UNIVERSITY OF CALIFORNIA,
IRVINE

Time Spectral and Space-Time LU-SGS Implicit Methods for Unsteady Flow Computations

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Lei Zhan

Dissertation Committee:
Professor Feng Liu, Chair
Professor Dimitri Papamoschou
Professor Said E. Elghobashi

2015

DEDICATION

To my parents, to my parents-in-law,
to my grandmother-in-law and to my wife.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

First I would like to express my deepest gratitude to Professor Feng Liu, my advisor, for his valuable guidance and support throughout the years of my dissertation research. Without him I would not have the opportunity to pursue PhD studies at UC Irvine and work in the UCI CFD lab. His deep understanding of physical problems, unique perspective of computational methods inspired me to great interest in research and creative thinking.

I also want to thank Professor Papamoschou and Professor Elghobashi for the very careful reading and helpful suggestions about my dissertation, as well as their wonderful teachings in fluid mechanics and computational fluid dynamics courses.

My many thanks are extended to the members at UCI CFD Lab for their help not only in my PhD studies but also in my daily life on campus: Dr. Juntao Xiong, Dr. Alan Lai, Dr. Ya Liu, Kevin Launglucknavalai, Ferran Marti Duran, Rayomand Gundevia and Jie Zhu. My thanks also go to my friends at UC Irvine: Franz Hernandez, Dr. Yong Wang, Dr. Guoxiong Zhang, Peyman Nazari and Qin Chen.

My endless gratitudes go to my parents, parents-in-law and grandmother-in-law, for their unconditional love and support. I am very thankful to my wife, Yingshu He, without her understanding and support I would not be able to get through this long journey.

Finally, I would like to express my gratitude to the Department of Mechanical and Aerospace Engineering for granting me the fellowship and teaching assistant positions, so that I can finish the studies at UC Irvine and practice my teaching skills.

CURRICULUM VITAE

Lei Zhan

EDUCATION

Ph.D. in Mechanical and Aerospace Engineering **2015**
University of California, Irvine *Irvine, USA*

Master of Engineering in Fluid Mechanics **2007**
Northwestern Polytechnical University *Xi'an, China*

Bachelor of Engineering in Aircraft Design and Engineering **2004**
Northwestern Polytechnical University *Xi'an, China*

ABSTRACT OF THE DISSERTATION

Time Spectral and Space-Time LU-SGS Implicit Methods for Unsteady Flow Computations

By

Lei Zhan

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine, 2015

Professor Feng Liu, Chair

This dissertation proposes numerical methods for the Euler and Navier-Stokes equations with spectral discretization in time and a fast space-time coupled LU-SGS (ST-LU-SGS) method for solving the resultant implicit equations. Firstly, the Fourier time spectral method is studied for periodic problems with test cases. The problem of non-symmetric solutions for symmetric periodic flow problems, caused by odd numbers of intervals in a period, is discovered and discussed in detail. The requirement of ensuring symmetric solution is proposed. In problems where frequency is not known a priori, a new frequency search approach based on Fourier analysis of the lift coefficient is proposed to work with the time spectral method. Computational results show that initial guesses of the frequency far away from the exact value can be used if the new approach is applied before employing a gradient based method. A new Chebyshev time spectral method is proposed to solve non-periodic unsteady problems and is validated by test cases. Computational results show that this method is very efficient in simulating both periodic and non-periodic unsteady flows, especially the non-periodic problems.

The use of Fourier or Chebyshev spectral discretization in time results in implicit equations in time marching. Explicit Runge-Kutta methods have often been used to solve such implicit system of equations through the use of the dual-time stepping algorithm. Such methods are,

however, slow despite the use of acceleration schemes such as implicit residual smoothing and multigrid. We propose a new space-time LU-SGS (ST-LU-SGS) implicit scheme for both the Fourier and Chebyshev time spectral methods. In this scheme, the time domain is regarded as one additional dimension in space. Computational experiments show that this new scheme is faster than the explicit Runge-Kutta solver. For Navier-Stokes flow test cases, computations using the ST-LU-SGS implicit scheme is over ten times faster than the explicit Runge-Kutta solver. The ST-LU-SGS implicit scheme also works very well with the proposed frequency search approach. The ST-LU-SGS scheme is as efficient as the Block-Jacobi implicit algorithm and is more robust than the Block-Jacobi implicit algorithm. The proposed ST-LU-SGS scheme works for problems with either low frequency or high frequency while the Block-Jacobi implicit algorithm fails for high frequency flow problems.

Chapter 1

Introduction

1.1 Background and Motivation

All fluid flows can be distinguished into two main categories as steady flows and unsteady flows. The second category is the main concern of this dissertation. In practice, time dependent calculations are needed for various important applications, such as the study of the internal flow of turbomachinary, the analysis of the flow over helicopter rotors during forward flight and flutter simulation. In addition to improvements in computer hardware, improvements in numerical methods also play a significant role in reducing computational cost of time dependent calculations. As a main topic of numerically solving governing equations for unsteady flows, temporal discretization has been extensively studied.

Magnus and Yoshihara [1] achieved acceptable computational results for transonic flows using an Euler code based on Lax-Wendroff scheme. Then MacCormack[2] developed a popular explicit scheme, which used a variation of the two-step Lax-Wendroff scheme and avoided computing unknowns at midpoints. A drawback of these schemes for steady-state calculations is that the computed solution depends on time step hence is not unique. This

is the motivation of using the so-called Method of Lines, which separates the space and time discretization procedures. Jameson, Schmidt and Turkel [3] adopted this idea and proposed an explicit Runge-Kutta time-stepping scheme in combination with Finite Volume Method (FVM). This method is also known as the JST scheme. In this scheme, central differencing is used to calculated convective fluxes on faces of grid cells. A blended second-order and fourth-order artificial dissipation is added for stability and shock capture. Computational results showed that shock waves are well captured without wiggles in their neighbourhood. Convergence to a steady state is accelerated by the use of variable local time step and enthalpy damping. It is well known that the time step of an explicit scheme is restricted by the Courant-Friedrichs-Lewy (CFL) condition, which requires that the region of dependence of the difference scheme must include at least that of the differential equation. To relieve or even remove this limitation, an implicit scheme is desired for the time marching scheme. Implicit schemes can be generally classified into two main types, those employing residual averaging and those using approximate factorization. The most successful one in the first type is Jameson's multi-stage scheme [4], fast convergence was achieved using CFL numbers up to nine. In the second type of implicit scheme, the Alternating Direction Implicit (ADI) scheme has been well studied. Beam and Warming [5] proposed such a scheme that has been proved to be unconditionally stable. Jameson and Yoon [6] improved convergence of the ADI scheme by multigrid for calculating steady-state solutions of the two dimensional Euler equations. In steady-state calculations, to make computational results independent of the time step, an ADI scheme in delta form (increment of conserved variables and fluxes) is desired. Unfortunately, the ADI scheme is well known of being unstable in delta form. Alternatively, Jameson and Turkel [7] proposed the lower-upper(LU) factored implicit scheme in delta form. This scheme is unconditionally stable with any number of space dimensions. And the number of factors are always two even in three dimensional space. Based on this LU factored implicit scheme, Yoon and Jameson [8] [9] [10] introduced the widely-used Lower-Upper Symmetric Gauss-Seidel (LU-SGS) scheme. This scheme has the advantages of low

computational cost and moderate memory requirements, which are both comparable to an explicit multi-stage scheme. Besides it can be easily implemented on vector and parallel computers.

Although the above scheme can be applied to time-accurate calculations using real time marching directly, convergence acceleration techniques such as local time stepping and multigrid cannot be employed since time accuracy is required. To overcome this problem, Jameson [11] proposed a dual-time stepping scheme. In this scheme the real time derivative is discretized by a three-step, second-order Backward Difference Formula (BDF). And at each real time level, explicit multi-stage Runge-Kutta pseudo-time marching is carried out. Multigrid and local time stepping can then be used in pseudo-time marching without losing time accuracy. Although these popular convergence acceleration techniques can be applied, the computational efficiency using the original dual-time stepping scheme is limited for two reasons. The first reason lies in the use of finite differencing in approximating the real time derivative. There are only three real time levels involved in BDF method and it can only achieve second-order accuracy in real time. Then very small real time steps are usually used due to time accuracy requirement. For example, this is the case in turbomachinery flows, which are highly complex, unsteady, three dimensional and turbulent. If methods of higher than algebraic order can be found to approximate the real time derivative, much larger real time steps, and hence much fewer real time levels can be used to improve computational efficiency. Second, the original dual-time stepping method uses explicit Runge-Kutta pseudo-time marching to obtain steady-state solution on each real time level. The pseudo-time step is limited by CFL condition. To greatly increase the maximum allowable CFL number hence much faster convergence of pseudo-time marching, implicit scheme is desirable. The work of this dissertation aims at replacing the BDF method and explicit Runge-Kutta scheme with new algorithms so as to improve efficiency of unsteady computations.

To increase the time accuracy of a dual-time stepping method, attempts were first made

for computations of periodic flows. Hall et al. [12] proposed a linearized method for unsteady computations. In this method every flow variable is assumed to be a sum of time averaged value and a small perturbation. Based on the small perturbation assumption the flow governing equation can be simplified and separated into two sets of equations. One is for time averaged variables and the other is for small perturbation components. Coefficients in the latter can be known by solving the former. Then the small perturbation components are assumed as harmonic functions with the known frequency. Although this approach can provide analysis on unsteady flow field quite efficiently, the flaw is obvious. This method is unable to solve flow problems with large time variation. For example when there is shock movement in the flow, then the time variation of flow variables is large and highly non-linear. And this is always the case that people have to face in the turbomachinery industry.

Although the time-linearized Euler methods provide the turbomachinery designers with efficient solvers for aerodynamic response predictions, their effectiveness is limited by the linear assumption. It is of great importance to develop a method that can account for nonlinear effects, while remain computationally efficient. Ning and He [13] proposed a Nonlinear Harmonic method to meet that objective. In their method, the harmonic perturbations are added to a time-averaged flow field to approximate the nonlinear unsteady flow field. Due to the time-averaging on the nonlinear flow equations, extra "unsteady stress" terms are generated in time-averaged equations. These extra terms are computed from the solution of the unsteady perturbation equations, while the coefficients of the perturbation equations depends on the solution of the time-averaged equations. The two sets of equations are solved simultaneously in a strongly coupled manner. Numerical results show that the nonlinear effects, such as oscillation of shock waves, can be effectively and efficiently modeled by the nonlinear harmonic method.

The above approach was among the earliest methods that account for nonlinear effect in the way of reserving extra "unsteady stress" terms in the time averaged equations. However, the

unsteadyness is still evaluated by the first harmonic mode only. No harmonic modes with high frequencies are involved. Therefore the effectiveness of this method could be limited in unsteady problems with large time variation. Besides the formulation of the method requires a lot of efforts, especially for three-dimensional viscous unsteady flow problems. Later, to develop a method that includes more harmonic modes and can be easily applied to the real problems in engineering practice, Hall[14] proposed a new method suitable for solving unsteady problems with large time variation. Since the flows inside Turbomachinery are usually periodic, flow variables can be expanded as Fourier series. If only the first several harmonics are kept in the expression then this model can be used in practice. Hall successfully developed a method based on this model. This method is called harmonic balance method. It is capable of dealing with large non-linear time variation since the model is non-linear. Hall derived the new governing equations in which the time derivative terms are evaluated using the time spectral operator. Then the equations look like steady ones and time derivative terms become source terms. In this method equations for all real time level are strongly coupled and solved at the same time. Solutions on all instants are connected by the spectral operator. Then pseudo-time is introduced into the equations. So the equations can be solved by pseudo-time marching. Convergence accelerating techniques such as the multigrid method can be used directly in the computation. Hall applied the method to solve two-dimensional viscous flow past cascades. Since the BDF is replaced by spectral operator, the discretization in real time achieves higher than algebraic accuracy. Therefore to reach the same time accuracy level, the harmonic balance method allows the use of larger real time steps. Since fewer real time levels are required, computational efficiency of harmonic balance should be higher than that using BDF. The greatly improved efficiency is verified and non-linear time variation of flow variables due to high flow speed and viscosity are well resolved. In this method the accuracy and efficiency are totally determined by how many and which harmonics are retained in the expansion. The more accurate the result is, the less efficient the computation is. Generally speaking this method has been both efficient and

accurate for many applications.

McMullen[15] proposed another way to implement the method. In that technique the governing equations are solved in the frequency domain rather than time domain. The corresponding solver is called the non-linear frequency domain solver. In the frequency domain the variables that need to be solved are complex coefficients for each Fourier mode. However, that is not to say the entire computation is carried out in the frequency domain. Repeated transformations between the time domain and the frequency domain for both solutions and residuals are involved in the computation. This is because the residuals are not linear functions of solutions. McMullen applied this technique to computations of external flows, such as flow past a pitching airfoil and vortex shedding flow behind a circular cylinder. McMullen's work not only provided another way to realize the idea of using Fourier expansion in the time domain but also showed that this type of method can be successfully used to simulate periodic external flows.

McMullen[16] also did a series of estimations on the efficiency of the non-linear frequency domain solver. The estimation is based on analyzing integral quantities such as the lift coefficient and moment coefficient. According to the estimation, the Non-linear frequency domain method is much more efficient than dual-time stepping scheme using BDF method if integral quantities are mainly concerned. McMullen also pointed out that the efficiency of the method may be undermined if the objective is to get satisfying results for individual unsteady quantities such as pressure coefficient.

Later Gopinath[17] applied the harmonic balance method to the computation of the 3-D unsteady flows inside multi-stage Turbomachinery. In that computation only one blade passage is used for each blade row. The harmonic balance method is shown to greatly reduce the whole cost of the simulation while capturing the dominant unsteadiness and the main flow features such as the interaction between two blade rows. This means the method can be successfully used in the simulation of 3-D periodic turbulent internal flows.

Ekici[18] accomplished the computation of 3-D periodic inviscid flows around helicopter rotors using the harmonic balance method. The method was reported to be able to resolve significant unsteady flow features. Since the harmonic balance equations are mathematically steady state equations the corresponding method was said to be well-suited for adjoint sensitivity analysis. So this method can provide a good platform for rotor optimization, such as improving performance and reducing noise.

The validity of both harmonic balance method and non-linear frequency domain method are limited for two reasons: First, these method can only solve periodic flow problems. Otherwise flow variables cannot be precisely expanded as Fourier series; Second, the frequency must be explicitly given for each iteration of the computation since the frequency appears explicitly in the approximation of time derivative terms. All the flow problems mentioned above fulfil these two requirements. However, in some periodic flow problems, the frequency is not known a priori such as the vortex shedding flow behind a cylinder and flow over an airfoil at large angles of attack. To apply the harmonic balance method or the non-linear frequency domain method to this kind of flows, a frequency search process must be involved in the computation. McMullen[15] proposed a method based on minimizing residuals. It is obvious that if the frequency is correct then the unsteady residual should be zero. In the opposite, an incorrect frequency results in a finite unsteady residual. Based on this fact the negative gradient of the squared unsteady residual with respect to frequency is used such that the updated frequency lowers the squared unsteady residual. The whole computation process is like a repeated feedback process. First an guessed initial value for frequency is used. Then the flow field is updated based on this frequency. The newly updated flow field produces a new gradient. Next the frequency is updated using the new gradient. This process will not stop until both the frequency and the flow field converge. McMullen found the correct frequency of the vortex shedding flow behind circular cylinder using this so-called Gradient Based Variable Time Period (GBVTP) approach. Also the curve for residual versus frequency was shown. It is interesting to notice that only when the frequency is very close

to the correct value the corresponding residual drops dramatically. Gopinath[19] used the GBVTP method along with a variation of the harmonic balance method to simulate vortex shedding flows behind circular cylinder and airfoil at high angle of attack. The set of the methods are tested to be able to give the correct frequency and flow field. Gopinath also gave the variation of harmonic balance method a new name, Time Spectral Method, since the governing equations are completely solved in time domain. Formula for both odd and even number of time intervals were derived. Stability of Time Spectral Method was also discussed there. Ekici[20] applied GBVTP along with harmonic balance method to find out frequency of limit cycle oscillation of blade in turbomachinery. In that work the harmonic balance method is used in solving both flow governing equations and structure vibration equations. However GBVTP works only on structure vibration equations. Generally speaking whether the GBVTP approach works or not depends on the choice of the initial guess for frequency. Since the GBVTP method depends on searching for a local optimum, the initial value for frequency should not be too far from the correct value. So this method is good at dealing with the case for which the estimated value of frequency is already known. Besides the GBVTP method is not efficient because the frequency searching is driven by a residual related quantity.

Spiker[21] suggested another way to do frequency searching. Recall that when the frequency is about right the corresponding residual is very small. That means the related flow field must be very similar to that for correct frequency. Spiker looked into these solutions and found that when frequency is quite close to the correct value the so-called phase drift phenomenon happens to integral quantities, such as lift coefficient. If the correct frequency is finally determined and used, the lift coefficient at all real time levels should reach steady state. Whereas for the case where the deviation in frequency is very small the lift coefficient on all time instants will keep varying. When viewing the time variation of the lift coefficient in the frequency domain and only looking at the first harmonic, there are two interesting features. The amplitude is about correct and does not change with iterations. However, the phase

is changing at a constant rate. Ekici[20] reported this phenomenon in the study of limit cycle oscillations too. Spiker also noticed that the phase change per unit iteration is directly proportional to deviation in frequency. And if the frequency is correct, phase change per unit iteration is zero. Then this linear relation was used to do frequency searching. The frequency searching process only needs to run two cases for two different frequencies to establish the linear relation. In this way the correct frequency can be found out. This method seems faster than GBVTP but the initial guess of frequency should be close to the exact frequency as well.

