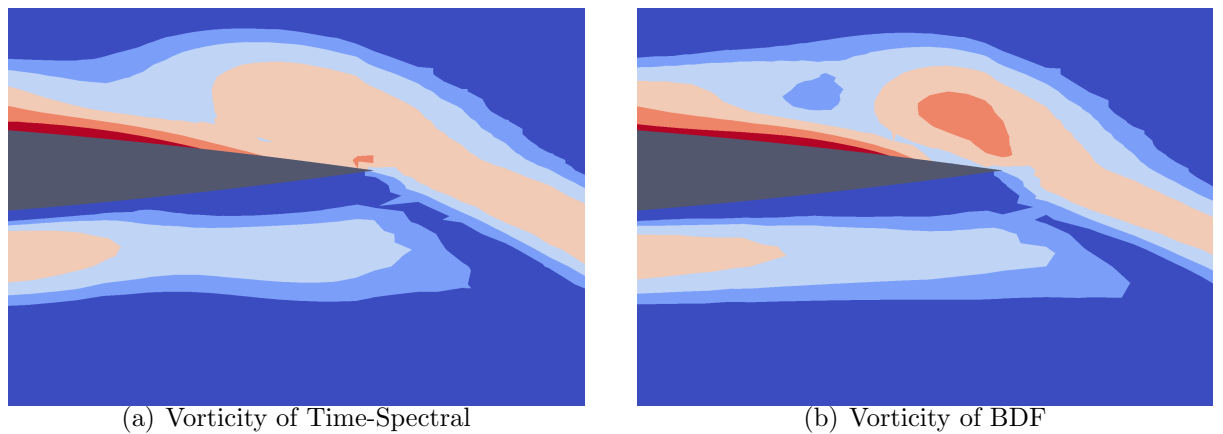
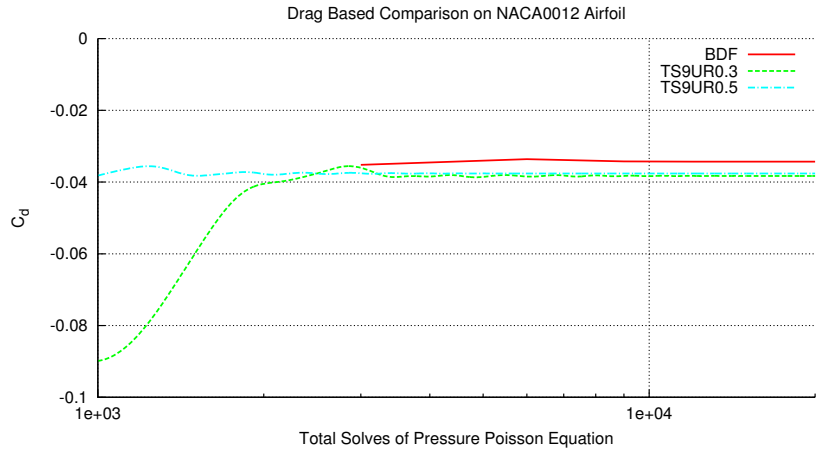


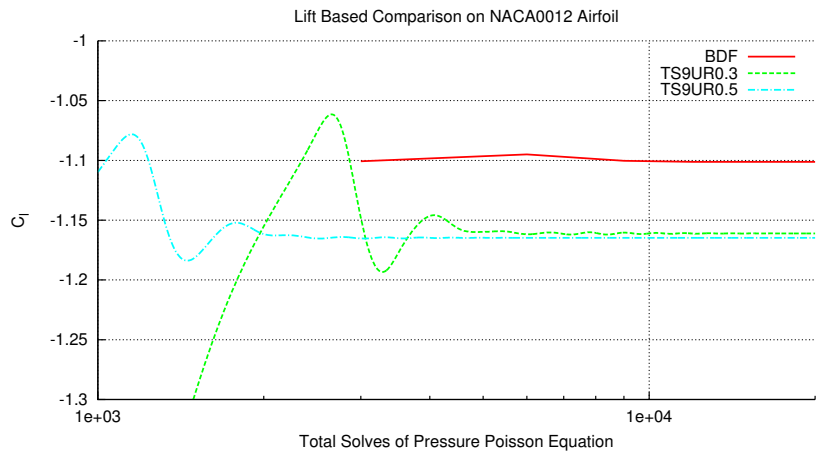
Figure 4.10: Vorticity Near the Trailing Edge of a Pitching Airfoil

Figure 4.11 shows a comparison between methods using the force at time  $\sin \omega t = 0$  as the objective function and the solves of the Pressure Poisson Equation as the cost. Several things can be noticed from these graphs. Firstly, the BDF formulation and

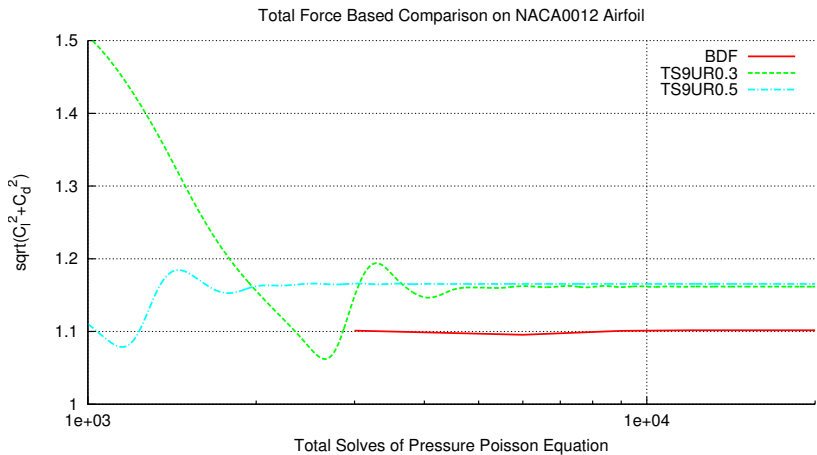
the Time-Spectral formulation do not converge to the same value. This discrepancy is likely due to the small changes in the trailing edge vortex shown above. Therefore, the BDF results most closely match the highly time-resolved case. Also, as described in Section 3.1.1, the choice of under-relaxation factor is important for the speed of the Time-Spectral algorithm. In Figure 4.11(b), when using an under-relaxation factor of 0.3 the oscillations in predicted lift force do not die out until about 10,000 pressure solves. This is about the same amount of work as the BDF formulation, which utilizes 1,000 solves per period. Increasing the under-relaxation factor to 0.5, however, allowed the lift force variation to die out in about 4,000 pressure solves. This is a 2.5x saving in computational time. While these results show that the choice of under-relaxation parameter is important for speed of convergence, the speed of convergence is not as sensitive to the under-relaxation parameter as linear advection in Section 3.1.1, which changed 2 orders of magnitude in computational time for a 0.08 change in the under-relaxation parameter. It should be noted that an under-relaxation factor of 0.6 was tested, but made computations about as quickly as the case with a 0.3 under-relaxation factor. Therefore, there may be an optimal under-relaxation factor that is not based purely on using the highest stable under-relaxation factor. Peric (1985) and Ferziger and Peric (2002) found a similar result for the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) scheme.



(a) Drag as Function of Cost



(b) Lift as Function of Cost



(c) Total Force as Function of Cost

Figure 4.11: Comparison Of Forces as Function of Processing Cost

## CHAPTER V

# Incompressible Turbulent Flow

Fluid flow at high Reynolds Number exhibits turbulence. The effects of turbulence can have a dramatic effect on fluid flow as small instabilities grow into large flow features at high Reynolds Numbers. This results in excessive computational cost as extremely fine space and time scales must be resolved even though the large scale features are more important to the flow. Indeed, for a ship flow, it is impossible to discretely model the flow down to the smallest scales. A common method around this issue is to model the smaller turbulence scales. This ensures that turbulence is accounted for, but does not give turbulence the full resolution, or computational cost, that it would otherwise require.

### 5.1 Time-Spectral Algorithm for the URANS Equations

The Unsteady Reynolds Averaged Navier-Stokes (URANS) equations are a time-averaged form of the Navier-Stokes Equations. By using time-averaged equations, the time-averaged component of the flow variables can be directly solved. This approach models the high frequency components of the flow variables so as to significantly reduce computational expense. Implementation of the model in the Time-Spectral framework is relatively straightforward as equations for the modeled terms frequently contain a time-derivative. This is supported by the fact that many compressible



flow implementations of either the harmonic balance technique or the Time-Spectral Method have been extended to turbulent flows (Hall et al., 2002; Thomas et al., 2004; Vassberg et al., 2005; Gopinath, 2007). The Time-Spectral Method is directly applied to this time-derivative and the turbulence modeling equations are thus put into the block-Jacobi framework. Additionally, only the time-averaged components are solved for, which are periodic. The aperiodic components of turbulence are accounted for in the averaging. Another common turbulence modeling strategy is Large Eddy Simulations (LES) which splits the stresses into spatially resolved and unresolved stresses. The unresolved stresses are called sub-grid stresses and are explicitly or implicitly modeled. It remains an open research question as to whether or not LES turbulence modeling can be used with the Time-Spectral Method. Although, due to the very high temporal accuracy requirements of LES, using the Time-Spectral Method with LES could produce dramatic computational savings due to spectral accuracy of the Time-Spectral Method.

In order to get the URANS equations from the Navier-Stokes Equations, equation 4.1.1 is time-averaged, resulting in the following equations:

$$\overline{\rho \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right)} = -\overline{\nabla p} + \overline{\mu \nabla^2 \vec{u}} + \overline{\vec{F}_b} \quad (5.1.1)$$

The Reynold's Decomposition splits  $\vec{u}$  and  $p$  into time averaged components and instantaneous components such that:

$$\vec{u} = \overline{\vec{u}} + \vec{u}' = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} \vec{u} dt + \vec{u}' \quad (5.1.2)$$

$$p = \overline{p} + p' = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} p dt + p' \quad (5.1.3)$$

Equation 5.1.1 now becomes:

$$\rho \left( \frac{\partial \bar{\vec{u}}}{\partial t} + \bar{\vec{u}} \cdot \nabla \bar{\vec{u}} + \overline{\vec{u}' \cdot \nabla \vec{u}'} \right) = -\nabla \bar{p} + \mu \nabla^2 \bar{\vec{u}} + \bar{\vec{F}}_b \quad (5.1.4)$$

where  $\partial \vec{u}' / \partial t$  is assumed small.

Using tensor identities,  $\overline{\vec{u}' \cdot \nabla \vec{u}'} = \overline{\nabla \cdot \vec{u}' \vec{u}'} = \nabla \cdot \overline{\vec{u}' \vec{u}'}$ . This is rearranged as:

$$\rho \left( \frac{\partial \bar{\vec{u}}}{\partial t} + \bar{\vec{u}} \cdot \nabla \bar{\vec{u}} \right) = -\nabla \bar{p} + \nabla \cdot (\mu \nabla \bar{\vec{u}} - \rho \overline{\vec{u}' \vec{u}'}) + \bar{\vec{F}}_b \quad (5.1.5)$$

The  $\overline{\vec{u}' \vec{u}'}$  term is called the Reynold's Stress and will be modeled. The other terms only involve the averaged variables and are solved like the laminar flow equations.