All the above works focused on periodic problems and all the methods that have been developed so far are only capable of dealing with periodic problems. For aperiodic flows suitable unsteady solvers are still under development. Ekici[22] did an exploration on an aperiodic flow in which there exist two periodic excitations. In this research partial sum of Fourier modes with different frequencies is also used to approximate flow variables. However, the frequency of each Fourier mode is a different linear combination of the two excitation frequencies. Ratio between any two frequencies of two different Fourier modes is not an integer or reciprocal of an integer so the flow is aperiodic. The method was applied to the flow inside turbines where both unsteady incoming wake and plunging motion of turbine blades exist. Computations showed that this method can obtain results with satisfying accuracy. At the same time the computation was quite efficient. Although there are better ways to approximate time derivatives for aperiodic flows but the attempt made by Ekici started the research on developing aperiodic flow solvers and the use of these solvers in practical engineering problems. Though the method proposed by Ekici is still called harmonic balance technique, the name Time Spectral Method (TSM) would be adopted in this dissertation since the governing equations are generally solved in time domain.

To make time spectral method suitable for general non-periodic flow problems, general expansions of a function rather than the Fourier expansion are adopted. These expansions

should work without requiring periodic condition in the time interval of interest. One of such expansions is the Chebyshev polynomial expansion. Kopriva [23] provided details of how to use Discrete Chebyshev Transforms and its inverse transform to evaluate derivative of function that is approximated by Chebyshev expansion. The Discrete Chebyshev Transform is performed based on Fast Fourier Transform, which is popular due to its well known high efficiency. These tools could be used in developing a new Chebyshev Time Spectral Method for non-periodic flow problems.

While the time accuracy of a dual-time stepping method can be increased by replacing BDF with TSM, the explicit Runge-Kutta pseudo-time scheme was still used in conjunction with time spectral method. Recently several efforts have been spent to employ implicit pseudo-time scheme suitable for time spectral methods. Sicot, Puigt and Montagnac [24] proposed a Block-Jacobi Implicit Algorithms for the time spectral method. This is a representative algorithm that makes use of the original LU-SGS scheme for steady-state computation. The approximated time derivative term by time spectral operator was included in the implicit operator. When solving this discretized system, Block Jacobi iteration method was introduced. Solution is updated after a number of Block Jacobi steps. And at each Block Jacobi step, even number of forward and backward LU-SGS sweeps are employed. During each LU-SGS sweep, the implicit term due to time spectral operator is regarded as a known source term and frozen. This term is only updated upon completion of an LU-SGS sweep. This method was applied to calculating turbulent unsteady flow over a pitching wing. Fast convergence was reported since large CFL numbers, such as one hundred, could be used in conjunction with V-cycle multigrid. Woodgate et al. [25] developed a totally different implicit scheme to efficiently solving for time spectral solutions. Since the implicit operator with contribution of time spectral discretization is a large sparse matrix, Krylov-type solver is used to replace pseudo-time iteration method. A preconditioner of the block incomplete lower and upper(BILU) factorization type is also employed with the Krylov-type solver. The method was tested with flows over pitching airfoil or wing tip in complex configuration.

Computational results showed favourable convergence speed and CPU time consumption. Su et al.[26] came up with a similar idea of using the multigrid preconditioned generalized minimal residual(GMRES) method, which is still a Krylov-type solver. However, the BILU type preconditioner was replaced by a Gauss-Seidel iteration type preconditioner to reduce memory requirement. Fast convergence was achieved for high frequency periodic flow over a blade of turbomachinery. Thomas et al. [27] proposed a two-step implicit scheme for time spectral method. In the first step of the method, a temporary solution increment is solved without considering time spectral operator in the implicit operator. Then the final solution increment is obtained based on the temporary one by converting a matrix that just reflects contribution of time spectral method. Existing ADI or LU-SGS scheme for steady-state solution is used in the first step of the method.

Although all the above proposed implicit schemes for time spectral method can further improve computational efficiency of periodic solution calculations, treatments in space and time are separated when solving the implicit system that results from time spectral method. So in addition to the implicit algorithm in space, these schemes require various complicated algorithm that deal with the coupling term due to the time spectral method in the implicit operator. For this reason, the work in this dissertation is aimed at seeking a new implicit scheme for time spectral method so as to improve the efficiency of pseudo-time marching for the time spectral method. The new implicit scheme uses a unified treatment in both space and time in solving the implicit system for the time spectral method; hence it is called the Space-Time LU-SGS (ST-LU-SGS) method. It is simpler and expected to provide faster and more robust convergence because of the close coupling between space and time.

1.2 This Dissertation

In this dissertation, firstly, the Fourier TSM will be studied with test cases. The non-symmetric solutions caused by odd number of intervals in a period for symmetric periodic flow problems will be discussed in detail. Requirements of ensuring symmetric solutions will be concluded. Then the Fourier time spectral method is applied to periodic flow problems where the period is not a priori. Effort will be spent to find better approach of frequency searching so as to use initial guesses of the frequency that are relatively far away from the exact value. Since the Fourier time spectral method is theoretically only capable of solving periodic unsteady problems, a new Chebyshev TSM will be proposed to solve non-periodic unsteady problems. The method is validated by solving one dimensional viscous Burgers' equation and two dimensional inviscid periodic and non-periodic flow problems. Computational results show that this method provides higher accuracy in time than any algebraic order finite difference method. So it is very efficient in solving unsteady problems, especially non-periodic flow problems.

To remove or at least relax the stability or accuracy requirement for the explicit Runge-Kutta scheme that is applied to solve the non-linear algebraic equations resulting from time spectral methods, a new space-time LU-SGS (ST-LU-SGS) implicit scheme is proposed for time spectral methods. In that scheme, time domain is regarded as one additional dimension in space. This new scheme can make computation converge faster and save CPU time in each multigrid cycle compared to an explicit Runge-Kutta solver. Hence the total consumed CPU time for an unsteady computation is greatly saved. Finally the ST-LU-SGS scheme will be compared with a similar but different Block-Jacobi implicit algorithm. The ST-LU-SGS method proposed in this dissertation offers at least the same efficiency as the Block-Jacobi implicit algorithm. In addition it is stable and works for problems with high frequency where the Block-Jacobian method becomes unstable and fails.

Chapter 2

Review on Basic Numerical Methods

In this chapter, basic numerical methods used in this dissertation are presented. All solvers employed in this dissertation use a cell-centered finite volume method. Central differencing is used to calculate numerical fluxes. The JST scheme [3] is adopted to add second-order and fourth-order artificial dissipation for stability consideration. To reach a steady-state solution, explicit multi-stage Runge-Kutta scheme is applied as pseudo-time stepping method. When time-accurate solution is desired, a dual-time stepping method using backward difference formula is introduced to modify the steady problem solver.

2.1 Cell-centered Finite Volume Method on Structured Grids

In the present work, structured grids and cell-centered finite volume method are used in all solvers. The flow domain is decomposed into small cells. For structured grids and cell-centered scheme, these cells can be easily ordered using indices i,j,k of cell centers. For the sake of simplicity, the finite volume method is presented on two dimensional Euler equations.

In conservative differential form, the Euler equations can be written as

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{2.1}$$

where

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \; f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho E u + p u \end{pmatrix}, \; g = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho E v + p v \end{pmatrix} \tag{2.2}$$

where t is time, ρ density, p pressure. x and y stand for coordinates in two dimensional space; u and v, are the flow velocity components in the x and y directions; Total energy and total enthalpy are $E = e + \frac{u^2+v^2}{2}$ and $H = h + \frac{u^2+v^2}{2}$, respectively, with $h = e + \frac{p}{\rho}$ and $e = \frac{p}{(\gamma-1)\rho}$. γ is the ratio of specific heats.

Integrating equation (2.1) over a grid cell, the equation becomes

$$\frac{\partial}{\partial t} \int_\Omega \mathbf{W} d\Omega + \int_S \mathbf{F} \cdot \vec{n} dS = 0 \tag{2.3}$$

where $\mathbf{F} = f\vec{i} + g\vec{j}$ and $\vec{n} = n_x\vec{i} + n_y\vec{j}$ is the outward unit normal vector on cell surfaces. Ω is volume of the grid cell. For moving grid, f and g in equation (2.2) should be modified as

$$f = \begin{pmatrix} \rho \bar{u} \\ \rho u \bar{u} + p \\ \rho v \bar{u} \\ \rho E \bar{u} + p u \end{pmatrix}, \; g = \begin{pmatrix} \rho \bar{v} \\ \rho u \bar{v} \\ \rho v \bar{v} + p \\ \rho E \bar{v} + p v \end{pmatrix} \tag{2.4}$$

where$(\bar{u}, \bar{v}) = (u - u_b, v - v_b)$ is convective velocity relative to moving grid on a cell surface, and (u_b, v_b) is the velocity of the same cell surface. For grid cell(i,j), following the method of

lines and only discretize in space, the semi-discrete form of equation (2.3) can be written as

$$\frac{d}{dt}(\mathbf{W}_{i,j}\Omega_{i,j}) + \mathbf{C}_{i,j} = 0 \tag{2.5}$$

where $\Omega_{i,j}$ is volume of grid cell (i,j), $\mathbf{W}_{i,j}$ is the volume-averaged solution vector of state variables over that cell; and $\mathbf{C}_{i,j}$ is the finite volume approximation of the net convective fluxes flowing out of the cell that can be formulated for cell-centered scheme as

$$\begin{aligned}\mathbf{C}_{i,j} =& \mathbf{F}_{i-\frac{1}{2},j} \cdot \vec{S}_{i-\frac{1}{2},j} + \mathbf{F}_{i+\frac{1}{2},j} \cdot \vec{S}_{i+\frac{1}{2},j} + \\ & \mathbf{F}_{i,j-\frac{1}{2}} \cdot \vec{S}_{i,j-\frac{1}{2}} + \mathbf{F}_{i,j+\frac{1}{2}} \cdot \vec{S}_{i,j+\frac{1}{2}}\end{aligned} \tag{2.6}$$

where $\vec{S}$ is outward normal surface vector on cell faces. It is expressed as $\vec{S}_{i+\frac{1}{2},j} = \vec{n}_{i+\frac{1}{2},j}\Delta S_{i+\frac{1}{2},j}$ on cell face $(i+\frac{1}{2},j)$ for example. Here $\Delta S_{i+\frac{1}{2},j}$ is the cell face area (length in two dimension) on cell face $(i+\frac{1}{2},j)$. In this dissertation, the convective flux vectors at all four cell surfaces are evaluated as the the average of those on the two adjacent cells as:

$$\begin{aligned}\mathbf{F}_{i-\frac{1}{2},j} &= \frac{1}{2}(\mathbf{F}_{i-1,j} + \mathbf{F}_{i,j}), \mathbf{F}_{i+\frac{1}{2},j} = \frac{1}{2}(\mathbf{F}_{i,j} + \mathbf{F}_{i+1,j}) \\ \mathbf{F}_{i,j-\frac{1}{2}} &= \frac{1}{2}(\mathbf{F}_{i,j-1} + \mathbf{F}_{i,j}), \mathbf{F}_{i,j+\frac{1}{2}} = \frac{1}{2}(\mathbf{F}_{i,j} + \mathbf{F}_{i,j+1})\end{aligned} \tag{2.7}$$

The above treatment of convective flux on cell faces is equivalent to second-order central differencing on a Cartesian grid. It is well known that a central differencing scheme alone suffers from odd-even decoupled spurious solutions as well as undesired oscillations in the neighbourhood of shock waves for transonic flows. To solve these problems, artificial dissipation is required and equation (2.5) can be revised as

$$\frac{d}{dt}(\mathbf{W}_{i,j}\Omega_{i,j}) + \mathbf{C}_{i,j} - \mathbf{D}_{i,j} = 0 \tag{2.8}$$

where $\mathbf{D}_{i,j}$ is the artificial dissipation flux added to grid cell (i,j). To remove oscillations near shock waves, MacCormack et al.[28] introduced a second-order dissipation term. Then

Beam et al.[29] added a fourth-order dissipation term to provide background damping for stability and convergence needs. In the JST scheme proposed by Jameson, Schmidt and Turkel [3], these two necessary components are combined together. A switching function is also introduced to activate the second-order dissipation near shock wave while turning off the fourth-order dissipation, and switch off the second order term in smooth region while allowing for the operating of the fourth order term. In this dissertation, the JST scheme is adopted and the artificial dissipation is given as

$$\mathbf{D}_{i,j} = \mathbf{D}^{(2)}_{i,j} - \mathbf{D}^{(4)}_{i,j} \tag{2.9}$$

where $\mathbf{D}^{(2)}_{i,j}$ and $\mathbf{D}^{(4)}_{i,j}$ are second order and fourth order artificial dissipation fluxes, respectively. Each of them is a summation of the corresponding artificial dissipation vectors calculated on all cell faces:

$$\begin{aligned} \mathbf{D}^{(2)}_{i,j} &= \mathbf{d}^{(2)}_{i+\frac{1}{2},j} - \mathbf{d}^{(2)}_{i-\frac{1}{2},j} + \mathbf{d}^{(2)}_{i,j+\frac{1}{2}} - \mathbf{d}^{(2)}_{i,j-\frac{1}{2}} \\ \mathbf{D}^{(4)}_{i,j} &= \mathbf{d}^{(4)}_{i+\frac{1}{2},j} - \mathbf{d}^{(4)}_{i-\frac{1}{2},j} + \mathbf{d}^{(4)}_{i,j+\frac{1}{2}} - \mathbf{d}^{(4)}_{i,j-\frac{1}{2}} \end{aligned} \tag{2.10}$$

where $\mathbf{d}^{(2)}$ and $\mathbf{d}^{(4)}$ are the second order and fourth order artificial dissipation vectors evaluated on cell faces and can be defined as

$$\begin{aligned} \mathbf{d}^{(2)}_{i+\frac{1}{2},j} &= \Lambda_{i+\frac{1}{2},j}\epsilon^{(2)}_{i+\frac{1}{2},j}(\mathbf{W}_{i+1,j} - \mathbf{W}_{i,j}) \\ \mathbf{d}^{(4)}_{i+\frac{1}{2},j} &= \Lambda_{i+\frac{1}{2},j}\epsilon^{(4)}_{i+\frac{1}{2},j}(\mathbf{W}_{i+2,j} - 3\mathbf{W}_{i+1,j} + 3\mathbf{W}_{i,j} - \mathbf{W}_{i-1,j}) \end{aligned} \tag{2.11}$$

on cell face $(i+\frac{1}{2}, j)$ for example. The Λ in equation (2.11) is a variable scaling factor and is set to be:

$$\Lambda_{i+\frac{1}{2},j} = 2\mathbf{min}(\lambda^I_{i+1,j}, \lambda^I_{i,j}) \tag{2.12}$$

where λ^I is the spectral radius of the convective flux Jacobian in the i direction, defined as

$$\lambda^I_{i,j} = \left|(\bar{u}_{i,j}\vec{i} + \bar{v}_{i,j}\vec{j}) \cdot \frac{\vec{S}_{i+\frac{1}{2},j} + \vec{S}_{i-\frac{1}{2},j}}{2}\right| + c_{i,j}\frac{\Delta S_{i+\frac{1}{2},j} + \Delta S_{i-\frac{1}{2},j}}{2} \tag{2.13}$$

where $c_{i,j}$ is the local speed of sound of the cell (i, j). Spectral radius in the J direction is given similarly. $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are the coefficients for the second order and fourth order artificial dissipation terms, respectively. They have the function of switching on or off the two dissipation terms and are defined as

$$\begin{aligned} \epsilon^{(2)}_{i+\frac{1}{2},j} &= \kappa^{(2)}\mathbf{min}[\frac{1}{4}, \mathbf{max}(\nu_{i+1,j}, \nu_{i,j})] \\ \epsilon^{(4)}_{i+\frac{1}{2},j} &= \mathbf{max}[0, (\kappa^{(4)} - \epsilon^{(2)}_{i+\frac{1}{2},j})] \end{aligned} \tag{2.14}$$

where $\kappa^{(2)}$ and $\kappa^{(4)}$ are two constants in the order of 1 and ν the JST switch function[quote ref] using the pressure as a sensor:

$$\nu_{i,j} = \left|\frac{p_{i-1,j} - 2p_{i,j} + p_{i+1,j}}{p_{i-1,j} + 2p_{i,j} + p_{i+1,j}}\right| \tag{2.15}$$

with the above switch function, the second order and fourth order dissipation can work properly in different regions of a flow field. In the region close to shock waves, the pressure has a sudden jump hence ν and $\epsilon^{(2)}$ are all in the order of 1. So the second order dissipation is enabled while the fourth order dissipation is disabled. On the other hand, in the smooth region, ν is negligible.Then the second order dissipation is switched off and the fourth order dissipation is functioning to provide background damping needed by central differencing scheme.