The turbulence model used is the Spalart-Allmaras one-equation model (Spalart and Allmaras, 1994, 1992). The Spalart-Allmaras turbulence closure model is chosen because it is a one-equation, linear eddy viscosity model. This turbulence closure model is widely used due to its combination of low computational cost and relatively good accuracy. These models, like many URANS turbulence closure models, contain the time derivative term that can be formed in the Time-Spectral framework.

The Spalart-Allmaras turbulence closure model uses the Boussinesq Approximation of isotropic turbulence, which allows the Reynold's Stress term to be modeled as an eddy viscosity coefficient times the Reynolds-averaged strain rate. This simplification allows the following substitution for the Reynold's Stress:

$$-\overline{\rho \vec{u}' \vec{u}'} = \mu_t \left( \nabla \bar{\vec{u}} + \nabla \bar{\vec{u}}^T \right) - \frac{2}{3} \rho k \bar{\vec{I}} \quad (5.1.6)$$

where  $\mu_t$  is the eddy viscosity,  $\bar{\vec{I}}$  is the identity matrix, and  $k = \frac{1}{2} \text{Tr} \left( \overline{\vec{u}' \vec{u}'} \right)$  is the mean turbulent kinetic energy. Dropping the overbars on the averaged variables, the

following equation is found:

$$\rho \left( \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla \left( p + \frac{2}{3} \rho k \right) + \nabla \cdot [(\mu + \mu_T) (\nabla \vec{u} + \nabla \vec{u}^T)] \quad (5.1.7)$$

where  $\mu + \mu_T$  is often combined to  $\mu_{eff} = \mu + \mu_T$  and  $p + \frac{2}{3} \rho k$  is often combined into a single variable.  $\rho \nu_T = \mu_T$  is spatially varying and is calculated by the turbulence model equation.

The Spalart-Allmaras turbulence model is formulated for incompressible flow such that  $\nu_t = \mu_t/\rho$ . The model then solves for  $\nu_t = \tilde{\nu} f_{v1}$  with  $f_{v1} = \chi^3/(\chi^3 + C_{v1}^3)$ , where  $\chi = \tilde{\nu}/\nu$ . The transport equation for  $\tilde{\nu}$  is as follows:

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + \vec{u} \cdot \nabla \tilde{\nu} &= C_{b1} (1 - f_{t2}) \tilde{S} \tilde{\nu} \\ &+ \frac{1}{\sigma} \nabla \cdot ((\nu + (1 + C_{b2}) \tilde{\nu}) \nabla \tilde{\nu}) - \frac{C_{b2}}{\sigma} \tilde{\nu} \nabla^2 \tilde{\nu} - \left[ C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right] \left( \frac{\tilde{\nu}}{d} \right)^2 \end{aligned} \quad (5.1.8)$$

The associated functions and coefficients are given in Ashford (1996) which has small changes from Spalart and Allmaras (1994) for numerical stability. Equation 5.1.8 shows turbulence modeling terms for advection ( $\partial \tilde{\nu}/\partial t + \vec{u} \cdot \nabla \tilde{\nu}$ ), production ( $C_{b1} [1 - f_{t2}] \tilde{S} \tilde{\nu}$ ), diffusion ( $(1/\sigma) \nabla \cdot [(\nu + (1 + C_{b2}) \tilde{\nu}) \nabla \tilde{\nu}] - (C_{b2}/\sigma) \tilde{\nu} \Delta \tilde{\nu}$ ), and destruction ( $- [C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2}] (\tilde{\nu}/d)^2$ ).

## 5.2 Pitching Foil

The pitching foil provides testing grounds for testing the implementation of turbulence with the Time-Spectral Method. As in the laminar case, the grid motion is  $\vec{U}_{Grid} = (\vec{\theta} \omega \cos \omega t) \times (\vec{r} - \vec{r}_0)$  with a maximum pitch angle of  $3^\circ$  about the quarter-chord. Table 5.1 shows the parameters of the tested case. The first notable difference is the addition of turbulent boundary conditions, including a wall function on the foil. The next difference is the higher Reynolds' Number, and the flow is turbulent.

The final difference is that the reduced frequency is smaller, as it was difficult to find numerical stability at the original  $\omega_{red}$  of 3.0. The grid used was the same as that used for the incompressible foil simulations.  $y^+$  was computed to be between 130 and 430. These values indicate that the grid is resolved to the log-layer and thus wall functions represent a good approximation of the flow.

Table 5.1: Parameters for the Turbulent Pitching Airfoil Case

$$\begin{array}{llll}
 \textit{Inlet} : & \vec{u} = \vec{U}_\infty & \partial p / \partial \hat{n} = 0 & \nu_t / \nu = 0.3 \\
 \textit{Outlet} : & \partial \vec{u} / \partial \hat{n} = 0 & p = 0 & \nu_t / \nu = 0.3 \\
 \textit{Top and Bottom} : & \partial \vec{u} / \partial \hat{n} = 0 & \partial p / \partial \hat{n} = 0 & \nu_t / \nu = 0.3 \\
 \textit{Walls} : & \vec{u} = \vec{U}_{Grid} & \partial p / \partial \hat{n} = 0 & \textit{Wall Function} \\
 & Re = U_0 c / \nu = 3.1 \times 10^6 & \omega_{red} = \frac{2\pi}{T} \frac{c}{2U_\infty} = 1.0 & 
 \end{array}$$

Airfoil Geometry: NACA0012

Figure 5.1 shows the lift to drag polar plot for the turbulent pitching foil. On this plot, the Time-Spectral Methods is abbreviated “TS” and Backward Difference Formula is abbreviated “BDF”. The number after each abbreviation is the number of time-levels per period. The Time-Spectral Methods also contain an under-relaxation factor. Comparing the two curves using BDF time-marching, it is clear that using only 200 time-steps per period is insufficient for resolving the flow field with BDF. As time-discretization is improved, the force histories show a wider spread of lift, and a shift in drag. Both discretizations of the Time-Spectral Method shown are in good agreement with the more highly resolved BDF computation, even the case with a mere 5 time steps per period. The Time-Spectral Method with smaller  $\Delta t$  does show closer agreement to the highly resolved BDF computation than the lower resolution Time-Spectral case.

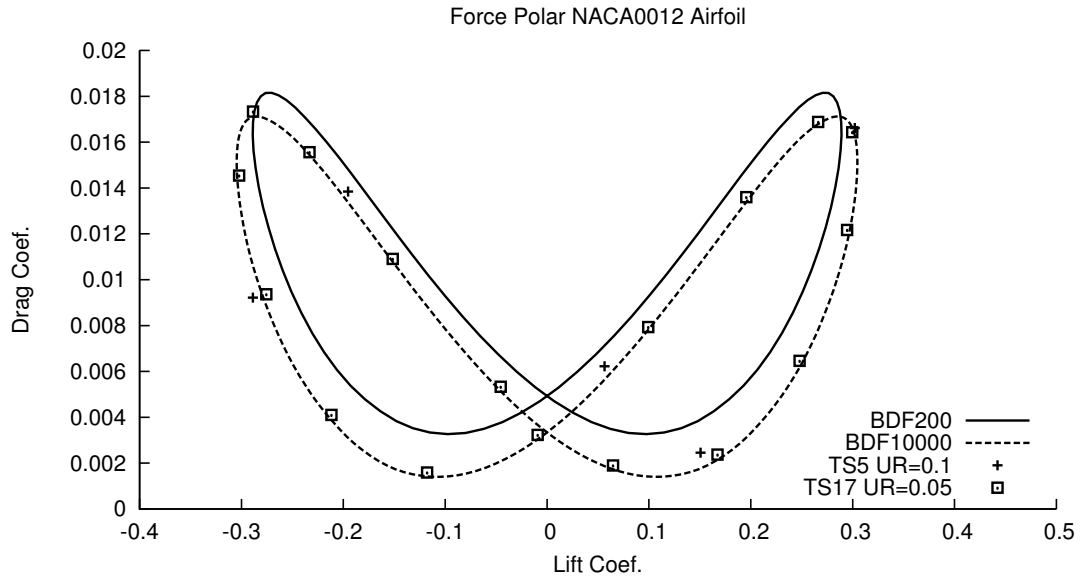


Figure 5.1: Drag versus Lift for Turbulent Pitching Foil

Figure 5.2 shows the comparison of pressure coefficient along the airfoil surface. The pressure matches almost perfectly between the two methods. Also, the pressure over the foil is very smooth, unlike the laminar flow shown in Figure 4.6. This difference is most likely due to delay of boundary layer separation in turbulent flows as explained by Lian and Shyy (2007).

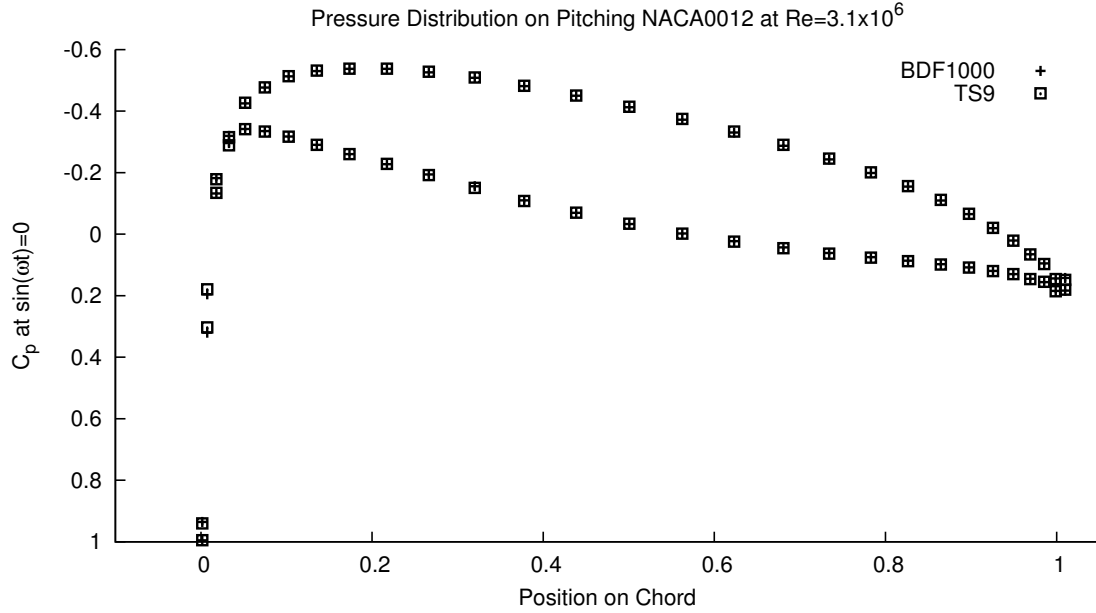


Figure 5.2:  $C_p$  Comparison for Turbulent Pitching Foil

### 5.3 Plunging Foil

A plunging foil provides a very challenging test case for the Time-Spectral Method to evaluate as the velocities from motion perpendicular to the flow can create very large angles of attack. These large angles of attack lead to flow separation and stall. The two-dimensional plunging foil is representative of the flow around flapping wings as well as winged keels in waves. The plunging foil tested has a Reynolds Number of 3.1 million and  $\omega_{red} = \frac{2\pi}{T} \frac{c}{2U_\infty} = 1$ . The same foil mesh that was used on the previous airfoil problems has been used to evaluate the case of a turbulent plunging foil. For this case, the computed  $y^+$  values varied between 5 and 680, with an average of 220. These values indicate that the near wall spacing was realistic for a coarse grid in early

investigations. The under-relaxation factor used for the Time-Spectral Method cases was 0.3 for 5 and 9 time levels per period, and dropped to 0.1 for 17 and 33 time levels per period.

Figure 5.3 shows a comparison of pressure coefficients for the most highly resolved cases that were tested of the turbulent plunging foil. Both methods display some oscillation at the leading edge of the foil that indicates more spatial discretization is necessary to properly represent the flow. BDF displays further oscillations that appear to be the result of checkerboarding. The Time-Spectral Method does not show these checkerboard oscillations. This is similar to the laminar pitching foil case in Figure 4.6. Shen et al. (2001) mathematically shows how the checkerboarding originates from co-located variable storage. This article demonstrates that using smaller time steps makes this problem worse. Figure 5.4 investigates this problem. Figure 5.4 demonstrates how, when using BDF, larger time steps maintain a smoother solution. This is shown in how the more finely time is discretized, the larger these checkerboarding oscillations become. As this problem has not appeared with the Time-Spectral Method, the ability to avoid this checkerboarding is another reason why it is beneficial to use the Time-Spectral Method for highly accurate solutions.

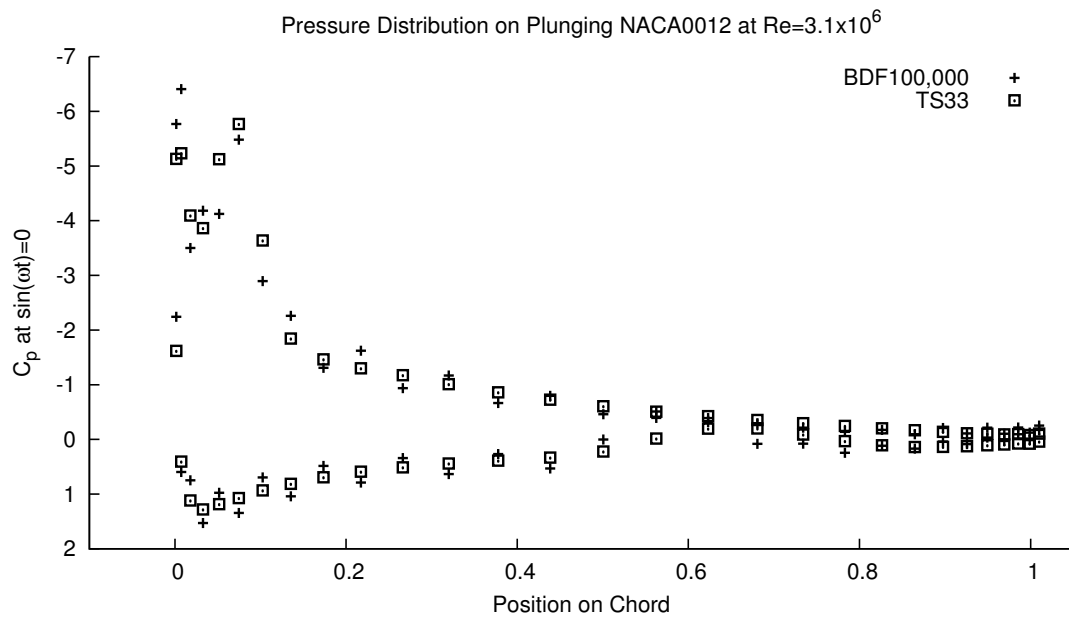


Figure 5.3:  $C_p$  Comparison for Highly Resolved Time-Spectral and BDF Case



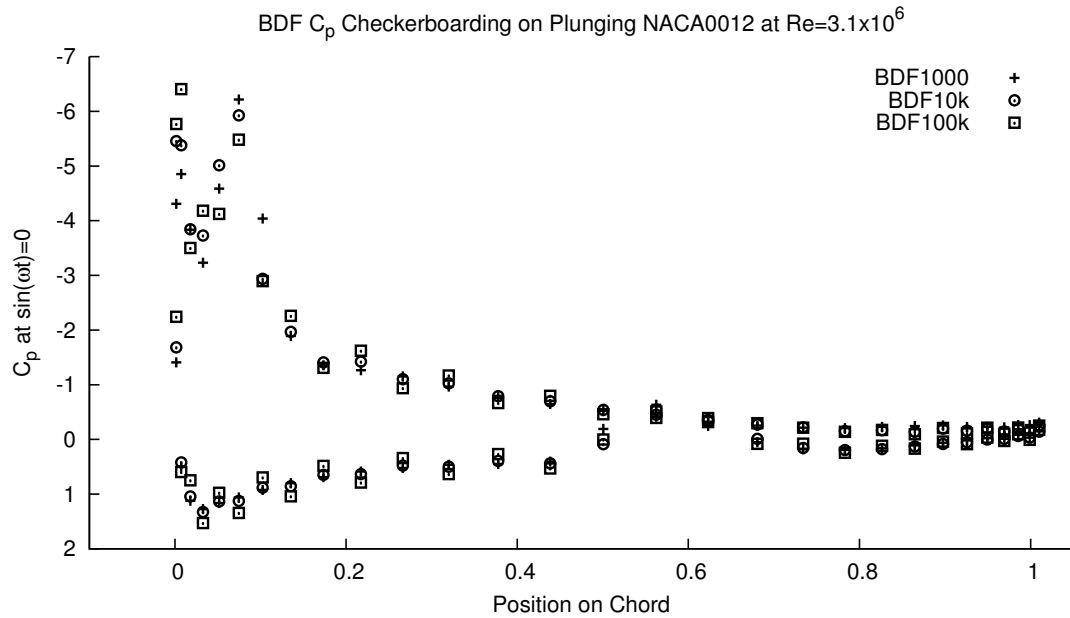


Figure 5.4:  $C_p$  Comparison for BDF Case with Varying Resolution

Figure 5.5 displays the lift to drag polar for the turbulent plunging foil. This figure shows that the Time-Spectral method with 17 time levels per period matches well with the BDF computations. The Time-Spectral computation with only 5 time levels per period is noticeably far away from the other computations, but will be valuable for its quick computation time when a coarse estimation is needed. Unlike previous comparisons between the methods, the Time-Spectral Method and BDF do not seem to be approaching the same solution for the turbulent plunging airfoil. This problem is most likely due to the checkerboarding discussed above.

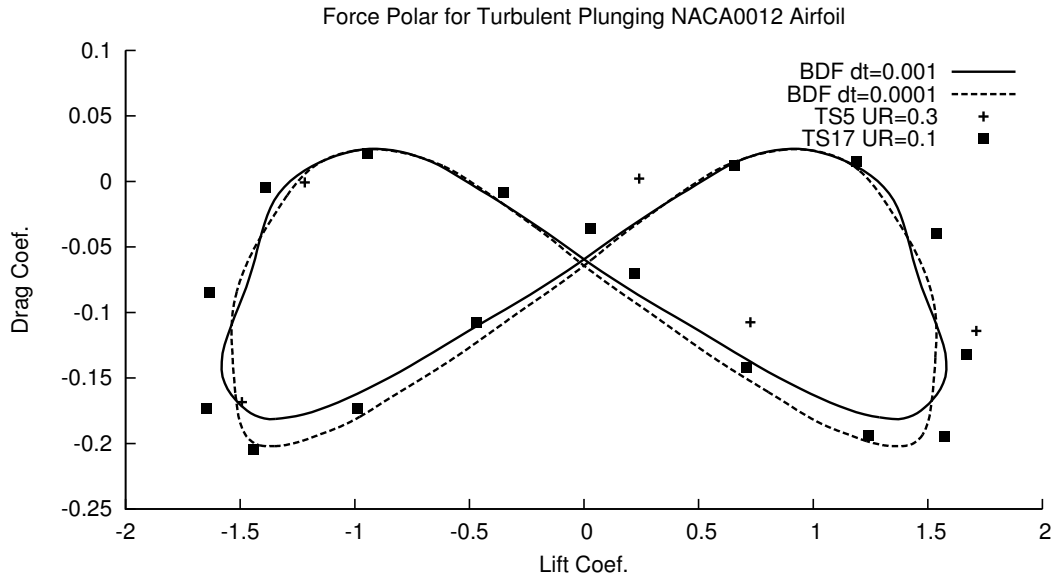


Figure 5.5: Lift versus Drag Polar for Turbulent Plunging Foil

## 5.4 Propeller in Shear Flow

The propeller in shear flow represents a challenging and important engineering problem. Propellers are continually operating in the wake of a ship which has strong changes in inflow conditions at different parts of the cycle. This is due to flow being slowed due to the viscous wake of a ship, with a comparatively much faster free stream flow away from the ship. To demonstrate the capabilities of the Time-Spectral Method, a propeller is simulated in a linear shear inflow. The propeller geometry is that of the OpenFOAM<sup>®</sup> propeller tutorial case. This geometry is chosen as it is readily available and has been tested using OpenFOAM<sup>®</sup>'s turbulent incompressible flow solver. This propeller is a four bladed design. Table 5.2 displays the parameters

of the propeller tested in this test case.  $D$  is the propeller diameter.  $s$  is the parameter to describe the amount of shear in the flow.  $n$  is the rotation rate in rotations per second.  $J$  is the advance coefficient. The flow is in the negative  $y$  direction, as such  $U_\infty$  is negative. Figure 5.6 shows the inflow flow field around the propeller to demonstrate the shear flow. The mesh had 95,000 cells and is generated using an automatic hexahedral mesh generator. The mesh on the propeller can be seen in Figure 5.6. The  $y^+$  on the propeller has an average of 250 and max of 1000. This indicates that most of the propeller is well modeled by wall functions, but there are cells that are outside of the applicable wall region. The outer domain is a cylinder sized  $3.08D$ .