2.2 Explicit Runge-Kutta Pseudo-time Stepping Scheme

The equation (2.8) can also be written in the extensively used semi-discrete form as

$$\frac{d}{dt}(\mathbf{W}_{i,j}\Omega_{i,j}) + \mathbf{R}_{i,j}(\mathbf{W}_{i,j}) = 0 \tag{2.16}$$

where $\mathbf{R}_{i,j}(\mathbf{W}_{i,j})$ is the residual for cell (i, j) and is defined as

$$\mathbf{R}_{i,j}(\mathbf{W}_{i,j}) = \mathbf{C}_{i,j} - \mathbf{D}_{i,j} \tag{2.17}$$

For steady problem, the time derivative term in equation (2.17) disappears, and the solution of the equation should make the residual equal zero. However, the steady problem can still be solved by time marching method if the time derivative term is retained and the time is interpreted as a pseudo-time τ, which has no physical meaning. Then the equation that is actually to be solved for a steady problem can be written as

$$\frac{d}{d\tau}(\mathbf{W}_{i,j}\Omega_{i,j}) + \mathbf{R}_{i,j}(\mathbf{W}_{i,j}) = 0 \tag{2.18}$$

At the end of pseudo-time marching, the residual $\mathbf{R}_{i,j}(\mathbf{W}_{i,j})$ achieves zero and the desired solution is reached.

In this dissertation, a five-stage Runge-Kutta scheme is employed as the explicit pseudo-time

marching method, which is given as

$$
\begin{aligned}
\mathbf{W}^{(0)} &= \mathbf{W}^{q} \\
\mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \frac{1}{4}\frac{\Delta\tau}{\Omega^{q}}\mathbf{R}(\mathbf{W}^{(0)}) \\
\mathbf{W}^{(2)} &= \mathbf{W}^{(1)} - \frac{1}{6}\frac{\Delta\tau}{\Omega^{q}}\mathbf{R}(\mathbf{W}^{(1)}) \\
\mathbf{W}^{(3)} &= \mathbf{W}^{(2)} - \frac{3}{8}\frac{\Delta\tau}{\Omega^{q}}\mathbf{R}(\mathbf{W}^{(2)}) \\
\mathbf{W}^{(4)} &= \mathbf{W}^{(3)} - \frac{1}{2}\frac{\Delta\tau}{\Omega^{q}}\mathbf{R}(\mathbf{W}^{(3)}) \\
\mathbf{W}^{(5)} &= \mathbf{W}^{(4)} - \frac{\Delta\tau}{\Omega^{q}}\mathbf{R}(\mathbf{W}^{(0)}) \\
\mathbf{W}^{q+1} &= \mathbf{W}^{(5)}
\end{aligned}
\tag{2.19}
$$

where superscript $q+1$ and q refer to two successive pseudo-time steps. Besides, techniques such as local time stepping, residual smoothing and multigrid method can be applied to accelerate convergence to the steady-state solution.

2.3 Dual-time Stepping Method Using Backward Difference Formula

In unsteady computations, the time appearing in equation (2.16) has physical meaning and the time derivative term does not vanish. Following Jameson [11], the real time derivative term can be approximated by an implicit(for real time) backward difference formula of second order accuracy, then the unsteady Euler equation in fully discretized form is written as

$$
\frac{3}{2\Delta t}(\mathbf{W}^{n+1}V^{n+1}) - \frac{2}{\Delta t}(\mathbf{W}^{n}V^{n}) + \frac{1}{2\Delta t}(\mathbf{W}^{n-1}V^{n-1}) + \mathbf{R}(\mathbf{W}^{n+1}) = 0 \tag{2.20}
$$

The above equation can be solved by pseudo-time marching method in the same way for steady problems, then the equation that is to be actually solved is

$$\frac{d\mathbf{W}}{d\tau} + \mathbf{R}^*(\mathbf{W}) = 0 \tag{2.21}$$

where

$$\mathbf{R}^*(\mathbf{W}) = \frac{3}{2\Delta t}(\mathbf{W}V) - \frac{2}{\Delta t}(\mathbf{W}^n V^n) + \frac{1}{2\Delta t}(\mathbf{W}^{n-1}V^{n-1}) + \mathbf{R}(\mathbf{W}) \tag{2.22}$$

is the discretized unsteady residual. Although equation (2.20) is an implicit equation in real time, the pseudo-time marching method used to solve it can still employ the explicit Runge-Kutta scheme. Convergence acceleration techniques mentioned before can all be used again during pseudo-time marching procedure at each real time level. Once a steady-state solution is obtained at one real time level, the real time is marched forward to next level and a new round of pseudo-time marching will be conducted. This implicit-explicit, real time and pseudo-time marching method for time-accurate computations is called the dual-time stepping scheme using backward difference formula(BDF).

The dual-time stepping scheme using BDF method and explicit Runge-Kutta pseudo-time marching is popular in unsteady calculations due to the following advantages. First, it can be applied to any unsteady problems and not restricted to periodic problems. For periodic flows, the method can also be used in those with unknown frequency. Second, only three real time levels are required in the BDF method. So the memory spent on recording flow fields is low. However, the use of both BDF method and the explicit Runge-Kutta scheme usually results in low computational efficiency for unsteady calculations. First, the time accuracy using the BDF method is low. To achieve satisfactory time accuracy, the real time step needs to be set very small; hence a large number of real time levels are involved in the time interval of interest. This is one source of high computational cost. Besides, the pseudo-time step of the explicit Runge-Kutta scheme is limited by CFL condition. So big pseudo-time steps are not allowed due to stability requirement and hence slow convergence to steady-

state solution on each real time level. The main concern of the work in this dissertation is to improve the computational efficiency of unsteady flow calculations. The dual-time concept is retained since convergence acceleration techniques, such as multigrid and local pseudo time stepping, can be applied with out losing time accuracy. The time accuracy could be increased by replacing BDF method with time spectral method, which is presented and discussed in Chapter 3. And the efficiency of pseudo-time marching can be improved by replacing the explicit Runge-Kutta scheme by implicit schemes suitable for time spectral method. This is the topic of Chapter 4.

Chapter 3

The Time Spectral Method

As discussed in preceding chapters, the time accuracy of BDF method is low since only three real time levels are used to discretize the real time derivative term. Whereas time spectral method couples all real time levels in the time interval of interest for discretization of the real time derivative term; hence the time accuracy of time spectral method is much higher. However, compared to BDF method, the time spectral method has the following drawbacks. First, since flow field on all real time levels are coupled in time spectral method, the memory spent on recording flow field is higher than that of BDF method. But due to the rapid development of computer hardware, this drawback is not a major issue. Second, recently the time spectral method has been developed based on Fourier expansion. Strictly speaking, the Fourier time spectral method only works for periodic problems. So far the reported application of Fourier time spectral method are mainly periodic flow problems. This means that the frequency of a periodic flow problem must be given since it appears explicitly in the formulation. So for a periodic problem where the frequency is not known a priori, a frequency search process must be included to work for the Fourier time spectral method. In this chapter the Fourier time spectral method is derived and validated with test case first. Then the non-symmetric solution of a symmetric problem using odd numbers of intervals

is discussed. For the periodic problems where the frequency is not known a priori, a new frequency search method is proposed. Finally, a time spectral method based on Chebyshev expansion is proposed for non-periodic unsteady flow computations.

3.1 The Fourier Time Spectral Method and its Derivation

Assuming that the volume of grid cell does not change with time, the unsteady flow governing equations in semi-discrete form can be written as

$$\Omega \frac{d\mathbf{W}(t)}{dt} + \mathbf{R}(\mathbf{W}(t)) = 0 \tag{3.1}$$

The discrete Fourier transform of $\mathbf{W}(t)$, for an angular frequency ω and N time intervals in a period, is given by

$$\tilde{\mathbf{W}}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{W}_n e^{-ik\omega n\Delta t} \tag{3.2}$$

where $\tilde{\mathbf{W}}_k$ is the Fourier coefficients for the Fourier mode of frequency $k\omega$, $\Delta t = \frac{2\pi}{\omega N}$, $\mathbf{W}_n$ is solution on real time level n at $t = n\Delta t$ and i the imaginary unit. The inverse transform is given by

$$\mathbf{W}_n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{\mathbf{W}}_k e^{ik\omega n\Delta t} \tag{3.3}$$

for N even and

$$\mathbf{W}_n = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \tilde{\mathbf{W}}_k e^{ik\omega n\Delta t} \tag{3.4}$$

for N odd. The steady residual $\mathbf{R}(\mathbf{W})$ can be expanded as discrete Fourier series similarly. Substituting these expansions into equation (3.1), McMullen et.al.[15] solved the unsteady solution by transforming equations (3.6) into frequency domain and using pseudo-

time marching as

$$\Omega\frac{\partial\tilde{\mathbf{W}}_k}{\partial\tau}+\Omega ik\omega\tilde{\mathbf{W}}_k+\tilde{\mathbf{R}}_k=0, 0\leq k\leq N-1 \tag{3.5}$$

This frequency domain method requires transforming both the solution $\mathbf{W}$ and the non-linear steady residual $\mathbf{R}(\mathbf{W})$ back and forth between time domain and frequency domain. Alternatively, the Time Spectral method solves equation (3.1) mainly in time domain, and at least Fourier transforms on the non-linear steady residual $\mathbf{R}(\mathbf{W})$ are avoided. Applying the Fourier Time Spectral Method, equation (3.1) can be written as

$$\Omega\mathbf{D}_{t,F,n}\mathbf{W}_n+\mathbf{R}(\mathbf{W}_n)=0, 0\leq n\leq N-1 \tag{3.6}$$

where $\mathbf{D}_{t,F,n}$ is the Fourier time spectral operator for $\mathbf{W}_n$. Gopinath et al.[19] specified the operator $\mathbf{D}_{t,F,n}$ as

$$\mathbf{D}_{t,F,n}\mathbf{W}_n=\omega\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}ik\tilde{\mathbf{W}}_k e^{ik\omega n\Delta t} \tag{3.7}$$

for N even or

$$\mathbf{D}_{t,F,n}\mathbf{W}_n=\omega\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}ik\tilde{\mathbf{W}}_k e^{ik\omega n\Delta t} \tag{3.8}$$

for N odd. If written in terms of solution vector $\mathbf{W}_n$ in time domain, equations (3.7) and (3.8) can be further specified as

$$\mathbf{D}_{t,F,n}\mathbf{W}_n=\omega\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}d_m^{even}\mathbf{W}^{n+m} \tag{3.9}$$

for N even or

$$\mathbf{D}_{t,F,n}\mathbf{W}_n=\omega\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}}d_m^{odd}\mathbf{W}^{n+m} \tag{3.10}$$

for N odd. d_m^{even} and d_m^{odd} are given by

$$d_m^{even} = \begin{cases} \frac{1}{2}(-1)^{m+1}\mathbf{cot}(\frac{\pi m}{N}), & m \neq 0 \\ 0, & m = 0 \end{cases} \tag{3.11}$$

and

$$d_m^{odd} = \begin{cases} \frac{1}{2}(-1)^{m+1}\mathbf{cosec}(\frac{\pi m}{N}), & m \neq 0 \\ 0, & m = 0 \end{cases} \tag{3.12}$$

respectively. Using this treatment, no discrete Fourier transforms are need in implementing the Fourier Time Spectral Method and the unsteady computations are completely carried out in time domain. However, this approach depends on knowing details of the Discrete Fourier transform and its inverse transform. Since the operations involved in these transforms are relatively simple, it is practical to implement time spectral method in this way. The Fourier time spectral operator $D_{t,F,n}$ can be easily formulated using simple functions. If Time Spectral Method is developed based on other expansions such as Chebyshev expansion or Legendre expansion, the operations involved in the corresponding discrete transform may not be simple. And the formulation of operator $D_{t,n}$ may be complicated. To avoid this potential problem and make the derivation of time spectral operator based on different expansions in the unified way, Hall's treatment [14] is adopted in present work. This involves applying Fast Fourier Transform(FFT) and its inverse transform(IFFT) on solution $\mathbf{W}(t)$ only to get the approximation of time derivative term, and the governing equations are still solved in time domain. Actually this is the original form of the so called Time Spectral Method when it was firstly proposed by Hall et al. [14].

Following the derivation given by Hall et al.[14], the extended solution vector $\mathbf{W}^*$ can be related to its Fourier coefficients $\tilde{\mathbf{W}}$ as

$$\mathbf{W}^* = \mathbf{E}^{-1}\tilde{\mathbf{W}}^* \tag{3.13}$$

where $\mathbf{W}^* = [\mathbf{W}_0, \ldots, \mathbf{W}_{N-1}]^T$ and $\tilde{\mathbf{W}}^* = [\tilde{\mathbf{W}}_0, \ldots, \tilde{\mathbf{W}}_{\frac{N-1}{2}}, \tilde{\mathbf{W}}_{-\frac{N-1}{2}}, ..., \tilde{\mathbf{W}}_{-1}]^T$ for N odd or $\tilde{\mathbf{W}}^* = [\tilde{\mathbf{W}}_0, \ldots, \tilde{\mathbf{W}}_{\frac{N}{2}-1}, \tilde{\mathbf{W}}_{-\frac{N}{2}}, \tilde{\mathbf{W}}_{-\frac{N}{2}+1}, ..., \tilde{\mathbf{W}}_{-1}]^T$ for N even. E^{-1} is the IFFT operator.

Correspondingly,

$$\tilde{\mathbf{W}}^* = \mathbf{E}\mathbf{W}^* \tag{3.14}$$

where E is the FFT operator. Using the Fourier time spectral method to approximate the time derivative of the extended solution vector $\mathbf{W}^*$, the original unsteady governing equation in semi-discrete form can be written as

$$\Omega \mathbf{D}_{t,F} \mathbf{W}^* + \mathbf{R}(\mathbf{W}^*) = 0 \tag{3.15}$$

and the Fourier time spectral operator $\mathbf{D}_{t,F}$ can be derived as

$$\mathbf{D}_{t,F}\mathbf{W}^* = \frac{\partial \mathbf{E}^{-1}\mathbf{E}\mathbf{W}^*}{\partial t} = \frac{\partial \mathbf{E}^{-1}\tilde{\mathbf{W}}^*}{\partial t} = \frac{\partial \mathbf{E}^{-1}}{\partial t}\tilde{\mathbf{W}}^* = \frac{\partial \mathbf{E}^{-1}}{\partial t}\mathbf{E}\mathbf{W}^* \tag{3.16}$$

Then it can be concluded that

$$\mathbf{D}_{t,F} = \frac{\partial \mathbf{E}^{-1}}{\partial t}\mathbf{E} \tag{3.17}$$

Making use of the relation between Fourier coefficients of $\mathbf{W}^*$ and those of $\frac{\partial \mathbf{W}^*}{\partial t}$ for N odd

$$\tilde{\mathbf{W}}_k^{*(1)} = ik\omega\tilde{\mathbf{W}}_k^*, k = -\frac{N-1}{2}, \ldots, 0, \ldots, \frac{N-1}{2} \tag{3.18}$$

where $\tilde{\mathbf{W}}_k^{*(1)}$ is the Fourier coefficient for mode of k of $\frac{\partial \mathbf{W}^*}{\partial t}$ while for N even, modification must be made so that

$$\tilde{\mathbf{W}}_k^{*(1)} = \begin{cases} ik\omega\tilde{\mathbf{W}}_k^*, & k = -\frac{N}{2}+1, \ldots, 0, \ldots, \frac{N}{2}-1 \\ 0, & k = -\frac{N}{2} \end{cases} \tag{3.19}$$

since Fourier mode of $k = \frac{N}{2}$ is not included in FFT and IFFT.

Denoting the operator for relation (3.18) or (3.19) as $\mathbf{N}_F$, the Fourier time spectral operator $\mathbf{D}_{t,F}$ can be expressed as

$$\mathbf{D}_{t,F} = \mathbf{E}^{-1}\mathbf{N}_F\mathbf{E} \tag{3.20}$$

Applying the Fourier Time Spectral Method and introducing pseudo-time derivative, the original unsteady governing equation in semi-discrete (3.1) form can be rewritten as

$$\Omega\frac{\partial \mathbf{W}^*}{\partial \tau} + \Omega\mathbf{E}^{-1}\mathbf{N}_F\mathbf{E}\mathbf{W}^* + \mathbf{R}(\mathbf{W}^*) = 0 \tag{3.21}$$

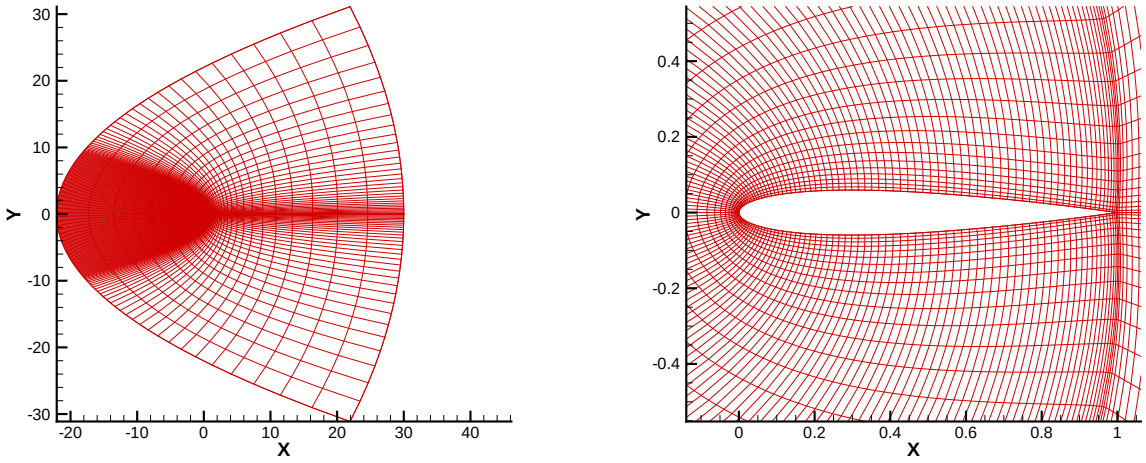
where τ is pseudo-time. Equation (3.21) can be solved by pseudo-time marching method until a steady-state solution is reached.

3.2 Test Cases for the Fourier Time Spectral Method

3.2.1 Inviscid Flow over a Pitching NACA0012 Airfoil

The unsteady flow over pitching airfoil has been studied extensively by experiments [30] and computations [25] [27] [31] [32] . It serves as a perfect external flow test cases for unsteady solvers. The present Fourier Time Spectral Method is tested with the inviscid flow over a NACA0012 airfoil pitching around its quarter-chord. Experimental data under turbulent flow conditions were provided in [30]. The pitching motion is given by the following equation:

$$\alpha(t) = \alpha_m + \alpha_0 sin\omega t \tag{3.22}$$



(a) 225×33 C-type mesh

(b) close view near surface

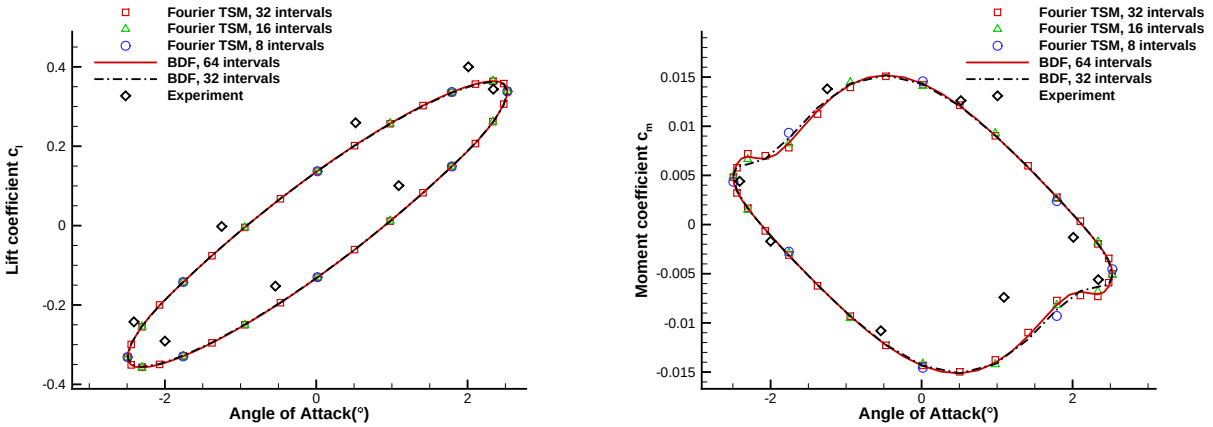
Figure 3.1: NACA0012 airfoil, body-fitted mesh.

where $\alpha(t)$ is the instantaneous angle of attack, α_0 the pitching range and $\alpha_m$the mean angle of attack. The angular frequency ω is related to the reduced frequency κ as

$$\kappa = \frac{\omega c}{2U_\infty} \tag{3.23}$$

where c is the chord of the airfoil and U_∞ is the free-steam speed. A 225×33 C-type mesh is generated by a conformal mapping grid generator. The mesh and its close view to the airfoil surface are shown in Figure 3.1. The AGARD CT5 case [30] is studied. The free stream Mach number is $M_\infty = 0.755$, $\alpha_m = 0.016°$, $\alpha_0 = 2.51°$ and $\kappa = 0.0814$. The flow Reynolds number is 5.5×10^6. All the above flow conditions are used in the present inviscid flow test case except for the Reynolds number since viscous effect is not considered here.

The computed lift and moment coefficients as a function of angle of attack are shown in Figure 3.2. Results using the Fourier time spectral method are compared with those using the Backward Difference Formula(BDF) as well as experimental results. It can be seen all computational results deviate the Experiment data systematically for the absolute value of lift coefficient. However the trend and the range of variation are about correct. Similar



(a) Lift coefficient

(b) Moment coefficient

Figure 3.2: Pitching NACA0012 airfoil, periodic inviscid flow, lift and moment coefficients versus angle of attack.

situation is reported by other computations [33]. The computational results for moment coefficient variation generally match the experimental data well. For both computational methods, results using different time resolution are provided. In the Fourier time spectral method, 8, 16 and 32 equal intervals in a period are used, while 32 and 64 intervals are used for the BDF method. For the time spectral method, even 8 intervals are sufficient to resolve the variation of force coefficient, especially for lift coefficient.

The variation of surface pressure coefficient distribution is studied in frequency domain through Fourier transformation. Results using time spectral method are compared with BDF method in Figures 3.3, 3.4, 3.5 and 3.6 for the time averaged component and the first three Fourier modes. Experimental data are also provided. For the BDF method, increasing the time resolution from 32 intervals to 64 does not make obvious difference. Only a slight deviation in the shock moving area can be found for the third mode. For the time spectral method, as time resolution increases from 8 intervals to 32 intervals, the solution converges to that of BDF with 64 intervals very quickly. In the shock moving area, approximately $0.2c < x < 0.6c$, 32 intervals are needed to resolve all three modes. Although the first three Fourier modes are retained in computation if 8 intervals are used in one period, truncation

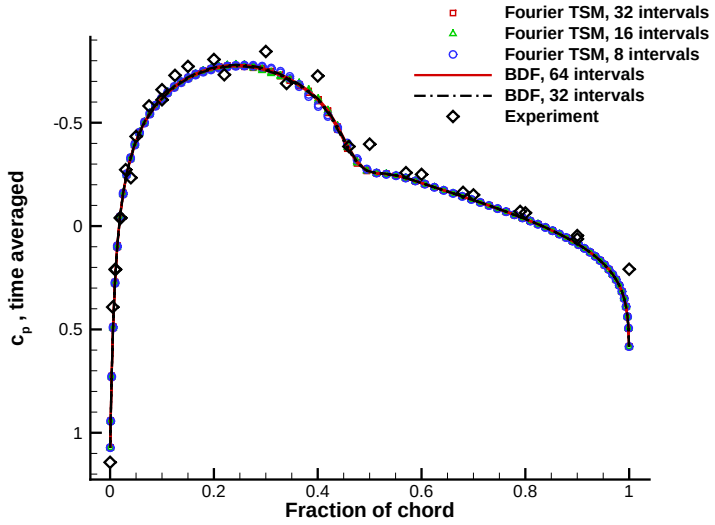
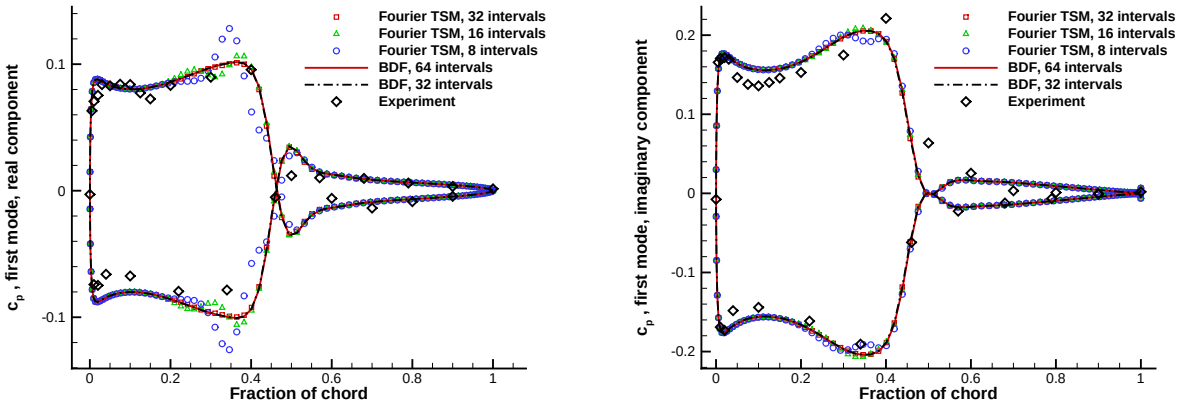


Figure 3.3: Pitching NACA0012 airfoil, inviscid flow, surface pressure coefficient, time averaged.

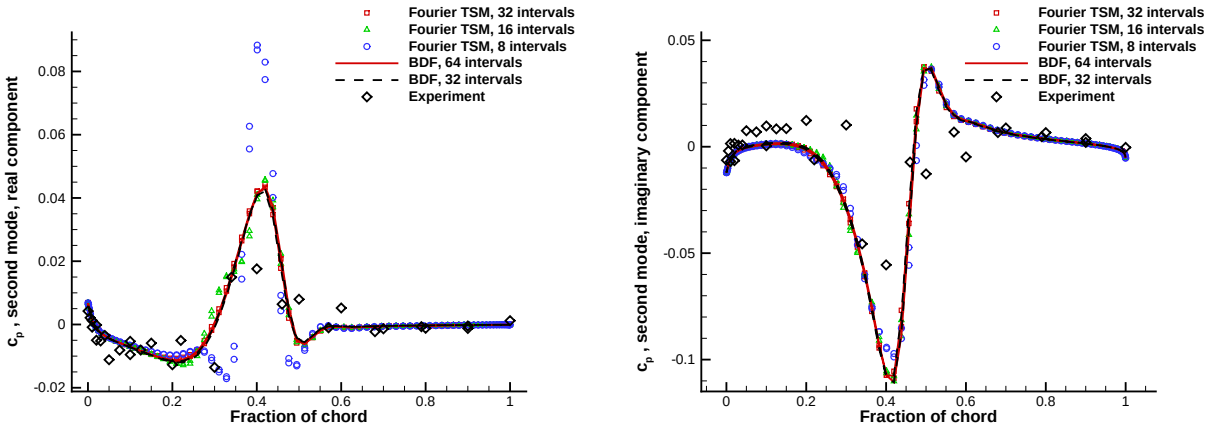
of the following non trivial modes would cause aliasing error to the retained modes. So it is not strange that the quality of the result is poor in the shock area even for the three retained modes using 8 intervals. But outside this region, even 8 intervals are sufficient to resolve all three modes. This is understandable since pressure experiences sudden increase across the shock wave, while the pressure distribution in the shock-free area is smooth. Computational results with sufficient resolution generally match experimental data well. This indicates that no separation occurs in the turbulent flow measured by the experiment. The viscous effect is confined in very narrow area close to the surface and in the wake. So the surface pressure distribution can be well predicted by solving the Euler equations.

In this test case, the time spectral method is validated by solving a periodic inviscid flow. If no discontinuity occurs, such as a shock wave, computational results using time spectral method can achieve satisfactory accuracy for both surface force distribution and integrated force coefficient with very coarse time resolutions, such as 8 intervals in a period. When phenomena like shock waves occur, 8 intervals in a period is still about enough to predict



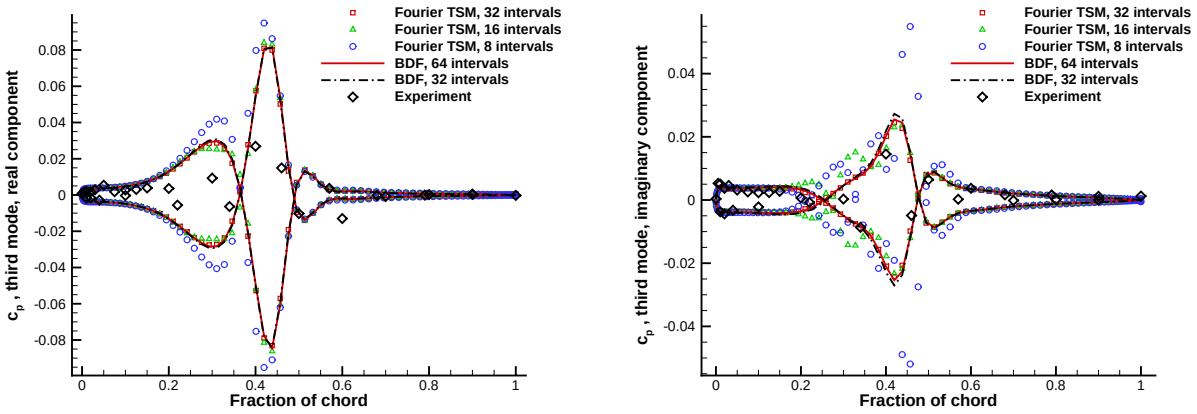
(a) real component (b) imaginary component

Figure 3.4: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, first mode.



(a) real component (b) imaginary component

Figure 3.5: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, second mode.



(a) real component

(b) imaginary component

Figure 3.6: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, third mode.

integrated forces. But to well predict surface force distribution across shock wave, more time levels are required.

3.2.2 Discussion on Nonsymmetric Numerical Solutions

Numerical experiments show that using odd number of intervals or even number of intervals does not make too much difference for non-symmetric problem. However, if the flow problem is symmetric, symmetric solution can only be guaranteed using even numbers of intervals. In the previous airfoil pitching problem, if the mean angle of attack α_m is set to zero, then the solution is expected to be symmetric. For surface pressure distribution, the part of the upper surface should be exactly the same or mirror image of the lower surface part. This is because the flow conditions, the forced motion and the geometry are all symmetric. Computational results using odd number of intervals and even number of intervals are compared for this theoretically symmetric problem.

The Fourier transformed surface pressure coefficient distribution is shown in Figures 3.7-3.10 for time spectral method with 7 intervals and 8 intervals in a period. The results are

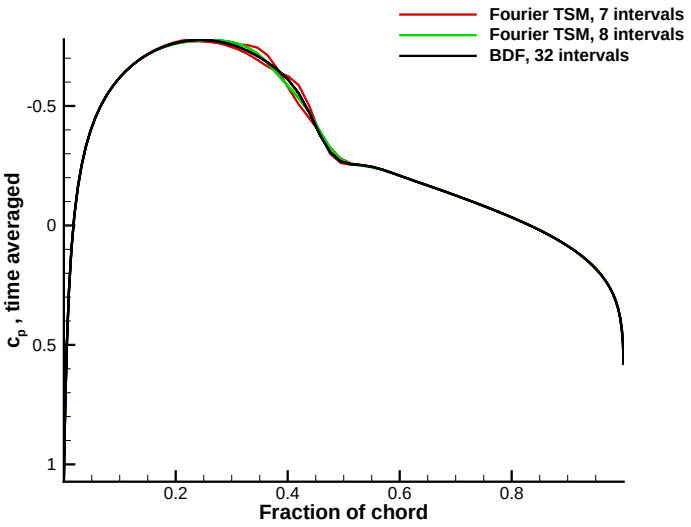
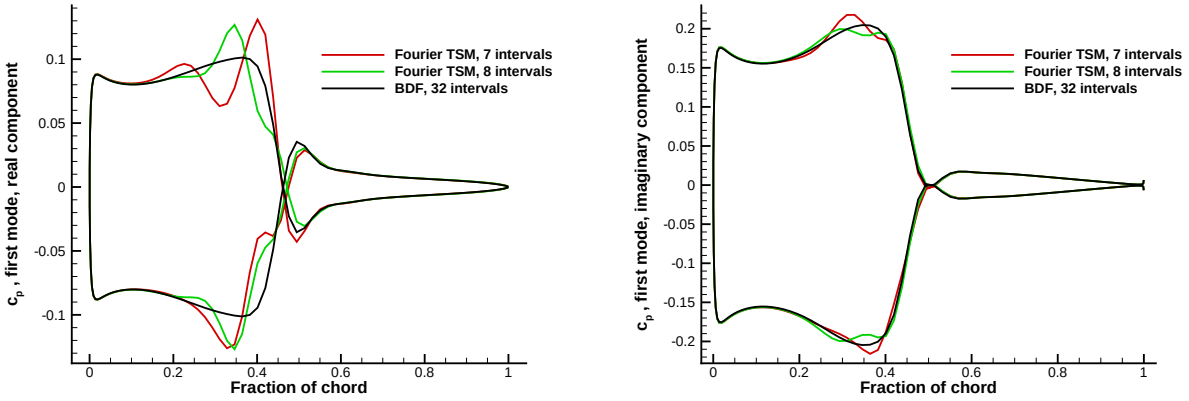


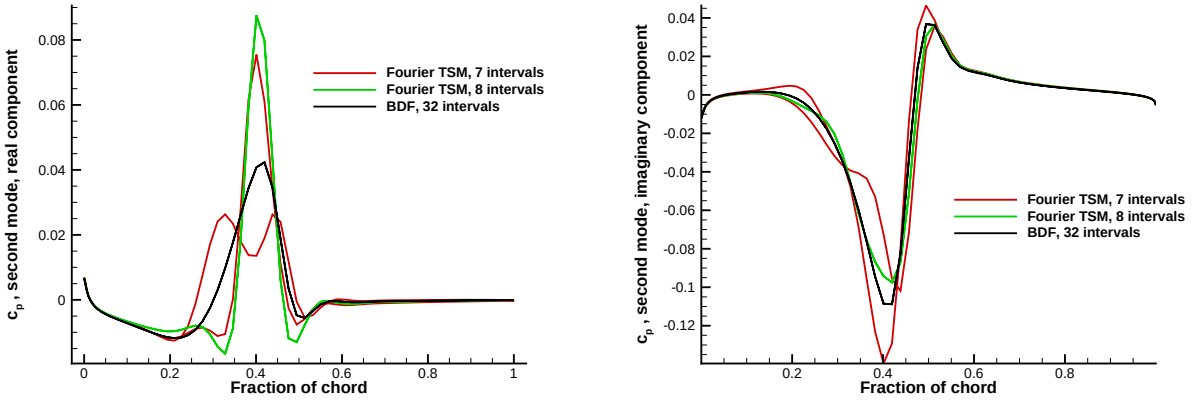
Figure 3.7: Pitching NACA0012 airfoil, inviscid flow, surface pressure coefficient, time averaged.