Table 5.2: Parameters for the Propeller in Shear Flow

<i>Inlet :</i>	$\vec{u} = U_\infty \hat{j} + (z - z_0)(s/D)\hat{j}$	$\partial p / \partial \hat{n} = 0$	$\nu_t / \nu = 0.3$
<i>Outlet :</i>	$\partial \vec{u} / \partial \hat{n} = 0$	$p = 0$	$\nu_t / \nu = 0.3$
<i>Outer Boundary :</i>	$\vec{u} = U_\infty \hat{j} + (z - z_0)(s/D)\hat{j}$	$\partial p / \partial \hat{n} = 0$	$\nu_t / \nu = 0.3$
<i>Shaft and Propeller :</i>	$\vec{u} = \vec{U}_{Grid} = \vec{\omega} \times (\vec{r} - \vec{r}_0)$	$\partial p / \partial \hat{n} = 0$	<i>Wall Function</i>
	$Re = U_\infty D / \nu = 9.7 \times 10^5$	$s / U_\infty = -0.584$	$J = U_\infty / (nD) = 1.03$

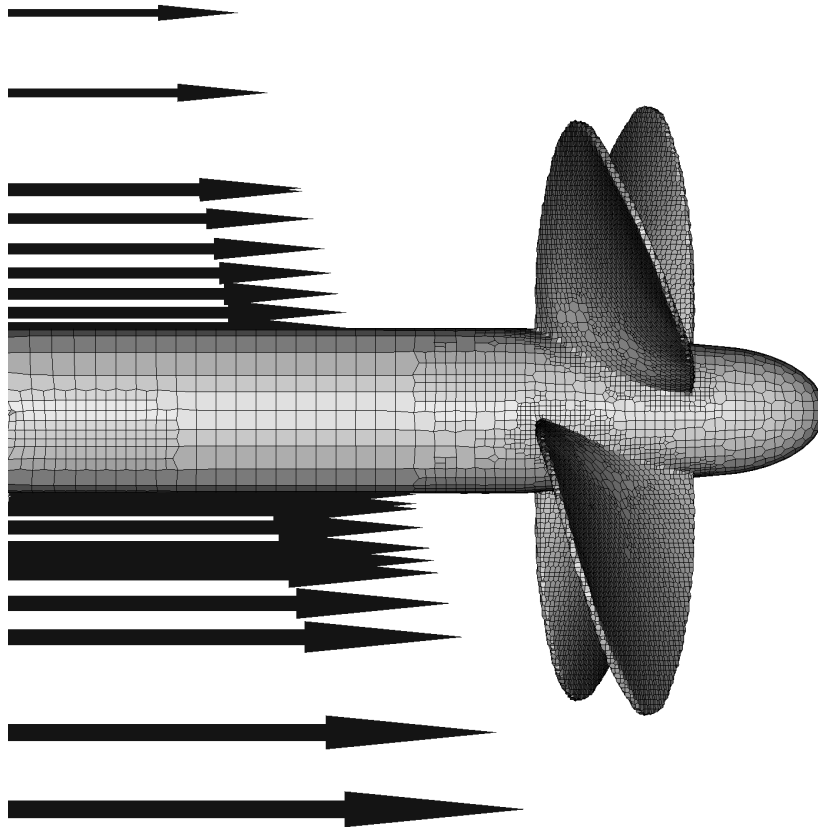


Figure 5.6: Shear Flow at Inlet

Simulations are performed over three levels of time discretizations for each method. For BDF, 1000, 2000, and 4000 time levels per period are used. These are denoted in Figs 5.7, 5.8, and 5.10 as BDF1000, BDF2000, and BDF4000 respectively. The Courant number corresponding to BDF1000 is just under 0.9 for most of the simulation. For the Time-Spectral Method, 5, 9, and 25 time levels per period are used. These are denoted as TS5, TS9, and TS25 respectively. All the cases using the Time-Spectral Method utilize an under-relaxation factor of 0.3. This ensured sufficient stability without overly slowing down convergence. As shown in Section 2.3, lowering the viscosity of a given flow means that a lower under-relaxation factor must be used to stabilize the flow. For that reason, under-relaxation factors tend to be lower for turbulent flows.

Figure 5.7 shows the force on a single blade over two cycles. All methods agree well. Despite the differences in force history being small, it can be seen that they are due to both phase shift and amplitude variation. This indicates that it is unlikely either error mode is dominating. BDF with 4000 time levels per period has some low-amplitude high frequency content that is visible at the lowest forces. This high frequency content is not present in the other simulations. Also, the force on this blade is always in the positive y direction, but has maximum force 3.5 times the minimum force. This shows a substantial change in loading over each cycle. Figure 5.8 shows the  $K_T$  vs  $K_Q$  comparison of the discretizations over one cycle. All simulations are plotted at their periodic steady state solution. All of the discretizations show a nearly linear relationship between  $K_T$  and  $K_Q$  which suggests that the flow is close to static airfoil behaviour as the angle of attack changes due to the shear flow. Also, all of the simulations lie nearly on top of each other. Figure 5.9 shows the pressure on the propeller for both time discretization methods. Both the Time-Spectral Method and BDF show nearly identical pressure distributions. It should be noted that both of these simulations show a pressure checkerboarding on the propeller. This effect is due to a lack of boundary layer prisms which would mitigate this issue. Thus an improved mesh will fix the pressure checkerboarding.

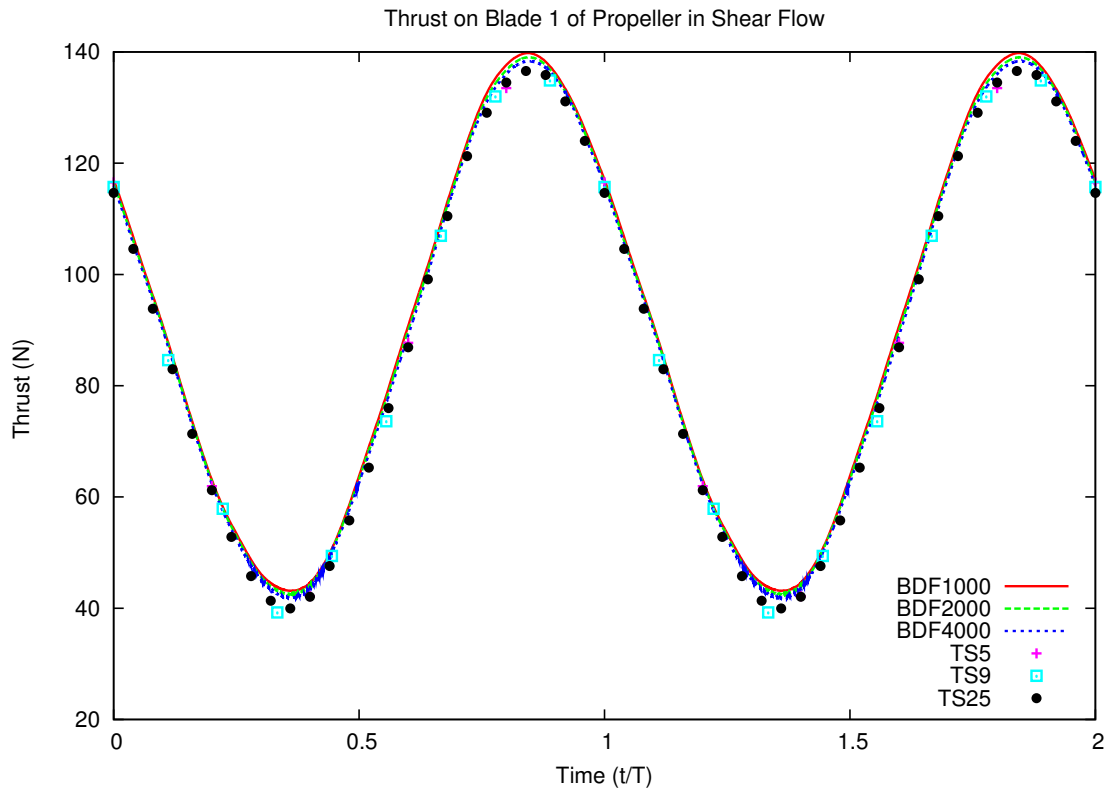


Figure 5.7: Force over a Cycle for Propeller Blade in Shear Flow

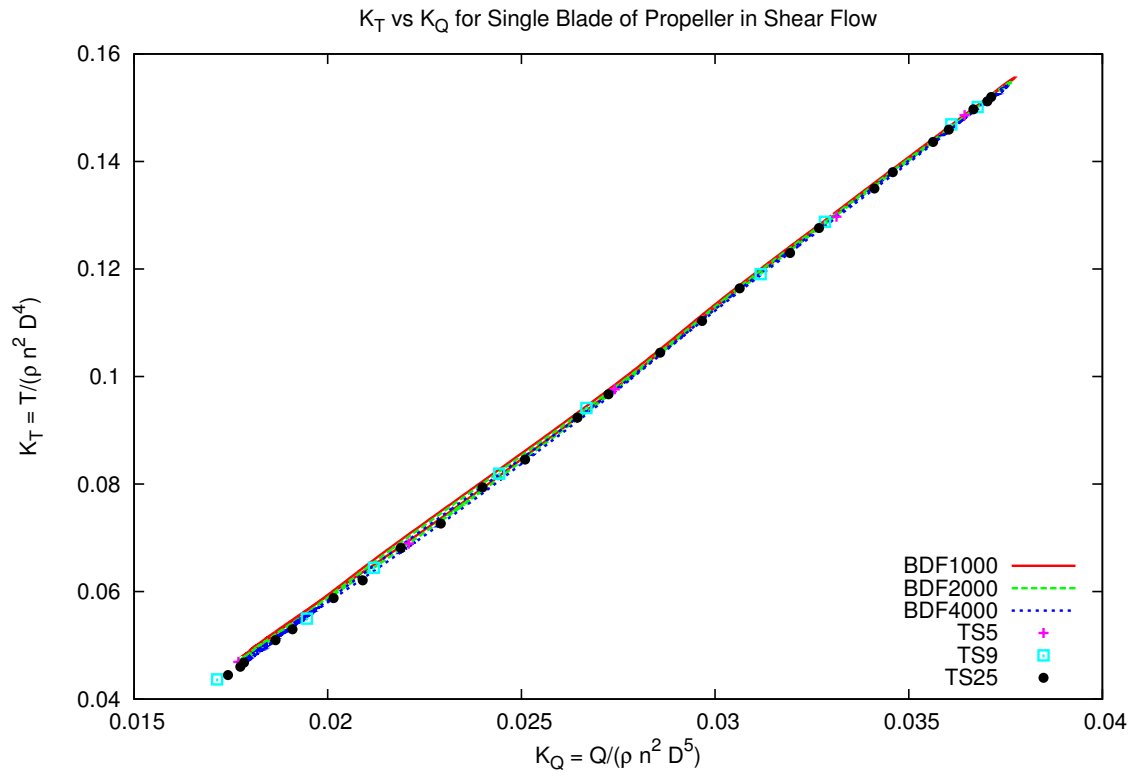
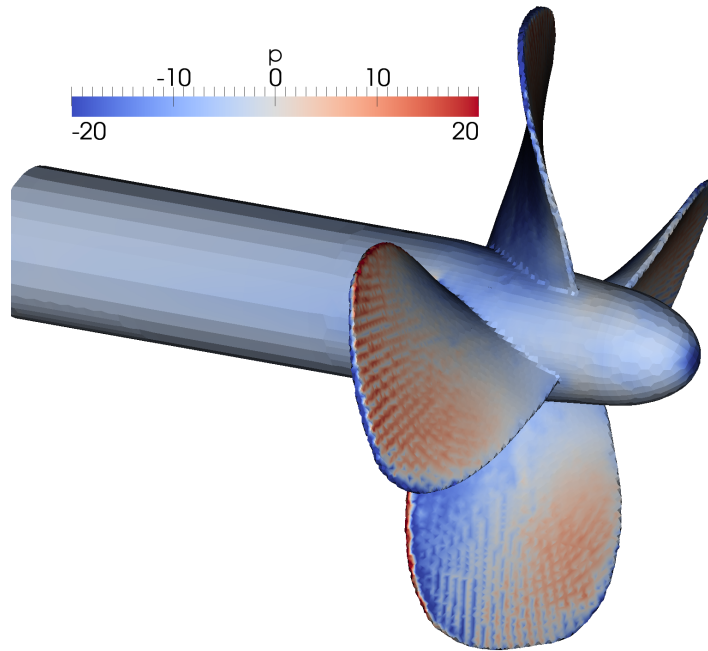
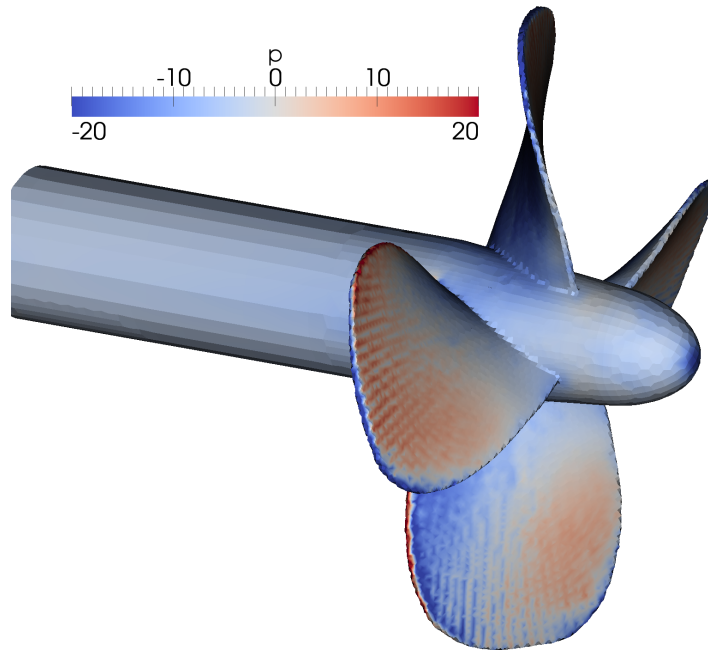