(a) real component

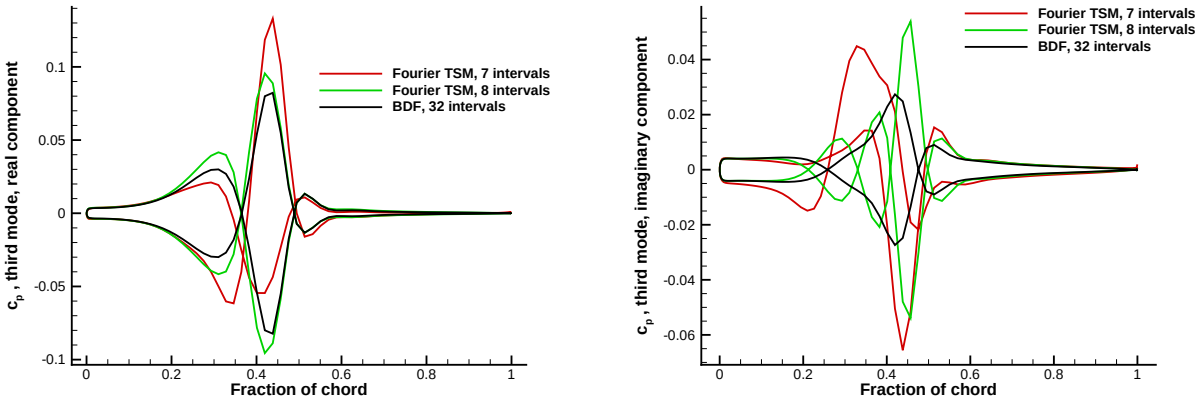
(b) imaginary component

Figure 3.8: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, first mode.



(a) real component (b) imaginary component

Figure 3.9: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, second mode.



(a) real component (b) imaginary component

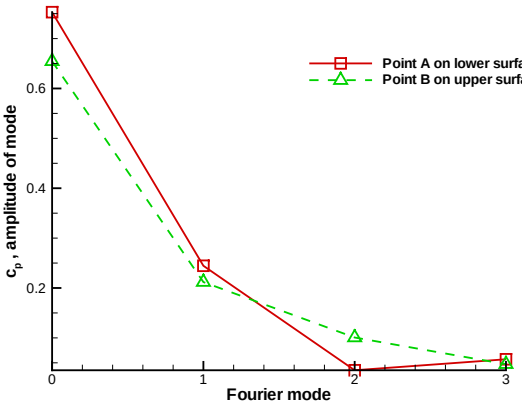
Figure 3.10: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, third mode.

compared with that using the BDF method with 32 intervals. Based on the present time spectral method, 7 or 8 intervals all correspond to retaining the first three Fourier modes in computation. So computation using either one has the same resolution in frequency domain. However, it can be found that if 7 intervals are used in a period, symmetry in solution is lost. In the computation using 8 intervals, the solution is still symmetric just as the situation of using 32 intervals in the BDF method. Actually this problem is not unique to time spectral method. Numerical experiment can show it also happens to BDF method. This phenomena indicates that to ensure symmetric solution for a symmetric problem, a period must be split in such a way that every time level should be able to find another one that has a 180° phase difference away from it. Obviously, this requirement can be fulfilled using even numbers of intervals, whereas employing odd number of intervals can not satisfy it. This problem is then studied on local points in different flow area. Solution on two pairs of symmetric surface points are analyzed. One pair of points are point A on lower surface and point B on upper surface. They are about $0.4c$ away from the leading edge and hence in the shock wave moving area. Points C on lower surface and D on upper surface are $0.7c$ away from leading edge, a location in shock-free area. The pressure spectra calculated using 7, 11, 15 and 21 intervals in a period for the time spectral method are shown in Figure 3.11. It can be observed that since odd numbers of intervals are used, the pressure spectrum on point A deviates dramatically from that on point B, which should not occur theoretically. As time resolution increases, the deviation becomes smaller. Similar results can be observed in Figure 3.12 for points C and D. The deviation between points C and D is generally smaller than that between points A and B. This suggests the problem of losing symmetry in solution becomes worse in shock wave moving area than in shock-free area when using odd numbers of intervals. This conclusion is further supported with the fine pressure spectrum on points A and C shown in Figure 3.13. The fine pressure spectrum is obtained from computation with 64 intervals using time spectral method. Compared to the spectrum on point A, the one on point C has much less non-trivial high frequency modes. So on point C, when time resolution improves

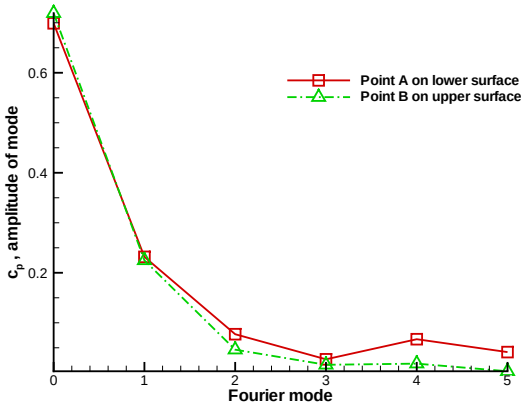
a little, the solution recovers to symmetry very quickly. That is why when 9 intervals are used, no visible deviation can be observed for pressure spectrum between points C and D. Though increasing time resolution could improve symmetry in computational results using odd number of intervals, even number of intervals in a period is adopted in this dissertation for the following reason. The time resolution in practical use for time spectral method is usually low for the sake of saving computational efforts. If an odd number of intervals is used, non-symmetric solution will be produced for a symmetric problem. Increasing time resolution is usually not affordable in practice. So to avoid fake non-symmetric solution, even number of intervals is the best choice. However, stability problems using even number of intervals may occur as reported in [19]. Despite this fact, even number of intervals is still adopted in this dissertation as long as computation using it can converge normally.

3.2.3 Vortex Shedding Flow over a Circular Cylinder

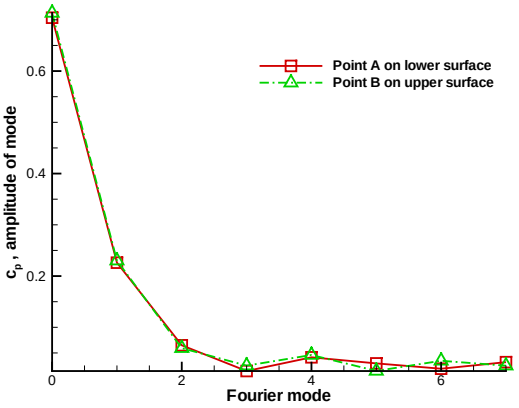
The Fourier time spectral method is then applied to the laminar vortex shedding flow over circular cylinder. Being different from the preceding unsteady flow over a pitching airfoil, the vortex shedding frequency is not known a priori. Only a very roughly estimated value of the vortex shedding frequency can be given by the experimental fact that the Strouhal number $St = \frac{fD}{U} \approx 0.2$ for Reynolds numbers from 100 to 10^5 [34]. To precisely measure the frequency of a specific vortex shedding flow, computation or experiment must be conducted. Unlike the dual-time stepping based on BDF method, the Fourier time spectral method uses the flow frequency as an explicit parameter since it appears explicitly in the approximated time derivative term. This indicates a frequency searching process must be involved in the computation. McMullen et al.[15] proposed a Gradient Based Variable Time Period (GBVTP) method for the Frequency Domain method. Gopinath et al. [19] proposed a similar gradient based method to the time spectral method. This type of method is based on knowing the following fact. If the frequency is exact, the residuals of the unsteady N-



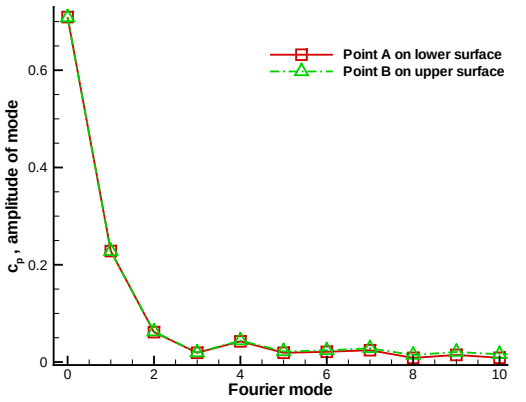
(a) 7 intervals



(b) 11 intervals

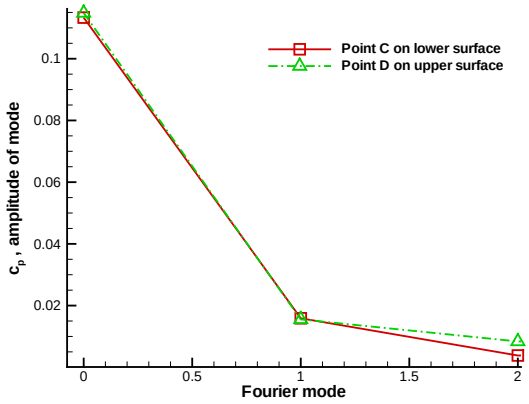


(c) 15 intervals

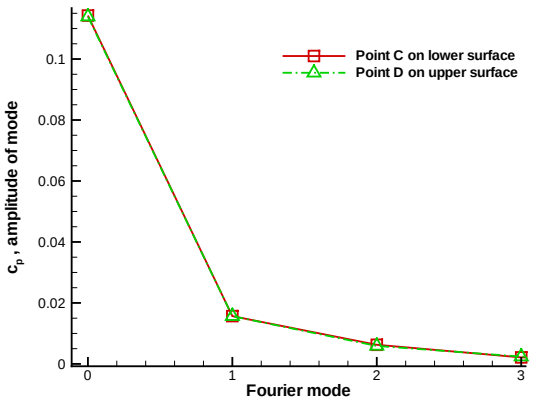


(d) 21 intervals

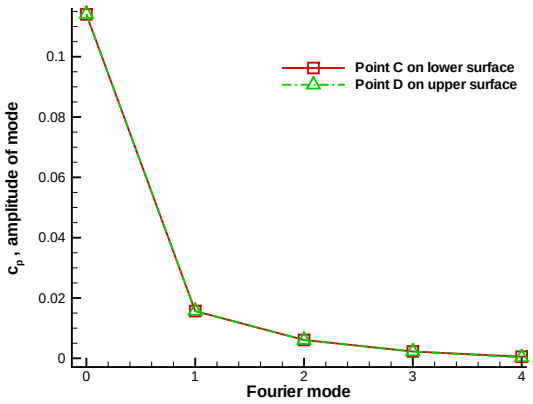
Figure 3.11: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient spectrum, surface points A and B.



(a) 5 intervals

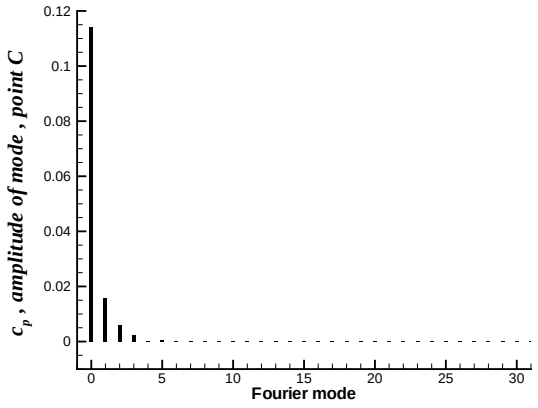


(b) 7 intervals

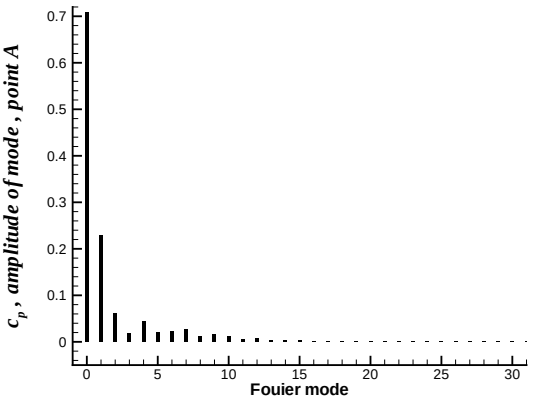


(c) 9 intervals

Figure 3.12: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient spectrum, surface points C and D.



(a) Point C



(b) Point A

Figure 3.13: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, fine spectrum, Point A and C.

S equation could be reduced to machine zero in computation. For different guess of the frequency, the residuals will converge to different levels of accuracy. So the problem becomes an optimization problem of finding exact frequency so as to make unsteady residuals zero. The gradient based method uses the negative gradient of residuals with respect to frequency to find the exact frequency gradually. In the unsteady Navier-Stokes equations that are discretized by the Fourier time spectral method in time, the gradient of the residual with respect to frequency can be calculated as

$$\frac{\partial \mathbf{R}_{i,j,n}^2}{\partial \omega} = 2\mathbf{R}_{i,j,n} \times \frac{V}{\omega} D_{t,n} \mathbf{W}^* \tag{3.24}$$

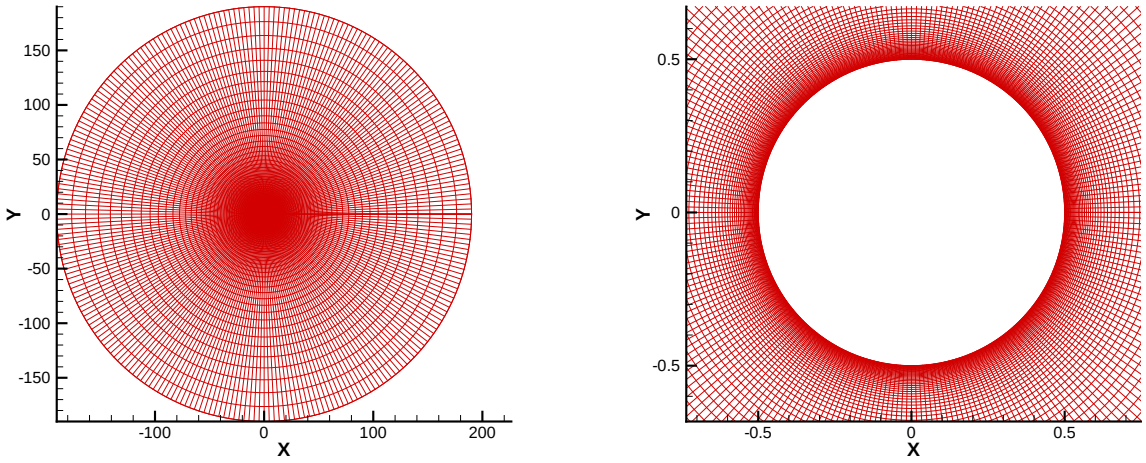
at real time level n where $D_{t,n}$ is nth row of the Fourier time spectral operator D_t and $\mathbf{R}_{i,j,n}$ is the unsteady residual at point (i, j) at time level n. Then this gradient is averaged over all time levels on all control volumes. The result of the averaging is denoted as $\frac{\partial \mathbf{R}^2}{\partial \omega}$ and is used to update frequency ω as follows

$$\omega^{l+1} = \omega^l - \alpha \frac{\partial \mathbf{R}^2}{\partial \omega} \tag{3.25}$$

where α is a suitable step for updating frequency ω and it has to be carefully chosen to guarantee convergence. The advantages of this method are that the concept behind it is clear and the method is easy to implement. If the search is performed in the correct range, the method can find the exact frequency very precisely. However, there is one well-known drawback that the initial guess of the frequency must be close to the exact value since the gradient based method is always locally optimized. To broaden the search range and make the initial guess of frequency less constrained, another approach is proposed in this dissertation to be used in combination with the gradient based method.

The new approach is based on the following reasoning. Usually for unsteady problems like the vortex shedding flows, the first Fourier mode of lift coefficient has the maximum amplitude. This suggests the frequency can be updated by finding out the Fourier mode of maximum amplitude of the temporary lift coefficient variation. If the Fourier mode of the maximum amplitude is not the first mode, then its frequency is set to be the new frequency of the flow. Furthermore, the Fourier modes with lower frequencies than this new flow frequency can be filtered out from the flow field. The filtered flow field is a closer match to the exact unsteady flow field and it serves as the new initial flow field for computation based on the newly updated frequency.

This new approach may help to enable the frequency search process from a initial guess of frequency far from exact value. However, it cannot be applied in the final phase of frequency search process. This is because frequency can only be updated as integer multiples of the old value (spectrum of lift coefficient is discrete) and the final frequency change is usually very small. So in the frequency search process, the newly proposed approach should be performed before the gradient based method. This new frequency search method is tested along with the Fourier time spectral method on the laminar vortex shedding flow at Reynolds number of 180. This test case is well studied by experiments [35] [36] [37] [38] and computations [15] [19]. The Mach number is set to be 0.2 and the flow is essentially incompressible. However



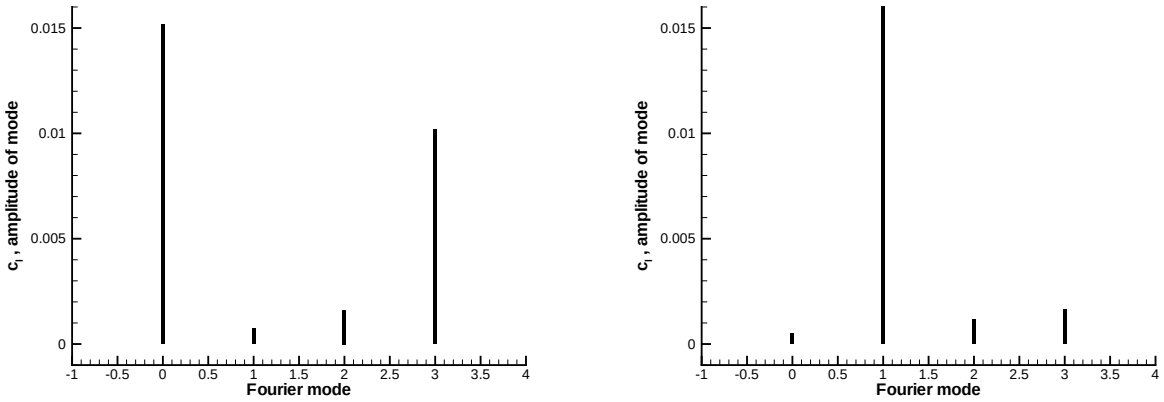
(a) 257×129 O-type mesh

(b) close view near surface

Figure 3.14: Circular cylinder, body-fitted mesh.

the present solver for compressible flow can still be used. A 257×129 O-type mesh is used and shown in Figure 3.14. The farfield boundary of the mesh extends about 200 diameters away from the cylinder surface. The normal distance from the first grid point to the surface is 0.001 for this laminar flow since the unsteadiness in the boundary layer need to be well resolved.

The initial guess of the reduced frequency is set to be 0.05 which is far away from the correct one, about 0.6. In the frequency search process, the analysis of the Fourier mode of lift coefficient is performed at the end of every 100 multigrid cycles. The spectrum of lift coefficient after 100 and 200 multigrid cycles is shown in Figure 3.15. It can be seen that at the end of 100 multigrid cycles, the third mode of the lift coefficient gets maximum amplitude. But since the aliasing error of the Fourier mode with highest frequency is usually big, then the frequency of the second mode is set to be the new flow frequency. The flow field is filtered accordingly. At the end of 200 multigrid cycles, the first mode of the lift coefficient achieves maximum amplitude. So the newly proposed approach is turned off. From this point on, the frequency is updated by the gradient based method. The whole



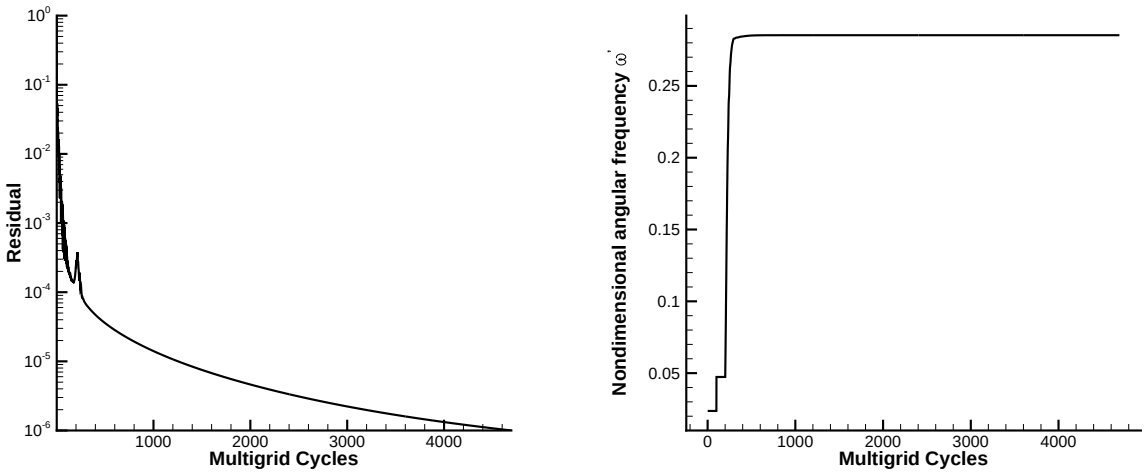
(a) after 100 multigrid cycles

(b) after 200 multigrid cycles

Figure 3.15: Circular cylinder, vortex shedding flow, spectrum of lift coefficient c_l.

frequency updating history as well as the flow field residual history with respect to multigrid cycles are shown in Figure 3.16. It can be observed that finally the exact frequency is found and the residual of flow field can drop to 10^{-6}. Based on the final searched frequency, the Strouhal number of present computation is 0.1919 and it matches Williamson's experimental results[35] [36] very well. The time-averaged total drag coefficient (including the component due to pressure and the one due to skin friction) is 1.337, which is very close to Henderson's experimental results [38].

The lift coefficient history on all real time levels with respect to multigrid cycles are shown in Figure 3.17. Before 100 multigrid cycles, there is no visible difference in the lift coefficient history between different real time levels. This is due to the initial flow field on all time levels are set to be the same for the sake of simplicity. The unsteadiness is triggered by making the velocity in the upper half flow field slightly different from that in the lower half. From 100 to 200 multigrid cycles, lift coefficient on different time levels begins to become different. The phase of the lift coefficient starts to drift. This is because the unsteadiness has been triggered already and the temporarily searched frequency is still not correct. Between 200 and 300 multigrid cycles, the phase drifting gradually slows down. This indicates the temporarily searched frequency is close to the exact one. After 300 multigrid cycles, the



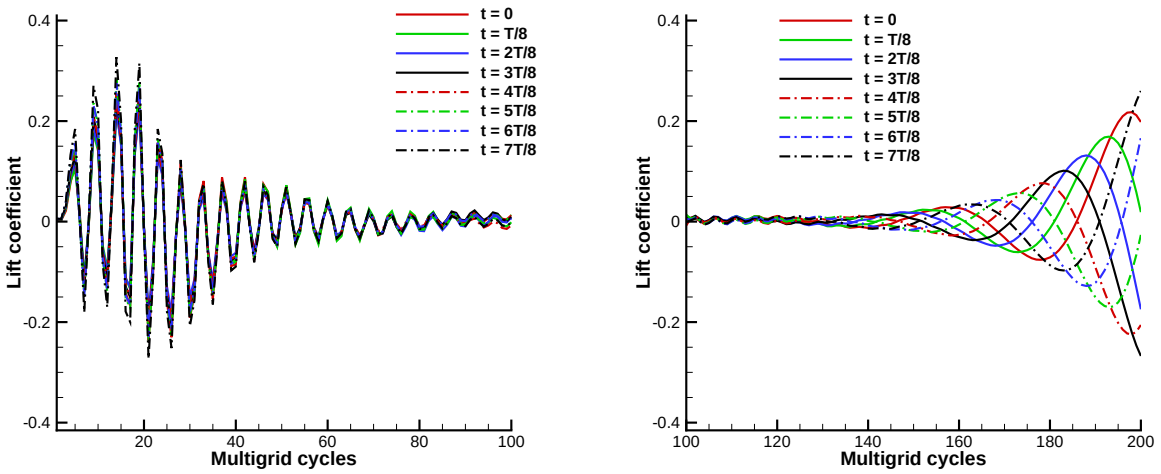
(a) residual (b) nondimensional angular frequency ω_r

Figure 3.16: Circular cylinder, vortex shedding flow, convergence history versus multigrid cycles.

phase drifting completely stops. Both the frequency and flow field are all close to the exact solutions. Frequency updating and flow field refinement do not stop until the flow field convergence criterion of 1×10^{-6} is reached.

3.3 The Chebyshev Time Spectral Method and its Derivation

The Chebyshev time spectral method aims at solving non-periodic problems. Theoretically speaking, Fourier time spectral method can only be applied to non-periodic problems when the problems can be regarded as periodic ones with infinity large period. In the theories of approximating functions, orthogonal polynomial series are frequently used to approximate solutions in non-periodic problems. Functions that are square integrable can be expanded with a set of orthogonal polynomial basis. The basis functions can be conveniently generated using the Sturm-Liouville theorem, which concerns the eigenfunctions u of the eigenvalue



(a) $1 - 100$ multigrid cycles

(b) $100 - 200$ multigrid cycles

(c) $200 - 300$ multigrid cycles

(d) $300 - 5000$ multigrid cycles

Figure 3.17: Circular cylinder, vortex shedding flow, lift coefficient history versus multigrid cycles.

problem known as Sturm-Liouville problem:

$$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right)+q(x)u=\lambda\omega(x)u, a<x<b \tag{3.26}$$

Chebyshev polynomials are known as one set of orthogonal eigenfunctions of the following singular(when $p(a)=p(b)=0$) Sturm-Liouville problem in interval $[-1,1]$

$$-\frac{d}{dx}\left((1-x)^{1+\alpha}(1+x)^{1+\beta}\frac{du}{dx}\right)=\lambda(1-x)^{\alpha}(1+x)^{\beta}u, a<x<b \tag{3.27}$$

for $\alpha=\beta=-\frac{1}{2}$. They are employed in this present work for solving non-periodic problems for the following reasons. Firstly, Chebyshev polynomials have well-known nice approximation properties. Secondly, Chebyshev series using these polynomials can be evaluated efficiently by Fast Fourier Transform (FFT). The trigonometric definition of Chebyshev polynomials can be written as

$$T_k(x)=\mathbf{cos}(k\mathbf{arccos}(x)), -1\leq x\leq 1 \tag{3.28}$$

Obviously, when $k=0$ and $k=1$, the Chebyshev polynomial of degree zero and one are

$$T_0(x)=1, T_1(x)=x \tag{3.29}$$

Using the recurrence relation

$$T_{n+1}(x)=2xT_{n+1}(x)-T_{n-1}(x) \tag{3.30}$$

the Chebyshev polynomials of higher degree written in algebraic polynomial form can be easily obtained. Some of them are shown in Figure 3.18.

With a complete set of Chebyshev polynomials, a square integrable function $f(x)$ can be

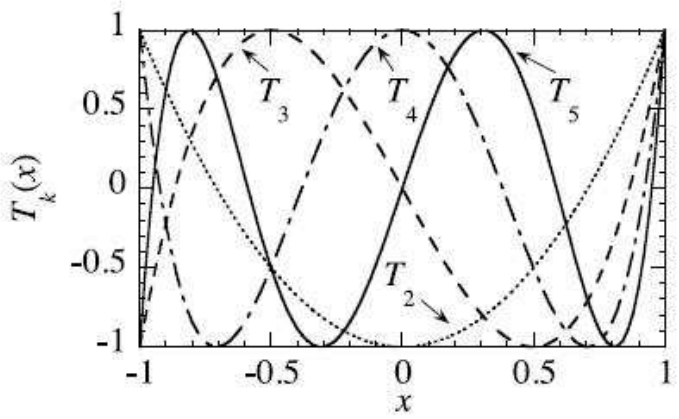


Figure 3.18: The Chebyshev polynomials of degree two through five[23].

expanded using Chebyshev series as

$$f(x) = \sum_{k=0}^{\infty} \tilde{f}_k T_k(x) \tag{3.31}$$

Note the fact that the Chebyshev polynomials are a set of orthogonal basis in square integrable function space and the orthogonality is expressed in inner product form as

$$\int_{-1}^{1} T_k(x) T_l(x) \omega(x) dx = \begin{cases} 0, & l \neq k \\ c_k \frac{\pi}{2}, & l = k \end{cases} \tag{3.32}$$

where $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ is the weight function associated with Chebyshev polynomials and c_k is given as

$$c_k = \begin{cases} 2, & k = 0 \\ 1, & k \geq 1 \end{cases} \tag{3.33}$$

Based on the above relations, the series coefficients $\tilde{f}_k$ can be formulated as

$$\tilde{f}_k = \frac{\int_{-1}^{1} f(x) T_k(x) \omega(x) dx}{\int_{-1}^{1} T_k^2(x) \omega(x) dx} = \frac{2}{\pi c_k} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx \tag{3.34}$$

In equation (3.31), if polynomials up to degree N are retained, then a discrete Chebyshev transformation pair can be established as

$$\begin{aligned} f_n &= \sum_{k=0}^{N} \tilde{f}_k T_k(x_n), 0 \leq n \leq N \\ \tilde{f}_k &= \frac{\sum_{n=0}^{N} f(x_n) T_k(x_n) \omega_n}{\sum_{n=0}^{N} T_k^2(x_n) \omega_n}, 0 \leq k \leq N \end{aligned} \tag{3.35}$$

where $f_n = f(x_n)$ and $\tilde{f}_k$ are the Chebyshev coefficients in the discrete transform. ω_n are the weight values for a given quadrature at quadrature points x_n. Various quadrature rules can be used to numerically calculate the inner product integrals in equation (3.34). In present work, the Gauss-Lobatto quardrature rule(covering endpoints, $x = \pm 1$) for Chebyshev weighted integrals is employed. The corresponding quadrature points x_n and weights ω_n are given as

$$\begin{aligned} x_n &= \mathbf{cos}(\frac{n\pi}{N}), n = 0, 1, \ldots, N, \\ \omega_n &= \begin{cases} \frac{\pi}{2N}, & n = 0, N \\ \frac{\pi}{N}, & n = 1, 2, \ldots, N-1 \end{cases} \end{aligned} \tag{3.36}$$

When the Chebyshev polynomials are evaluated at the above quadrature points

$$T_k(x_n) = \mathbf{cos}(\frac{kn\pi}{N}) \tag{3.37}$$

substitute equations (3.36) and (3.37) into equation (3.35), the Discrete Chebyshev Transform reduces at Gauss-Lobatto points to Discrete Cosine Transform[23]

$$\begin{aligned} \tilde{f}_k &= \frac{2}{N\bar{c}_k} \sum_{n=0}^{N} \frac{f_n}{\bar{c}_n} \mathbf{cos}(\frac{kn\pi}{N}), k = 0, 1, \ldots, N, \\ f_n &= \sum_{k=0}^{N} \tilde{f}_k \mathbf{cos}(\frac{kn\pi}{N}), n = 0, 1, \ldots, N \end{aligned} \tag{3.38}$$

where

$$\bar{c}_l = \begin{cases} 2, & l = 0, N, \\ 1, & l = 1, 2, \ldots, N-1 \end{cases} \tag{3.39}$$

It is well known that if a sequence $\{f_n\}_{n=0}^{N}$ is even, which corresponds to a even function $f(x)$, then the Fast Fourier Transform on it reduces to the Discrete Cosine Transform. For a sequence that does not satisfy this special condition, the two transforms can still be made equivalent if the original sequence $\{f_n\}_{n=0}^{N}$ is extended to a new one of length $2N$ [23], $\{\bar{f}_n\}_{n=0}^{2N-1}$, as follows:

$$\begin{cases} \bar{f}_n = f_n, & n = 0, 1, \ldots, N, \\ \bar{f}_{2N-n} = f_n, & n = 1, 2, \ldots, N-1 \end{cases} \tag{3.40}$$

Suppose the derivative function $f'(x)$ of function $f(x)$ is also square integrable, then it can be expanded as a Chebyshev series in the similar way of equation (3.31)

$$f'(x) = \sum_{k=0}^{\infty} \tilde{f}_k^{(1)} T_k(x) \tag{3.41}$$

Then a recursion that relates the coefficients of the derivative $\tilde{f}_k^{(1)}$ to the the coefficients of the original function $\tilde{f}_k$ can be established as

$$c_k \tilde{f}_k^{(1)} = \tilde{f}_{k+2}^{(1)} + 2(k+1)\tilde{f}_{k+1}, k \geq 0. \tag{3.42}$$

where c_k is given in equation (3.33). when the discrete Chebyshev transform is applied to the derivative function $f'(x)$, special treatments must be added, then the Chebyshev coefficients

of the derivative should be calculated based on those of the original function as

$$\begin{aligned}
\tilde{f}_N^{(1)} &= 0 \\
\tilde{f}_{N-1}^{(1)} &= 2N\tilde{f}_N \\
\tilde{f}_k^{(1)} &= \tilde{f}_{k+2}^{(1)} + 2(k+1)\tilde{f}_{k+1}, k = N-2, \ldots, 1 \\
\tilde{f}_0^{(1)} &= \frac{1}{2}\tilde{f}_2^{(1)} + \tilde{f}_1
\end{aligned} \tag{3.43}$$

This is because in the discrete Chebyshev transforms for a function and its derivative with $N+1$ quadrature points, coefficients for Chebyshev polynomials of degree higher than N are not involved, hence

$$\tilde{f}_{N+1}^{(1)} = \tilde{f}_{N+2}^{(1)} = \tilde{f}^{N+1} = 0 \tag{3.44}$$

The key issue of developing Time Spectral Method based on Chebyshev expansion is to approximate the real time derivative by Chebyshev time spectral operator. Firstly, the Discrete Chebyshev Transform pair for solutions on $N+1$ real time levels in an unsteady flow problem can be written as

$$\mathbf{W}^* = \mathbf{C}\tilde{\mathbf{W}} \tag{3.45}$$

where $\mathbf{W}^* = [\mathbf{W}_{t_0}, \ldots, \mathbf{W}_{t_N}]$, $\mathbf{C}$ is the forward Discrete Chebyshev Transform operator and $\tilde{\mathbf{W}} = [\tilde{\mathbf{W}}_0, \ldots, \tilde{\mathbf{W}}_N]$. Then the Chebyshev coefficients $\tilde{\mathbf{W}}^{(1)}$ for the real time derivative of solution on all $N+1$ real time levels can be calculated as

$$\tilde{\mathbf{W}}^{(1)} = \mathbf{N}_C\tilde{\mathbf{W}} \tag{3.46}$$

where operator $\mathbf{N}_C$ is given by equation (3.43). Next, perform backward Discrete Chebyshev Transform to $\tilde{\mathbf{W}}^{(1)}$

$$(\frac{d\mathbf{W}}{dt})^* = \mathbf{C}^{-1}\tilde{\mathbf{W}}^{(1)} \tag{3.47}$$

where $\mathbf{C}^{-1}$ is the backward Discrete Chebyshev Transform operator and the extended vector $(\frac{d\mathbf{W}}{dt})^* = [\frac{d\mathbf{W}}{dt}|_{t_0}, \ldots, \frac{d\mathbf{W}}{dt}|_{t_N}]^T$. So in Chebyshev Time Spectral Method, the operator

approximating real time derivative term can be given as

$$\mathbf{D}_{t,C} = \mathbf{C}^{-1}\mathbf{N}_C\mathbf{C} \tag{3.48}$$

The above derivations are all based on the interval $[-1, 1]$. To make operator in equation (3.48) work for any time intervals $t \in [a, b]$, modification must be made on it. Replace the time variable t in equation (3.45) to (3.48) with ξ, then the linear transform between t and ξ is given by

$$t = a + \frac{b-a}{1-(-1)}(\xi - (-1)) \tag{3.49}$$

Next the partial derivative with respect to t can be calculated using that with respect to ξ as

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial t} = \frac{2}{b-a}\frac{\partial}{\partial \xi} \tag{3.50}$$

Finally, the Chebyshev time spectral operator working for any time interval $t \in [a, b]$ can be formulated as

$$\mathbf{D}_{t,C} = \frac{2}{b-a}\mathbf{C}^{-1}\mathbf{N}_C\mathbf{C} \tag{3.51}$$

Applying Chebyshev Time Spectral Method to unsteady computations, the discretized governing equation can be modified as

$$\Omega\frac{\partial \mathbf{W}^*}{\partial \tau} + \Omega\frac{2}{b-a}\mathbf{C}^{-1}\mathbf{N}_C\mathbf{C}\mathbf{W}^* + \mathbf{R}(\mathbf{W}^*) = \mathbf{0} \tag{3.52}$$

Since pseudo-time derivative is added, the above equation can solved by pseudo-time marching method until a steady-state solution is obtained.