Figure 5.8:  $K_T$  versus  $K_Q$  for Propeller Blade in Shear Flow



(a) Time-Spectral with 9 Time Levels per Period



(b) BDF with 2000 Time Levels per Period

Figure 5.9: Pressure On Propeller Surface at Time  $\sin \omega t = 0$



Figure 5.10 shows the computational convergence history of the force on one blade of the propeller in shear flow. Both methods show good convergence with all time discretizations that were tested. Also, both methods show monotonic convergence as more time levels are added. This indicates that the highly resolved solution is most likely somewhere just less than the final force of the Time-Spectral Method with 25 time levels per period as this was the lowest force computed. As far as cost to convergence, using the Time-Spectral with 5 time levels per period was the fastest method, being about twice as fast as BDF with 1000 time levels per period, which was the fastest BDF method tested. To a given level of accuracy, the Time-Spectral Method with 5 time levels per period produces results much closer to the BDF method with 2000 time levels per period. The Time-Spectral Method, therefore, is a little better than three times faster for this this level of accuracy. For higher accuracy, the Time-Spectral Method with 9 time levels per period has better accuracy than BDF with 4000 time levels per period. Comparing these two cases with regards to computational time again puts the Time-Spectral method a little better than three times faster. Figure 5.10 also shows that the two methods obtain values that are within about 2% of each other at a time equivalent to  $\sin \omega t = 0$ . Visonneau et al. (2012) estimated a numerical uncertainty of 4% for a grid with eight times as many cells for a similar problem. Using that as a baseline, these values are well within spatial discretization error. By examining the force history, it can be seen that all of the methods compute a transient reduction in the predicted force early in the computation, then the computation finds an equilibrium. This force reduction occurs when the effect of the shear flow at the inlet first affects the propeller.

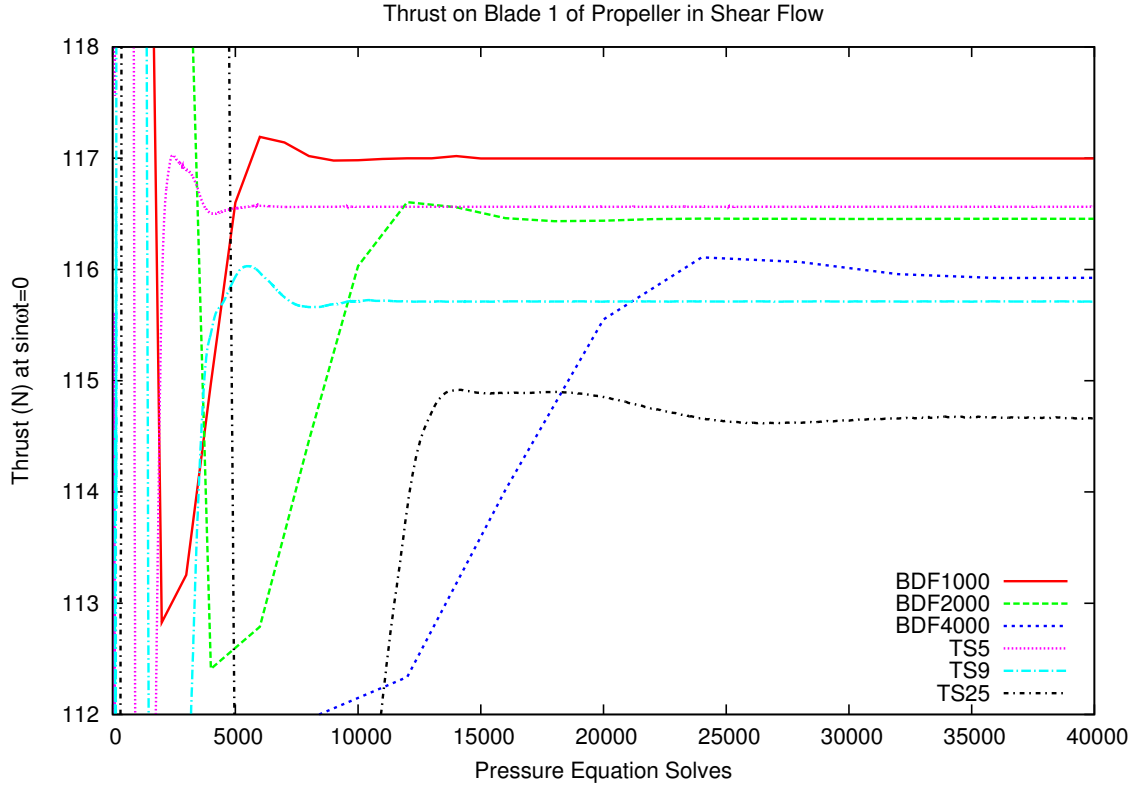


Figure 5.10: Force Convergence Comparison of Propeller at Time  $\sin \omega t = 0$

The Time-Spectral Method is extended for the URANS equations with incompressible flow. The Spalart-Allmaras turbulence closure model was used. This code is demonstrated on a pitching foil, a plunging foil, and a propeller in shear flow. The computational results of the pitching foil show a quick convergence of the Time-Spectral Method. The plunging foil results show that the Time-Spectral Method avoids checkerboarding problems that can appear in BDF for highly resolved simulations. The propeller test demonstrates the applicability of the Time-Spectral Method for three-dimensional simulations with complex geometry. The propeller case shows a four-fold speedup by using the Time-Spectral Method.

## CHAPTER VI

### Conclusions

This thesis reviews the Time-Spectral Method and developed its use for incompressible flow with Computational Fluid Dynamics (CFD). The introduction discusses the motivation of the Time-Spectral Method method and gives a review of its development. The algorithm development section shows the derivation of the discrete Time-Spectral formulation for a general periodic function. This formulation is then put into a fully time-coupled matrix so that the Time-Spectral Method can be used for Partial Differential Equations (PDEs) with separate time and space derivatives. The block-Jacobi algorithm is introduced as a method to solve this fully time-coupled matrix for a PDE solver that only forms a system of equations for one time level at a time. Under-relaxation is introduced as a way to boost diagonal dominance for this fully time-coupled matrix when using the block-Jacobi algorithm and control convergence. The utility of Time-Spectral Methods is demonstrated for simulating a number of problems of interest. These investigations show the impressive accuracy of the Time-Spectral Method and also give a number of practical observations about the method.

## 6.1 Accomplishments

The Time-Spectral Method is shown to be an effective time-discretization scheme for periodic flows. Chapter III demonstrates results of using the Time-Spectral Method with the linear-advection diffusion equation as well as Burgers' Equation. These results indicate that the Time-Spectral Method has much greater accuracy than Backward Difference Formula (BDF), and needs orders-of-magnitude fewer time levels for the same accuracy. For Burgers' Equation, the Time-Spectral Method is shown that it could be over an order of magnitude faster than BDF to achieve a given accuracy. The results from Burgers' Equation also demonstrate the utility of the Time-Spectral Method for solving nonlinear problems. Chapter IV extends the Time-Spectral Method to incompressible flows via a pressure projection method. A pressure projection method with the Time-Spectral Method is demonstrated on backward-facing step, a laminar pitching foil, and a laminar plunging foil. The laminar pitching foil is shown to have a factor of five speedup when using the Time-Spectral Method. The laminar plunging foil is shown to have a factor of 2.5 speedup when using the Time-Spectral Method. Chapter V demonstrates the first use of the Time-Spectral Method to simulate incompressible turbulent flows. The results of turbulent flow simulation over the pitching foil indicate that by using only a very few number of time points, the Time-Spectral Method can resolve flow that would take orders of magnitude more time points with BDF. The Time-Spectral Method is implemented for these governing equations in an open-source CFD Library that is typical of commercial CFD products.

The Time-Spectral Method has a number of interesting numerical properties that distinguish it from the conventional BDF time marching. Firstly, as observed with linear advection, the choice of under-relaxation factor can significantly determine the speed of convergence of the fully time-coupled matrix, where even small changes can have a dramatic effect. For more complicated flows, such as the laminar plunging foil,

however, a wider range of under-relaxation factors show similar computational costs. An analysis of the under-relaxed fully-time coupled matrix is presented in Section 2.3. This analysis gives an estimation of maximum usable under-relaxation value of a simplified governing equation. This analysis shows that, for given schemes, a larger Courant number and larger diffusion coefficient,  $\beta$ , reduce stability requirements on the under-relaxation factor. These two properties are contrary to the stability requirements of most time marching schemes. This knowledge of the Time-Spectral Method will be valuable for future investigations.