3.4 Test Cases for the Chebyshev Time Spectral Method

3.4.1 Numerical Solutions to the Viscous Burger's Equation

The Chebyshev time spectral method is firstly tested by solving the following viscous Burgers equation

$$\frac{\partial w}{\partial t} + w\frac{\partial w}{\partial x} = \mu\frac{\partial^2 w}{\partial x^2} \tag{3.53}$$

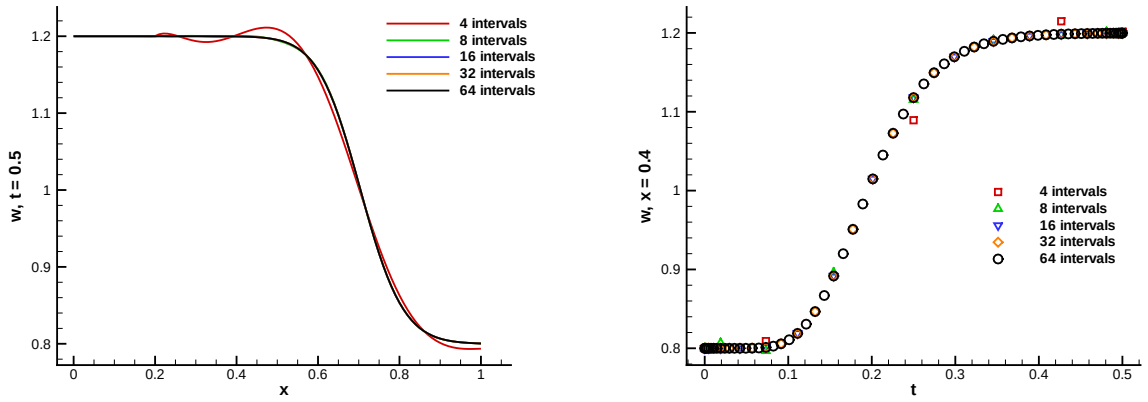
where the viscous coefficient $\mu = 0.015$. The equation is solved in $x \in [0, 1]$ and time domain $t \in [0, 0.5]$. The initial value is specified as

$$w(x, 0) = \begin{cases} 1.2, & x \leq 0.2 \\ 0.8, & x > 0.2 \end{cases} \tag{3.54}$$

and the condition on the left boundary is set as $w(0, t) = 1.2$.

To numerically solve the above problem, a 5-stage explicit Runge-Kutta solver is used. The real time derivative term is approximated by Chebyshev time spectral operator. Pseudo-time derivative is added so that a steady-state solution can be obtained by marching pseudo-time forward. The convergence criteria is set to be 1×10^{-4}.

The details of the converged solution, such as the final solution at $t = 0.5$ and solution history at $x = 0.4$, are shown in Figure 3.19. At $t = 0.5$, the theoretical solution to the inviscid Burgers' equation should be a discontinuity located at $x = 0.7$. When the viscous effect is included, the discontinuity reduces to a gradual jump. However the location of the jump is the same as the discontinuity. This location is precisely captured in the present computational results. As time resolution increases, solution converges to the one with highest time resolution. Actually, there is no visible difference between the result using 8 intervals and the one using 64 intervals. The spatial point $x = 0.4$ should be swept by



(a) $w(x)$ at $t = 0.5$ (b) $w(t)$ at $x = 0.4$

Figure 3.19: Viscous Burgers'equation, solution details.

the moving jump at $t = 0.2$. Computational results using all time resolutions capture this solution feature very well. Difference between solution using 8 intervals and those using higher time resolution can be found near $t = 0$. This suggests higher time resolution should be adopted to well predict local solution history if the local point is swept by a discontinuity.

Finally the error analysis using different time resolution is conducted. The solution using 64 intervals serves as reference solution. Then the error is calculated firstly by comparing the Chebyshev transformation coefficients between a given solution and the reference solution and then averaging the results over entire spatial area. The result is shown in Figure 3.20. It can be seen that the solution error drops faster than any algebraic order method. The error drop rate of a second-order method is also provided as reference for comparison. Computational results of this test case show that the Chebyshev time spectral method is capable of solving non-periodic problems even in those where discontinuity occurs. The result of error analysis show that the Chebyshev time spectral method may achieve high efficiency since it provides higher accuracy of approximation for time derivatives.

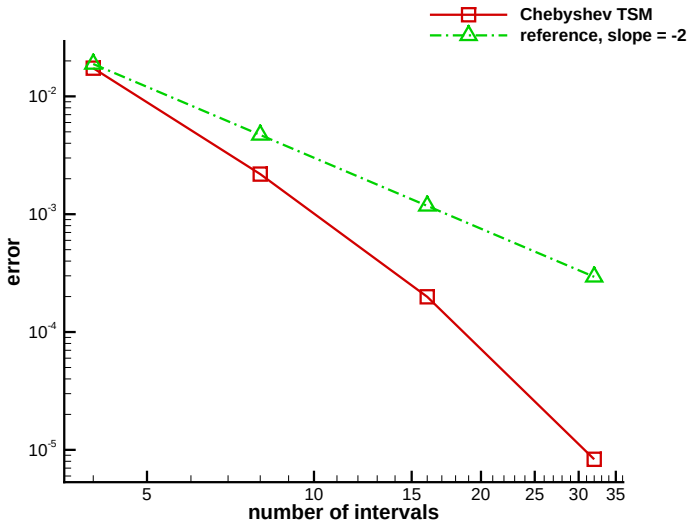
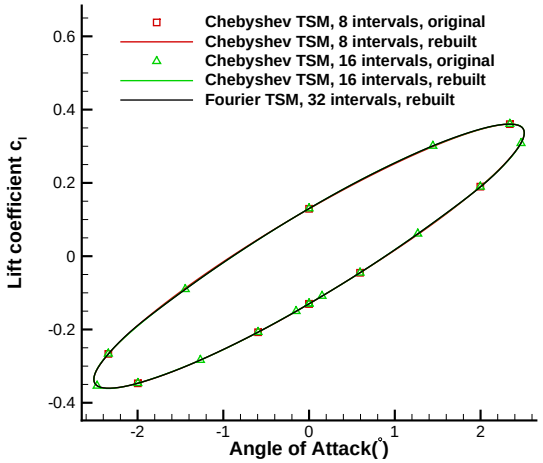


Figure 3.20: Viscous Burgers'equation, error analysis.

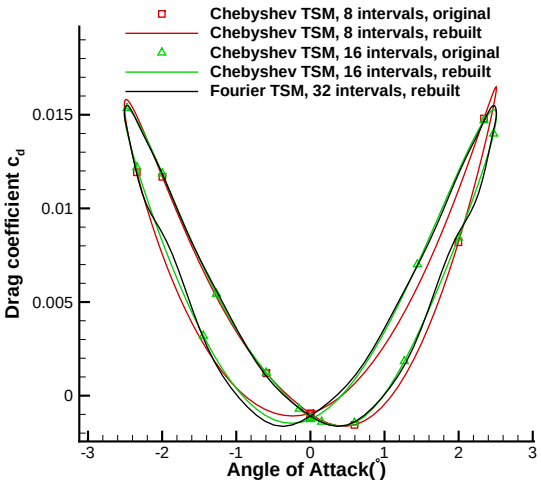
3.4.2 Periodic Inviscid Flow over Pitching NACA0012 Airfoil

The Chebyshev time spectral method is able to solve non-periodic problems. There is no doubt that it should have the ability to solve periodic problems if periodic boundary condition is properly applied in the time domain. The periodic inviscid flow over a pitching NACA0012 airfoil can be used for verification. In this test case, the mean angle of attack α_m is still set to 0, so the problem is symmetric. Computations using 8 and 16 intervals can both reach convergence criteria of 1×10^{-14}. Results using Fourier time spectral method with 32 intervals are also provided as the exact solution.

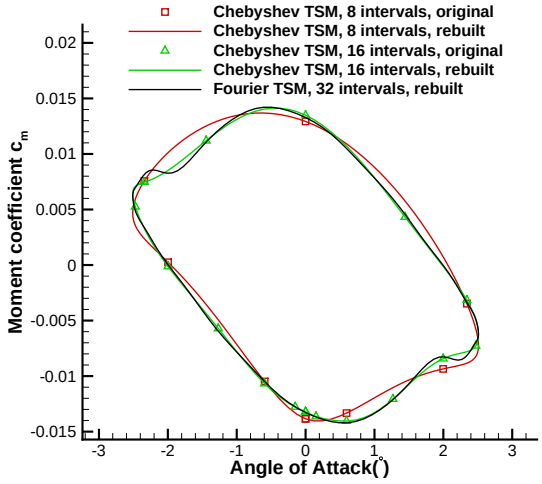
Variation of integrated force coefficients versus angle of attack is shown in Figure 3.21. For both Chebyshev and Fourier time spectral method, results rebuilt at a much finer time resolution, 128 intervals in a period, are provided as well. These rebuilt results are obtained using the Fourier modes or Chebyshev polynomials that the original resolution can retain. A rebuilt result is completely coincident with the computed result that it is rebuilt from at the original real time levels. Obvious non-symmetric solution can be observed for drag



(a) lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 3.21: Pitching NACA0012 airfoil, periodic inviscid flow, force coefficient versus angle of attack.

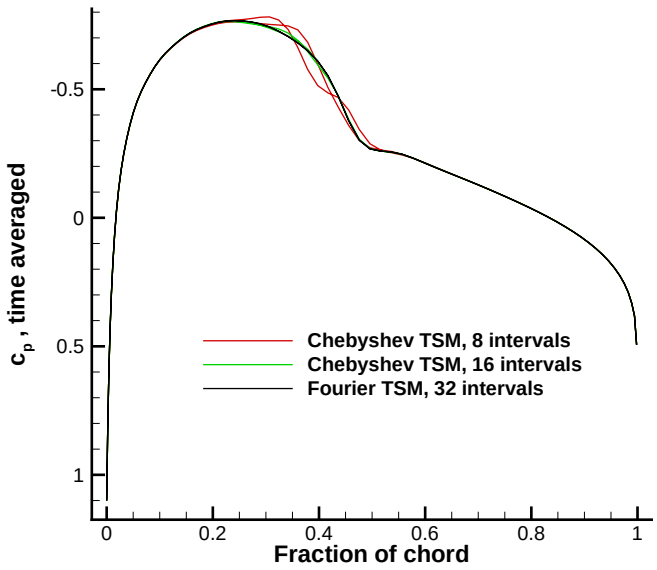
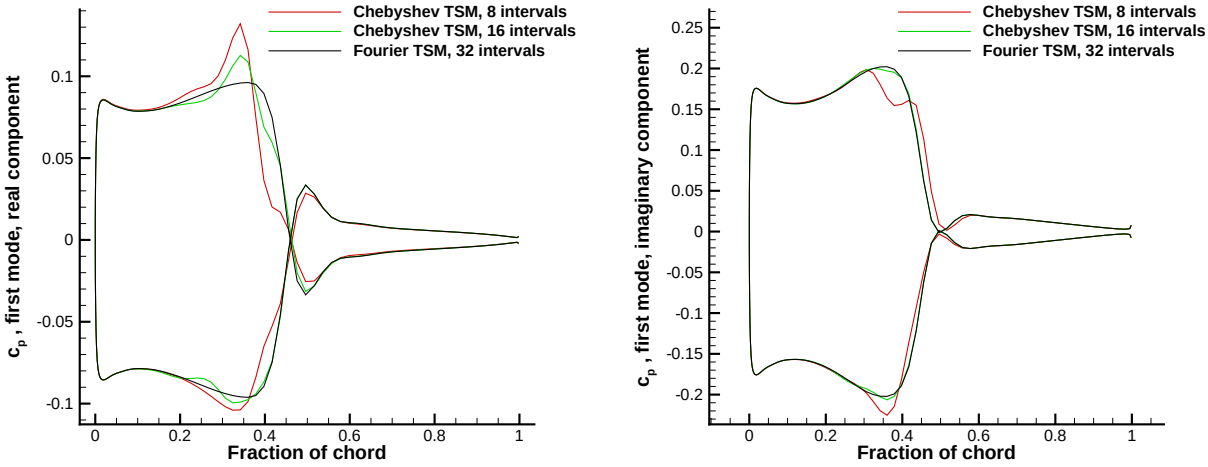


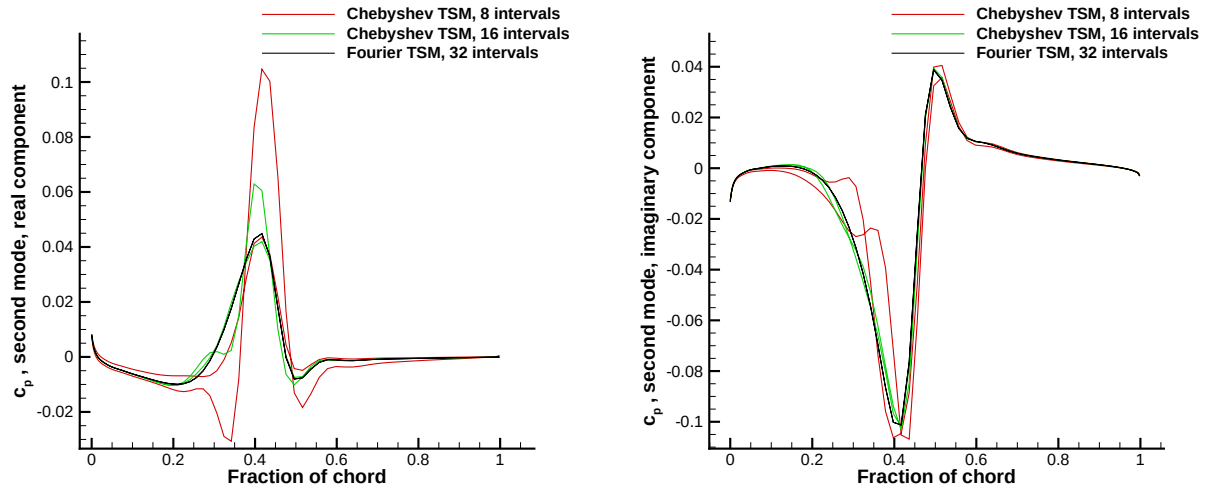
Figure 3.22: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, time averaged.

coefficient and moment coefficient. As time resolution increases, the solution becomes less non-symmetric. Similar situation can be found for surface force distribution. The computed time averaged component and the first three modes of surface pressure coefficient distribution are shown in Figures 3.22, 3.23, 3.24 and 3.25 for both time spectral methods. Dramatic non-symmetric solution can be observed even when 16 intervals are used in the Chebyshev time spectral method. The reason for this is time levels are not equally spaced in time domain. Though the number of intervals is even, the requirement that every time level should be able to find another one with phase difference of 180 degrees away from it is not satisfied. So its understandable that symmetric solution is not achieved using even number of intervals in the Chebyshev time spectral method.