## 6.2 Future Work

Due to the vast amount of periodic fluid flow problems, the Time-Spectral Method has many available applications. Many of these will, however, require new capabilities in order to be used. A sliding grid technique, such as that employed by Visonneau et al. (2012), could be used to match propeller and ship geometry. This is an important naval architecture problem for propeller selection and hull forming. A sliding grid would also be very important for coupling a rotor and stator in waterjets and water turbines. These two problems typically have very small distances between rotor and stator which can create strong transients as the rotor passes through the wake of the stator and vice versa. A sliding grid technique could be used to investigate rotor tower interactions in wind turbines. The Time-Spectral Method could be extended with the Gradient Based Variable Time Period method of Gopinath and Jameson (2006) to be used where the time period is not known, such as naturally periodic flows. This would be useful to study vortex shedding behind a cylinder, which is an area of research for underwater pipelines and many underwater support structures. A multi-frequential technique similar to that done by Rauschenberger (2008) could be used to address flows that experience multiple forcing frequencies that are not integer multiples of each other. Flows with forcing at integer multiples of each other

can be addressed with the current method by using more harmonics to represent the flow. This two capabilities could be combined to address flows that are partially naturally forced and partially cyclically forced. Large Eddy Simulations (LES) is used in numerous engineering fields to study turbulent flow. It remains an open research question as to how well the Time-Spectral Method will work with LES. As LES produces very high fidelity flow information, LES would complement the very high accuracy Time-Spectral Method should they be able to be implemented together.

Besides extending to new problems, there is an abundance of work available for understanding the Time-Spectral Method in conjunction with the Navier-Stokes equations. While linear stability could be easily evaluated by examining the fully time-coupled matrix, nonlinear effects have introduced unusual oscillations that are not fully understood. A thorough understanding of nonlinear stability would provide valuable insight to what problems are most suitable for the Time-Spectral Method.

The use of Time-Spectral Methods for free surface flows represents a very promising, yet very challenging endeavor. The periodic motion of waves is a very challenging research field that has many application areas. Firstly, wave power generation is a procedure that extracts energy from free surface waves. Free surface waves also affect ship seakeeping motions. By using the Time-Spectral Method, ship motions in waves can be studied so as to be mitigated. This is especially important for the roll degree of freedom, which has strong viscous effects that are poorly accounted for by potential flow methods. The challenge of simulating free-surface flows with the Time-Spectral Method arises from advection of the free-surface variable. The Volume Of Fluid (VOF) approach is, in principle, a pure advection problem. The boundedness limitations on the Volume Of Fluid variable, however, make it difficult to advect with the Time-Spectral Method. This is related to Gibb's phenomena where using Fourier Basis functions to represent a discontinuity, such as that between air and water, nonmonotonic values are introduced. A level-set method, on the other hand, has

no boundedness limitation and no sharp discontinuity between air and water. The difficulty with using a level-set method, however, is that mass is not automatically conserved, so reinitialization functionality needs to be implemented. Clearly both methods have advantages and disadvantages. Another complication of simulating free-surface waves is that waves reflect off the computational boundary. This keeps energy in the domain that would propagate away in an infinite domain. This energy builds up and causes long computation time instabilities that cause the computation to diverge. While this is also a problem with standard time-marching techniques, this issue will likely be more limiting to the Time-Spectral Method as these boundary-based instabilities may preclude the existence of a “periodic steady state” even though such a state physically exists. Using the Time-Spectral Method for free-surface flows, therefore, has many complications, but also has many engineering applications.

## APPENDICES



## APPENDIX A

### Power-Sum Simplification

#### A.1 Power-Sum Simplification

Starting with the equation:

$$Du^n = \frac{2\pi}{T} \sum_{m=0}^{N-1} u^m \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\frac{2\pi}{N}(n-m)} \quad (\text{A.1})$$

it is possible to simplify the second summation. First, this equation is rewritten as:

$$\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\frac{2\pi}{N}(n-m)} = \frac{i}{N} \sum_{k=-\frac{N-1}{2}}^{-1} k e^{ik\phi} + \frac{i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{ik\phi} \quad (\text{A.2a})$$

for

$$\phi = \frac{2\pi}{N} (n - m) = \frac{2\pi j}{N} \quad (\text{A.2b})$$

Notice that Equation (A.2a) loses the  $k = 0$  term since the leading  $k$  multiplicand causes the  $k = 0$  term to drop out.

The signs are changed in order to get the negative summation into a standard

form, resulting in:

$$\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\frac{2\pi}{N}(n-m)} = \frac{-i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{-ik\phi} + \frac{i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{ik\phi} \quad (\text{A.3})$$

These individual summations can now be simplified by putting them into the following power summation's form:

$$\sum_{h=1}^n h x^h = \frac{x - (n+1)x^{n+1} + nx^{n+2}}{(x-1)^2} \quad (\text{A.4})$$

Substituting,

$$n = \frac{N-1}{2} \quad (\text{A.5a})$$

$$x = e^{\pm ik\phi} \quad (\text{A.5b})$$

gives:

$$\frac{i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{ik\phi} = \frac{i}{N} \frac{e^{i\phi} - \frac{N+1}{2} e^{i\phi\frac{N+1}{2}} + \frac{N-1}{2} e^{i\phi\frac{N+3}{2}}}{e^{2i\phi} - 2e^{i\phi} + 1} \quad (\text{A.6a})$$

$$\frac{-i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{-ik\phi} = -\frac{i}{N} \frac{e^{-i\phi} - \frac{N+1}{2} e^{-i\phi\frac{N+1}{2}} + \frac{N-1}{2} e^{-i\phi\frac{N+3}{2}}}{e^{-2i\phi} - 2e^{-i\phi} + 1} \quad (\text{A.6b})$$

Which is then multiplied by a factor of  $1 = \left[ \frac{e^{\pm i\phi}}{e^{\pm i\phi}} \right]$

$$\frac{i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{ik\phi} \left[ \frac{e^{-i\phi}}{e^{-i\phi}} \right] = \frac{i}{N} \frac{1 - \frac{N+1}{2} e^{i\phi\frac{N-1}{2}} + \frac{N-1}{2} e^{i\phi\frac{N+1}{2}}}{e^{i\phi} - 2 + e^{-i\phi}} \quad (\text{A.7a})$$

$$\frac{-i}{N} \sum_{k=1}^{\frac{N-1}{2}} k e^{-ik\phi} \left[ \frac{e^{+i\phi}}{e^{+i\phi}} \right] = +\frac{i}{N} \frac{-1 + \frac{N+1}{2} e^{-i\phi\frac{N-1}{2}} - \frac{N-1}{2} e^{-i\phi\frac{N+1}{2}}}{e^{-i\phi} - 2 + e^{i\phi}} \quad (\text{A.7b})$$

These equations can now be added since they have the same denominator, resulting

in:

$$\begin{aligned}
\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\phi} &= \frac{i}{N} \frac{1}{e^{i\phi} - 2 + e^{-i\phi}} \times \\
&\left[ 1 - 1 + \frac{N+1}{2} \left( e^{-i\phi \frac{N-1}{2}} - e^{i\phi \frac{N-1}{2}} \right) + \frac{N-1}{2} \left( e^{i\phi \frac{N+1}{2}} - e^{-i\phi \frac{N+1}{2}} \right) \right] \\
&= \frac{i}{N} \frac{1}{2 \cos \phi - 2} \times \\
&\left[ \frac{N+1}{2} \left( -2i \sin \left( \phi \frac{N-1}{2} \right) \right) + \frac{N-1}{2} \left( 2i \sin \left( \phi \frac{N+1}{2} \right) \right) \right] \\
&= \frac{-1}{N} \frac{1}{\cos \phi - 1} \times \\
&\left[ -\frac{N+1}{2} \left( \sin \left( \frac{\phi N}{2} \right) \cos \left( \frac{\phi}{2} \right) - \cos \left( \frac{\phi N}{2} \right) \sin \left( \frac{\phi}{2} \right) \right) \right. \\
&\quad \left. + \frac{N-1}{2} \left( \sin \left( \frac{\phi N}{2} \right) \cos \left( \frac{\phi}{2} \right) + \cos \left( \frac{\phi N}{2} \right) \sin \left( \frac{\phi}{2} \right) \right) \right]
\end{aligned} \tag{A.8}$$

Which comes from:

$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y$$

Notice that the imaginary term has disappeared. This result is expected because the input and output should both be real numbers.

Remembering that  $\phi = \frac{2\pi j}{N}$ , we continue the simplification of time-spectral terms:

$$\begin{aligned}
\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\phi} &= \frac{-1}{N} \frac{1}{\cos \phi - 1} \times \\
&\left[ -\frac{N+1}{2} \left( \sin(\pi j) \cos \left( \frac{\phi}{2} \right) - \cos(\pi j) \sin \left( \frac{\phi}{2} \right) \right) \right. \\
&\quad \left. + \frac{N-1}{2} \left( \sin(\pi j) \cos \left( \frac{\phi}{2} \right) + \cos(\pi j) \sin \left( \frac{\phi}{2} \right) \right) \right]
\end{aligned} \tag{A.9}$$

Now, we need to look closely at  $\sin(\pi j)$  and  $\cos(\pi j)$ . Because  $j$  is an integer by

definition,  $\sin(\pi j) = 0$  and  $\cos(\pi j) = (-1)^j = (-1)^{n-m}$ . Therefore,

$$\begin{aligned} \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\phi} &= \frac{-1}{N} \frac{1}{\cos \phi - 1} \left[ \frac{N+1}{2} \left( (-1)^{n-m} \sin\left(\frac{\phi}{2}\right) \right) + \frac{N-1}{2} \left( (-1)^{n-m} \sin\left(\frac{\phi}{2}\right) \right) \right] \\ &= \frac{-1}{N} \frac{1}{\cos \phi - 1} \left[ N \left( (-1)^{n-m} \sin\left(\frac{\phi}{2}\right) \right) \right] \end{aligned} \quad (\text{A.10})$$

Utilizing the double angle formula  $\cos(2x) = 1 - 2\sin^2 x$ ,

$$\begin{aligned} \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\phi} &= \frac{-1}{N} \frac{1}{1 - 2\sin^2\left(\frac{\phi}{2}\right) - 1} \left[ N \left( (-1)^{n-m} \sin\left(\frac{\phi}{2}\right) \right) \right] \\ &= \frac{(-1)^{n-m} \sin\left(\frac{\phi}{2}\right)}{2\sin^2\left(\frac{\phi}{2}\right)} \end{aligned} \quad (\text{A.11})$$

This equation is simplified and  $2\pi(n-m)/N$  is substituted for  $\phi$ . This gives the final result:

$$\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \frac{ik}{N} e^{ik\phi} = \frac{(-1)^{n-m}}{2} \csc\left(\frac{2\pi(n-m)}{2N}\right) \quad (\text{A.12})$$

Gopinath and Jameson (2006) achieved the same result using the series summation:

$$\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{ik\phi} = \frac{\sin\left(\frac{N}{2}\phi\right)}{\sin\left(\frac{\phi}{2}\right)} \quad (\text{A.13})$$

and taking the derivative. This result gets put back into the original equations to yield the time-spectral formulation for all time levels  $m \neq n$  :

$$Du^n = \frac{2\pi}{T} \sum_{m=0}^{N-1} u^m \frac{(-1)^{n-m}}{2} \csc\left(\frac{2\pi(n-m)}{2N}\right) \quad (\text{A.14})$$

## APPENDIX B

### Von Neumann Stability Analysis

#### B.1 Von Neumann Analysis

Presented here is Von Neumann stability analysis of the Time-Spectral method. Hirsch (2007) presents the procedure of Von Neumann analysis. Von Neumann analysis uses a Fourier analysis to determine stability of a scheme by looking at a local amplification factor. This amplification factor is representative of whether or not a small disturbance will grow. As Fourier analysis is being used, the Von Neumann assumes periodic boundary conditions. This state is mostly representative of the interior of a domain where boundary conditions have little influence. As with most PDE analysis, the linear-advection diffusion equation will be used as the model problem.

$$D_t\phi + c\frac{\partial\phi}{\partial x} - \nu\frac{\partial^2\phi}{\partial x^2} = 0 \tag{B.1}$$

The first scheme to be tested is linear space discretization with  $\nu = 0$ . This

becomes the following equations:

$$d_1 \phi_i^{n+1} - d_1 \phi_i^{n-1} + c \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x} = 0 \quad (\text{B.2})$$

$$\phi_i^{n+1} - \phi_i^{n-1} + \frac{c}{2d_1 \Delta x} (\phi_{i+1}^n - \phi_{i-1}^n) = 0 \quad (\text{B.3})$$

These equations utilize  $d_1$  from Equation 2.1.7 with three time levels per period. By substituting  $\phi_{i+m}^{n+k} = V^{n+k} e^{I(i+m)\psi}$ , the solution is represented by Fourier components. In this substitution  $I = \sqrt{-1}$ . Making this substitution as well as  $c/(2d_1 \Delta x) = \xi$  for simplification results in the equation:

$$V^{n+1} e^{I(i)\psi} - V^{n-1} e^{I(i)\psi} + \xi (V^n e^{I(i+1)\psi} - V^n e^{I(i-1)\psi}) = 0 \quad (\text{B.4})$$

The intermediate goal of Von Neumann analysis is to find  $G = V^{n+1}/V^n$ .  $G$  is the amplification factor.  $G$  should satisfy  $|G| \leq 1$  for stability. Solving for  $G$ :

$$e^{Ii\psi} \frac{V^{n+1} - V^{n-1}}{V^n} + \xi e^{Ii\psi} (e^{I\psi} - e^{-I\psi}) = 0 \quad (\text{B.5})$$

$$\frac{V^{n+1}}{V^n} - \frac{V^{n-1}}{V^n} + \xi (e^{I\psi} - e^{-I\psi}) = 0 \quad (\text{B.6})$$

$$\frac{V^{n+1}}{V^n} - \frac{V^{n-1}}{V^n} + \xi 2I \sin \psi = 0 \quad (\text{B.7})$$

$$G - \frac{1}{G} + \xi 2I \sin \psi = 0 \quad (\text{B.8})$$

$$G = -\xi I \sin \psi \pm \frac{1}{2} \sqrt{4\xi^2 I^2 \sin^2 \psi + 4} \quad (\text{B.9})$$

$$G = -\xi I \sin \psi \pm \sqrt{-\xi^2 \sin^2 \psi + 1} \quad (\text{B.10})$$

$G$  must be examined for several cases to understand what regions are stable. Starting

with  $\xi^2 \leq 1$ :

$$G = -\xi I \sin \psi \pm \sqrt{|1 - \xi^2 \sin^2 \psi|} \quad (\text{B.11})$$

$$|G|^2 = |1 - \xi^2 \sin^2 \psi| + |\xi^2 \sin^2 \psi| \quad (\text{B.12})$$

$$|G|^2 = 1 \rightarrow A - \text{Stable} \quad (\text{B.13})$$

Now look at  $\xi^2 > 1$ , the worst case is when  $\xi^2 \sin^2 \psi > 1$ , which will happen for some  $\psi$ :

$$G = -\xi I \sin \psi \pm I \sqrt{|\xi^2 \sin^2 \psi - 1|} \quad (\text{B.14})$$

$$|G| = |\xi \sin \psi| \mp \sqrt{|\xi^2 \sin^2 \psi - 1|} \quad (\text{B.15})$$

If  $|G|$  is unstable in either root, the case is unstable. In the worst case both parts add, resulting in:

$$|G| = |\xi \sin \psi| + \sqrt{|\xi^2 \sin^2 \psi - 1|} \quad (\text{B.16})$$

Again, at some point  $|\xi \sin \psi| > 1$  so  $|G| > 1$ . Notice that this only happens when  $\sin \psi \rightarrow 1$ .  $\psi$  represents the wavenumber, with  $\psi = \pi$  corresponding to short wavelengths and  $\psi = 0$  corresponding to long wavelengths. As the instability in question is for a moderate value of  $\psi$ , this instability will most likely destabilize all solutions. That is, the algorithm is unstable for all  $|\xi| = \left| \frac{c}{2d_1 \Delta x} \right| > 1$  and only A-Stable for  $\xi^2 \leq 1$ .

A more common scheme is upwind spatial discretization. The upwind scheme is only first order accurate, and highly diffusive. Despite these limitations, it is very stable with most time-marching schemes. The upwind scheme only uses the current space point and the space point that is “upwind.”

$$\phi_i^{n+1} - \phi_i^{n-1} + \frac{c}{d_1 \Delta x} (\phi_i - \phi_{i-1}) = 0 \quad (\text{B.17})$$

Replace  $\phi_{i+m}^{n+k} = V^{n+k} e^{I(i+m)\psi}$ , where  $I = \sqrt{-1}$  and  $\frac{c}{2d_1 \Delta x} = \xi$ . Note that  $c > 0$  for the upwind scheme.

$$V^{n+1} e^{Ii\psi} - V^{n-1} e^{Ii\psi} + 2\xi (V^n e^{Ii\psi} - V^n e^{I(i-1)\psi}) = 0 \quad (\text{B.18})$$

$$\frac{e^{Ii\psi}}{V^n} (V^{n+1} - V^{n-1}) + 2\xi (e^{Ii\psi} - e^{I(i-1)\psi}) = 0 \quad (\text{B.19})$$

$$\frac{1}{V^n} (V^{n+1} - V^{n-1}) + 2\xi (1 - e^{-I\psi}) = 0 \quad (\text{B.20})$$

$$G - \frac{1}{G} + 2\xi (1 - e^{-I\psi}) = 0 \quad (\text{B.21})$$

$$G = -\xi (1 - e^{-I\psi}) \pm \sqrt{\xi^2 (1 - e^{-I\psi})^2 + 1} \quad (\text{B.22})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (1 - 2e^{-I\psi} + e^{-2I\psi}) + 1} \quad (\text{B.23})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (1 - 2[\cos \psi - I \sin \psi] + [\cos 2\psi - I \sin 2\psi]) + 1} \quad (\text{B.24})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (1 - 2 \cos \psi + 2I \sin \psi + \cos 2\psi - 2I \sin \psi \cos \psi) + 1} \quad (\text{B.25})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (1 - 2 \cos \psi + 2I \sin \psi [1 - \cos \psi] + 2 \cos^2 \psi - 1) + 1} \quad (\text{B.26})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (-2 \cos \psi + 2I \sin \psi [1 - \cos \psi] + 2 \cos^2 \psi) + 1} \quad (\text{B.27})$$

$$G = \xi (\cos \psi - I \sin \psi - 1) \pm \sqrt{\xi^2 (2 \cos \psi [\cos \psi - 1] + 2I \sin \psi [1 - \cos \psi]) + 1} \quad (\text{B.28})$$

Lets look at several cases. First the short wave case.

$$G(\psi = 0) = 0 \pm \sqrt{0 + 1} = \pm 1 \rightarrow |G(\psi = 0)| = |\pm 1| \quad (\text{B.29})$$



So in all cases with upwind, this scheme is A-Stable as  $\psi \rightarrow 0$ . Now for the long wave case.

$$G(\psi = \pi) = -2\xi \pm \sqrt{4\xi^2 + 1} \quad (\text{B.30})$$

The worst case is subtraction for  $\xi \geq 0$ . Use  $\Theta = \sqrt{4\xi^2 + 1}$ . Notice that  $\Theta \geq 1$  due to  $\xi$  being real.

$$|G(\psi = \pi)| = |-2\xi - \Theta| \geq 1 \quad (\text{B.31})$$

$$|G(\psi = \pi)| > 1 \rightarrow \textit{Unconditionally Unstable} \quad (\text{B.32})$$

So Von Neumann predicts that the Time-Spectral Method is unstable with upwind advection. This instability occurs as  $\psi \rightarrow \pi$  which is a long wavelength instability.

Now the diffusion term will be investigated. Again the equation under consideration is the linear advection-diffusion equation given in Equation B.1. For diffusion analysis,  $c = 0$  and the equation will be discretized using a linear diffusion scheme.

$$d_1\phi_i^{n+1} - d_1\phi_i^{n-1} - \frac{\nu}{\Delta x^2} (\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n) = 0 \quad (\text{B.33})$$

Define  $\kappa = \frac{\nu}{d_1\Delta x^2}$

$$\phi_i^{n+1} - \phi_i^{n-1} - \kappa (\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n) = 0 \quad (\text{B.34})$$

Replace  $\phi_{i+m}^{n+k} = V^{n+k} e^{I(i+m)\psi}$ , where  $I = \sqrt{-1}$

$$V^{n+1} e^{Ii\psi} - V^{n-1} e^{Ii\psi} - \kappa (V^n e^{I(i+1)\psi} - 2V^n e^{Ii\psi} + V^n e^{I(i-1)\psi}) = 0 \quad (\text{B.35})$$

$$\frac{e^{Ii\psi}}{V^n} (V^{n+1} - V^{n-1}) - \kappa e^{Ii\psi} (e^{I\psi} - 2 + e^{-I\psi}) = 0 \quad (\text{B.36})$$

$$G + \frac{1}{G} - \kappa (-2 + 2 \cos \psi) = 0 \quad (\text{B.37})$$

$$G + \frac{1}{G} - 2\kappa (\cos \psi - 1) = 0 \quad (\text{B.38})$$

$$G = \kappa (\cos \psi - 1) \pm \sqrt{\kappa^2 (\cos \psi - 1)^2 - 1} \quad (\text{B.39})$$

For  $\kappa^2 (\cos \psi - 1)^2 < 1$ , which requires  $\kappa^2 < \frac{1}{4}$ , the following amplification factor results:

$$G = \kappa (\cos \psi - 1) \pm I \sqrt{|1 - \kappa^2 (\cos \psi - 1)^2|} \quad (\text{B.40})$$

$$|G|^2 = \kappa^2 (\cos \psi - 1)^2 + |1 - \kappa^2 (\cos \psi - 1)^2| \quad (\text{B.41})$$

$$|G|^2 = 1 \rightarrow |G| = 1 \rightarrow A - \text{Stable} \quad (\text{B.42})$$

When  $\chi = \kappa^2 (\cos \psi - 1)^2 > 1$

$$G = \sqrt{\chi} \pm \sqrt{\chi^2 - 1} \quad (\text{B.43})$$

$$\sqrt{\chi^2 - 1} > 0 \quad (\text{B.44})$$

$$|\sqrt{\chi}| > 1 \quad (\text{B.45})$$

Both roots of  $G$  must satisfy  $|G| \leq 1$  for stability. The worst case for  $\chi = \kappa^2 (\cos \psi - 1)^2 > 1$  is addition resulting in

$$G = \sqrt{\chi} + \sqrt{\chi^2 - 1} > 1 \quad (\text{B.46})$$

$$G > 1 \quad (\text{B.47})$$

Resulting in  $G$  being unconditionally unstable for  $\left(\frac{\nu}{d_1 \Delta x^2}\right)^2 (\cos \psi - 1)^2 > 1$ . Notice again that the worst problems for this instability are at the long wave length, although this wave length is strongly dependent on  $\kappa$ .

For pure linear-advection, the linear spatial discretization is A-Stable with a three level time-spectral scheme as long as  $\frac{c}{2d_1 \Delta x} < 1$  and unconditionally unstable otherwise. For pure linear-advection with upwind spatial discretization, a three level time-spectral scheme is unconditionally unstable. For linear-diffusion with linear scheme, a three level time-spectral scheme is A-Stable as long as  $\kappa^2 = \left[\frac{\nu}{d_1 \Delta x^2}\right]^2 < \frac{1}{4}$ . However, notice that the problems are for  $\psi \rightarrow \pi$  which is a long wavelength instability rather than the more typical short wavelength instabilities. This indicates that the boundary conditions are coming into play as noticed by Hall et al. (2002). As such, Von Neumann analysis shows instability that is actually stabilized by boundary conditions. The Von Neumann analysis does indicate, however, that there will likely be problems with large domains and that small domains are preferable for using the Time-Spectral Method. This same long wavelength instability is also what makes both linear-advection schemes unstable.

The current analysis also does not include the stabilizing effect of under-relaxation. This indicates that another stability method may be more appropriate for analyzing time-spectral schemes as we have implemented them.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- Antheaume, S. and C. Corre (2011, April). Implicit Time Spectral Method for Periodic Incompressible Flows. *AIAA Journal* 49(4), 791–805.
- Ashford, G. A. (1996). *An unstructured grid generation and adaptive solution technique for high-Reynolds-number compressible flows*. Ph. D. thesis, University of Michigan.
- Butsuntorn, N. and A. Jameson (2008). Time Spectral Method for Rotorcraft Flow. *AIAA 46th Aerospace Sciences Meeting & Exhibit, Reno, NV*.
- Canuto, C. G., M. Y. Hussaini, A. Quarteroni, and T. A. Zang (2007, July). *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics (Scientific Computation)* (1 ed.). Springer.
- Ekici, K. and K. Hall (2007). Nonlinear analysis of unsteady flows in multistage turbomachines using harmonic balance. *AIAA Journal* 45(5), 1047–1057.
- Ekici, K., K. Hall, and E. Dowell (2008). Computationally fast harmonic balance methods for unsteady aerodynamic predictions of helicopter rotors. *Journal of Computational Physics* 227, 6206–6225.
- Ferziger, J. H. and M. Peric (2002). *Computational Methods for Fluid Dynamics* (Third ed.). Springer.
- Gopinath, A. (2007, April). *Efficient Fourier-Based Algorithms For Time-Periodic Unsteady Problems*. Ph. D. thesis, Stanford University.
- Gopinath, A. and A. Jameson (2005). Time spectral method for periodic unsteady computations over two-and three-dimensional bodies. *AIAA 2005-1220*.
- Gopinath, A. and A. Jameson (2006). Application of the time spectral method to periodic unsteady vortex shedding. *AIAA 2006-0449*.
- Gopinath, A., E. van der Weide, J. Alonso, A. Jameson, K. Ekici, and K. Hall (2007). Three-dimensional unsteady multi-stage turbomachinery simulations using the harmonic balance technique. *AIAA 2007-0892*, 2007.
- Hall, K. and E. Crawley (1989, June). Calculation of unsteady flows in turbomachinery using the linearized Euler equations. *AIAA Journal* 27(6), 777–787.

- Hall, K., J. Thomas, and W. Clark (2002, May). Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. *AIAA Journal* 40(5), 879–886.
- Hirsch, C. (2007). *Numerical Computation of Internal and External Flows* (Second ed.). Butterworth-Heinemann.
- Issa, R. I. (1985). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* 62(1), 40–65.
- Jackson, A., M. Campobasso, and M. Baba-Ahmadi (2011). On the Parallelization of a Harmonic Balance Compressible Navier-Stokes Solver for Wind Turbine Aerodynamics. *ASME paper GT2011-45306*.
- Jameson, A. (2009, October). An Assessment of Dual-Time Stepping, Time Spectral and Artificial Compressibility based Numerical Algorithms for Unsteady Flow with Applications to Flapping Wings. *19th AIAA Computational Fluid Dynamics*, 1–21.
- Jasak, H. (1996, June). *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Ph. D. thesis, Imperial College of London.
- Jasak, H. (2009). Dynamic Mesh Handling in OpenFOAM. In *AIAA 2009-0341*.
- Jasak, H., H. Weller, and A. Gosman (1999). High resolution NVD differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids* 31(2), 431–449.
- Khoo, B. C., J. White, J. Peraire, and A. Patera (2003, April). 16.920J Numerical Methods for Partial Differential Equations (SMA 5212). In *Massachusetts Institute of Technology: MIT OpenCourseWare, <http://ocw.mit.edu>*, pp. 1–38.
- Ko, S. and W. McCroskey (1997). Computations of unsteady separating flows over an oscillating airfoil. *AIAA Journal* 35(7), 1235–1238.
- Kumar, M. and V. Murthy (2007). Analysis of Flow Around Multibladed Rotor Using CFD in the Frequency Domain. In *25th AIAA Applied Aerodynamics Conference, AIAA 2007-3608*, Miami, FL, pp. 1–17.
- Lian, Y. and W. Shyy (2007, July). Laminar-Turbulent Transition of a Low Reynolds Number Rigid or Flexible Airfoil. *AIAA Journal* 45(7), 1501–1513.
- Mavriplis, D. J. and Z. Yang (2011, May). Time Spectral Method for Periodic and Quasi-Periodic Unsteady Computations on Unstructured Meshes. *Mathematical Modelling of Natural Phenomena* 6(3), 213–236.
- Mavriplis, D. J., Z. Yang, and N. Mundis (2012). Extensions of Time Spectral Methods for Practical Rotorcraft Problems. In *50th AIAA Aerospace Sciences Meeting*, Nashville, TN.

- McMullen, M., A. Jameson, and J. Alonso (2002). Application of a non-linear frequency domain solver to the Euler and Navier-Stokes Equations. *AIAA 2002-0120*.
- Milgram, J. H. (1998). Fluid mechanics for sailing vessel design. *Annual Review of Fluid Mechanics* 30(1), 613–653.
- Nadarajah, S., M. McMullen, and A. Jameson (2003). Optimum shape design for unsteady flows using time accurate and non-linear frequency domain methods. *AIAA 2003-3875*.
- Peric, M. (1985, January). *A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts*. Ph. D. thesis, Imperial College - University of London.
- Pitz, R. W. and J. W. Daily (1983, November). Combustion in a turbulent mixing layer formed at a rearward-facing step. *AIAA Journal* 21(11), 1565–1570.
- Quarteroni, A., R. Sacco, and F. Saleri (2000). Iterative Methods for Solving Linear Systems. In *Numerical Mathematics.*, pp. 125–182. Springer.
- Rauschenberger, P. (2008). Multifrequential Harmonic Balance Technique. Technical Report CERFACS : WN/CFD/08/83.
- Rhie, C. and W. Chow (1983, November). Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 21(11), 1525–1532.
- Saad, Y. (2003). *Iterative methods for sparse linear systems* (Second Edition ed.). Society for Industrial and Applied Mathematics.
- Shen, W., J. A. Michelsen, and J. Sørensen (2001). Improved Rhie-Chow interpolation for unsteady flow computations. *AIAA Journal* 39(12), 2406–2409.
- Shyy, W., Y. Lian, J. Tang, H. Liu, P. Trizila, B. Stanford, L. Bernal, C. Cesnik, P. Friedmann, and P. Ifju (2008, August). Computational aerodynamics of low Reynolds number plunging, pitching and flexible wings for MAV applications. *Acta Mechanica Sinica* 24(4), 351–373.
- Sicot, F., G. Dufour, and N. Gourdain (2012). A Time-Domain Harmonic Balance Method for Rotor/Stator Interactions. *Journal of Turbomachinery* 134, 011001.
- Sicot, F., G. Puigt, and M. Montagnac (2008). Block-Jacobi Implicit Algorithms for the Time Spectral Method. *AIAA Journal* 46(12), 3080–3089.
- Spalart, P. R. and S. R. Allmaras (1992). A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting & Exhibit*, Reno, NV, pp. 1992.
- Spalart, P. R. and S. R. Allmaras (1994). A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 5–21.

- Stokes, G. G. (1851). On the Effect of the Internal Friction of Fluids on the Motion of Pendulums. *Transactions of the Cambridge Philosophical Society* 9, 8–106.
- Su, X. and X. Yuan (2010). Implicit solution of time spectral method for periodic unsteady flows. *International Journal for Numerical Methods in Fluids* 63, 860–876.
- Thomas, J., E. Dowell, and K. Hall (2004). Modeling viscous transonic limit-cycle oscillation behavior using a harmonic balance approach. *Journal of Aircraft* 41(6), 1266–1274.
- van der Weide, E., A. Gopinath, and A. Jameson (2005). Turbomachinery applications with the time spectral method. *AIAA 2005-4905*.
- Vassberg, J., A. Gopinath, and A. Jameson (2005, January). Revisiting the Vertical-Axis Wind-Turbine Design using Advanced Computational Fluid Dynamics. In *43rd AIAA ASM, AIAA 2005-0047*, Reno, NV, pp. 1–23.
- Visonneau, M., P. Queutey, G. B. Deng, J. Wackers, E. Guilmineau, A. Leroyer, and B. Mallol (2012, March). Computation of Free-Surface Viscous Flows around Self-propelled Ships with the Help of Sliding Grids. *COMPIT 2012*, 344–358.
- Welch, G., I. Milanovic, and K. Zaman (2005). Application of Harmonic Balance Technique to Synthetic Jets in Cross-Flow. *AIAA Paper 2005-1111*.
- Weller, H., G. Tabor, H. Jasak, and C. Fureby (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics* 12, 620.
- White, F. (2005, January). *Viscous fluid flow*. McGraw-Hill series in mechanical engineering. McGraw-Hill Higher Education.
- Yang, Z., M. DJ, and J. Sitaraman (2011). Prediction of Helicopter Maneuver Loads Using BDF/Time Spectral Method on Unstructured Meshes. *AIAA 2011-1122*, 1–21.
- Yang, Z. and D. J. Mavriplis (2010). Time spectral method for periodic and quasi-periodic unsteady computations on unstructured meshes. In *40th Fluid Dynamics Conference and Exhibit, AIAA 2010-5034*, pp. 1–18.