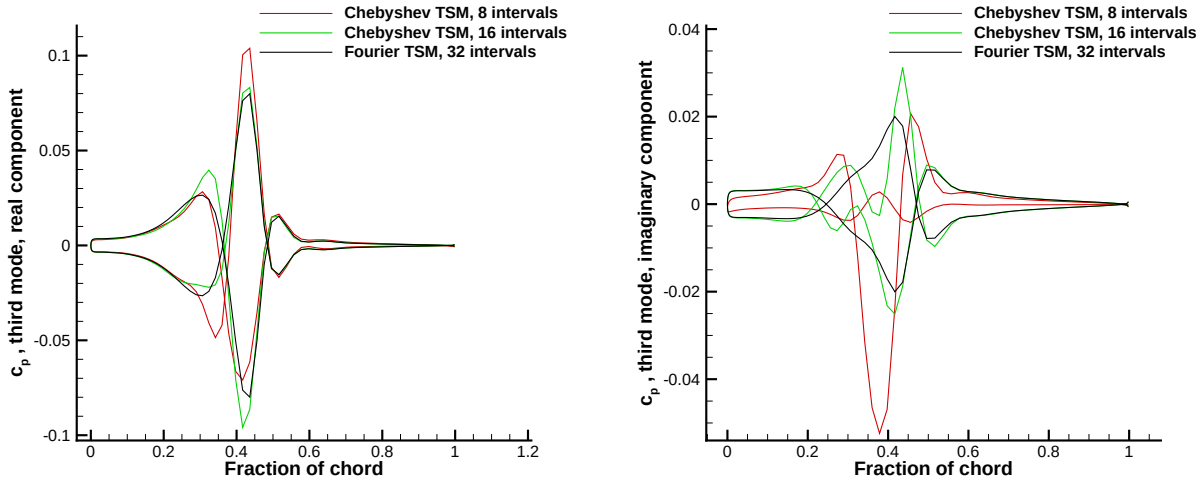
(a) real component (b) imaginary component

Figure 3.23: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, first mode.



(a) real component (b) imaginary component

Figure 3.24: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, second mode.



(a) real component

(b) imaginary component

Figure 3.25: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, third mode.

3.4.3 Non-periodic Inviscid Flow over Pitching NACA0012 Airfoil

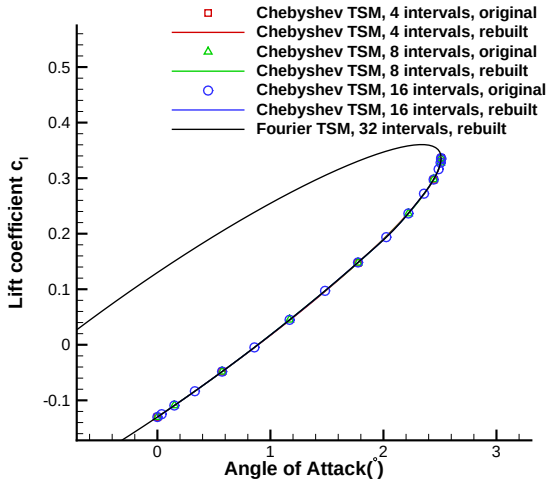
As can be seen in previous inviscid test case, the Chebyshev Time Spectral Method can be applied to solve periodic flow problems though non-symmetric solution could be produced for a symmetric problem. Most importantly, the Chebyshev time spectral method could be used for non-periodic flow cases since it is not based on temporal periodic condition. For the sake of convenience, a non-periodic inviscid test case is constructed from the airfoil pitching case discussed before. In this non-periodic test case, a NACA0012 airfoil is still forced to pitch around its quarter chord at $M_\infty = 0.755$. The mean angle of attack, pitching range and reduced frequency are still given by $\alpha_0 = 0.0°$, $\alpha_m = 2.51°$ and $\kappa = 0.0814$, respectively.

To construct the non-periodic inviscid test case, the pitching motion of the airfoil is considered only in the time interval $(0, 0.25T)$, where T is the period of the pitching motion. In non-periodic problems, since no temporal periodic condition is performed, the initial flow field must be specified to guarantee the unique unsteady solution in the time interval of interest. For all the constructed non-periodic test cases in this dissertation, the initial flow

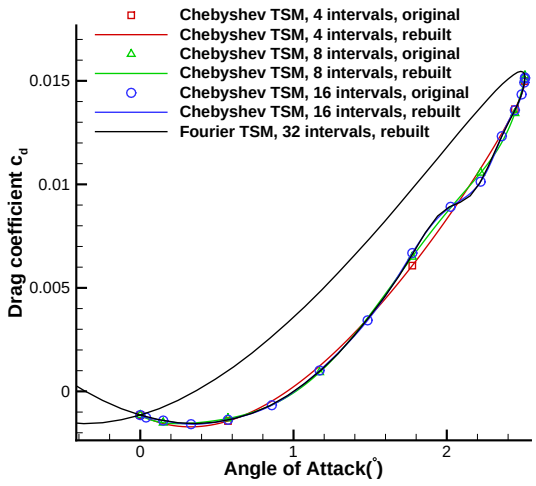
field is chosen to be the exact flow solution at the initial time level. In this way, the error of the entire non-periodic solution due to the error in initial flow field is excluded. The exact flow field can be obtained from the periodic solution using Fourier time spectral method with sufficiently high time resolution. Then the periodic solution on the zero time level can be used directly as the initial flow field for the non-periodic problem. To achieve the exact initial flow field, a period is divided into 32 intervals in the periodic calculation. As can be seen from the periodic solutions using Fourier time spectral method, 32 or even more intervals in a period are needed to well resolve flow details including shock wave movement.

The same 225×33 C-type grid in section 3.2.1 is used in the present non-periodic test case. The convergence criterion is set to be 1×10^{-14}. Three different time resolution are used in Chebyshev Time Spectral Method, for which a period is split into 4, 8 and 16 intervals respectively. Results using Fourier time spectral method with 32 intervals are provided as exact solution.

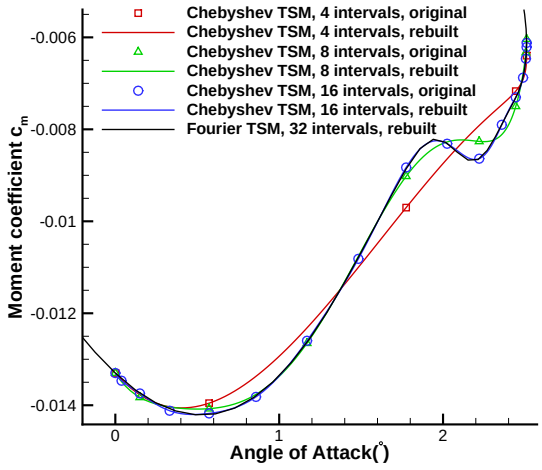
The variation of integrated force coefficients with respect to angle of attack obtained from the present computations are shown in Figure 3.26. The symbols denote the results on the original real time levels, while the lines represent the rebuilt results by the resolved Fourier modes or Chebyshev polynomials. The rebuilt results are plotted on 129 equally spaced real time levels. As time resolution increases, the results using Chebyshev time spectral method converge to the exact solution in the quarter period of interest. It can be seen that, if the lift coefficient variation is the main concern, then four intervals are sufficient to resolve it. For drag coefficient and moment coefficient, 16 intervals are required to resolve details of the variation. In the present non-periodic test case, shock waves occur and move on both upper and lower surfaces of the airfoil. The temporal variation of the lift coefficient is not sensitive to the shock-wave movement. So the variation is relatively simple and 5 Chebyshev polynomials are enough to resolve it. Whereas the shock-wave movement has more influence on the variation of the drag coefficient and moment coefficient. Then as



(a) lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 3.26: Pitching NACA0012 airfoil, non-periodic inviscid flow, force coefficient versus angle of attack.

many as 17 Chebyshev polynomials are needed to the complicated variation of these two force coefficients.

3.4.4 Non-periodic Laminar Flow over Pitching NACA0012 Airfoil

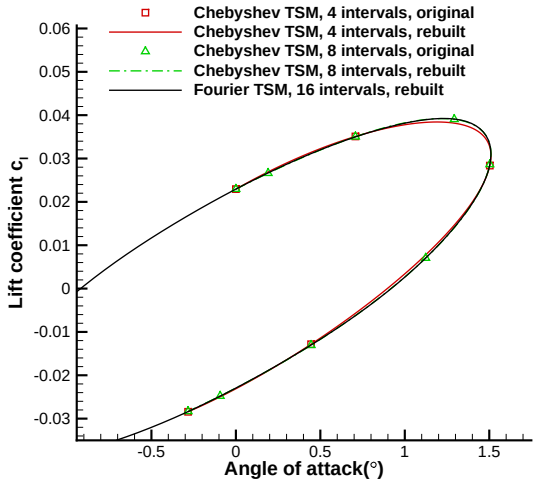
Though the flow in the previous non-periodic test case is inviscid, the shock-wave movement imposes great challenge for the Chebyshev time spectral method to save computational efforts. It's interesting to test the method in non-periodic laminar flow test cases. For the sake of convenience, one of such test cases is constructed from the laminar periodic flow over the pitching NACA0012 airfoil. In this constructed non-periodic test case, the NACA0012 airfoil pitches at $M_\infty = 0.5$ and $Re = 5000$. So the flow is shock-free and laminar. For the pitching motion, $\alpha_0 = 0.0°$, $\alpha_m = 1.51°$ and $\kappa = 0.05$. Since the mean angle of attack is zero and the pitching motion is moderate, flow in most surface area is attached. Only small vortices could occur near the trailing edge.

To construct the present non-periodic test case, the pitching motion is considered only in the time interval $(0, 0.53T)$, where T is the pitching period. The exact initial flow field is obtained in the similar manner used for the inviscid non-periodic test case. The difference is only 16 intervals are used in one period for the periodic laminar calculation using the Fourier time spectral method. This is because the present laminar flow is shock-free and 16 intervals in a period should be sufficient to achieve accurate periodic solution.

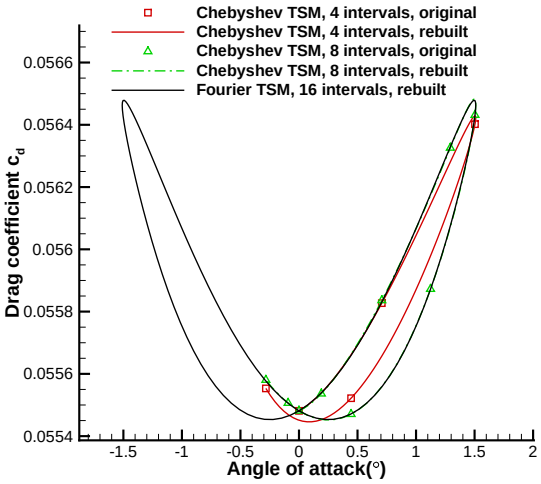
A 257×65 C-type mesh is used for both the LU-SGS and the explicit Runge-Kutta solver. The distance in the surface normal direction from the first grid point to the surface is 1×10^{-4}. The convergence criteria is set to 1×10^{-5}. In the computations using the Chebyshev time spectral method, the time interval of interest is divided into either 4 or 8 smaller intervals for two different time resolution. The periodic solution obtained by Fourier time spectral

method with 16 intervals in a period is also plotted to provide exact solution.

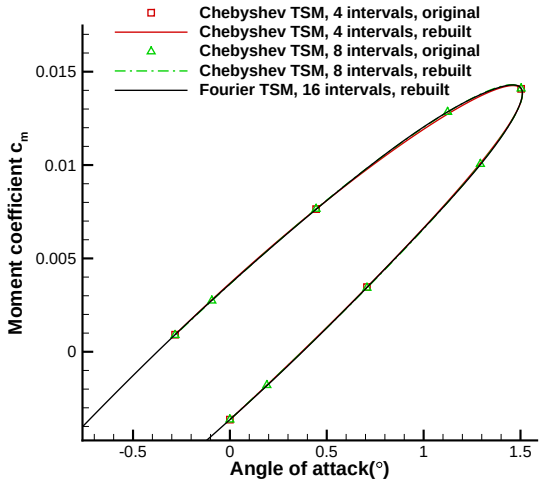
The variation of integrated force coefficients with respect to angle of attack obtained from the present computations are shown in Figure 3.27. The symbols denote the results on the original real time levels, while the lines represent the rebuilt results by the resolved Fourier modes or Chebyshev polynomials. The rebuilt results are plotted on 129 equally spaced real time levels. As time resolution increases, the results using Chebyshev time spectral method converge to the exact solution in the quarter period of interest. It can be seen, results using Chebyshev time spectral method with 4 intervals deviate from the exact solution. The deviation for lift and moment coefficient is small, while that for the drag coefficient is dramatic. When time resolution is increased to having 8 intervals in the time interval of interest, the non-periodic variation of all the force coefficients can be well predicted by the Chebyshev time spectral method. It can be concluded from the present laminar test case, phenomena such as the boundary layer adjacent to airfoil surface and small vortices near trailing edge does not impose great challenge for the Chebyshev time spectral method to save computational effort. However, if separation occurs in the flow above airfoil surface or shock wave appears and interacts with the boundary layer. Much finer time resolution may be required to well resolve the temporal variation of the aerodynamic forces.



(a) lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 3.27: Pitching NACA0012 airfoil, non-periodic laminar flow, force coefficient versus angle of attack.

Chapter 4

The Space-Time LU-SGS Scheme for the Time Spectral Method

As discussed in Section 1.1, the dual-time stepping scheme using the Backward Difference Formula (BDF) method could be improved to achieve higher computational efficiency in two aspects. The first aspect is the real time accuracy. The BDF method only couples three real time levels to discretize the real time derivatives hence it is second-order in real time. Though it is more accurate than the two-step first-order backward Euler discretization, the accuracy of the BDF method is still of algebraic order. In contrast, the time spectral method couples all real time levels in the time interval of interest. This method could achieve higher than any algebraic order time accuracy in the most ideal cases, like the subsonic shock-free flows over a pitching airfoil. So as shown and concluded in last chapter, the efficiency of unsteady flow computations can be improved by replacing the BDF method with the time spectral method.

However, in the early development of time spectral method, the explicit Runge-Kutta pseudo-time marching is still employed to get the steady-state time spectral solution. In explicit

schemes, the pseudo-time step is limited due to CFL condition. The situation becomes worse when the explicit pseudo-time marching schemes are applied to time spectral method. It is well-known that the stability requirement could be greatly relaxed or even removed in implicit schemes. So developing implicit pseudo-time marching schemes suitable for time spectral method has great potential to further improve efficiency of unsteady computations.

Several attempts were made in this direction recently [24] [25] [26] [27]. Fast convergence and improved computational efficiency were reported for these schemes. However these implicit schemes are all based on separating treatments in space and time; hence they are very complex and not straightforward. The present work is seeking a new simpler algorithm using unified treatments in both space and time.

In those recently proposed implicit schemes, the original Lower-Upper Symmetric Gauss-Seidel (LU-SGS) scheme for steady-state computations are frequently employed due to its low numerical complexity and modest memory requirements, which are both comparable to an explicit Runge-Kutta scheme. For this reason, the original LU-SGS scheme is also adopted in the new implicit scheme proposed by this dissertation. In this chapter, the original LU-SGS scheme is derived and validated with test cases first. Then the Block-Jacobi implicit algorithm [24] for time spectral method, which is one of the most representative methods making use of the original LU-SGS scheme, is introduced. Finally, a new space-time LU-SGS scheme is proposed and validated in test cases. Computational performance of this scheme is compared with that of Block-Jacobi implicit algorithm.

4.1 The LU-SGS Scheme for Steady Problems and its Formulation

For the sake of simplicity, the original Lower Upper-Symmetric Gauss Seidel (LU-SGS) scheme for steady-state calculation is derived on the two-dimensional Euler equations, which can be written in differential form as

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{4.1}$$

where

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \; f = \begin{pmatrix} \rho\bar{u} \\ \rho u\bar{u} + p \\ \rho v\bar{u} \\ \rho E\bar{u} + pu \end{pmatrix}, \; g = \begin{pmatrix} \rho\bar{v} \\ \rho u\bar{v} \\ \rho v\bar{v} + p \\ \rho E\bar{v} + pv \end{pmatrix} \tag{4.2}$$

and $(\bar{u}, \bar{v}) = (u - u_b, v - v_b)$ stands for the local convective velocity relative to the control surface moving at the velocity of (u_b, v_b).

Integrating the above equations over a quadrilateral grid cell in two-dimensional space, the general implicit scheme can be formulated as

$$\Omega_{i,j} \frac{w_{i,j}^{q+1} - w_{i,j}^{q}}{\Delta t} + \beta \sum_{m=1}^{4} [\mathbf{F}(w_m^{q+1}) \cdot \vec{S}_m] + (1-\beta) \sum_{m=1}^{4} [\mathbf{F}(w_m^{q}) \cdot \vec{S}_m] = 0 \tag{4.3}$$

where $\Omega_{i,j}$ is the volume of cell (i, j), m the index of a cell face, $\mathbf{F}(w_m) = f(w_m)\vec{i} + g(w_m)\vec{j}$ and $\vec{S}_m = (n_{x,m}\vec{i} + n_{y,m}\vec{j})\Delta S_m$ the surface vector with surface length ΔS_m and surface direction vector $(n_{x,m}, n_{y,m})$ pointing outwards for face m. The sketch of such a grid cell is shown in Figure 4.1.

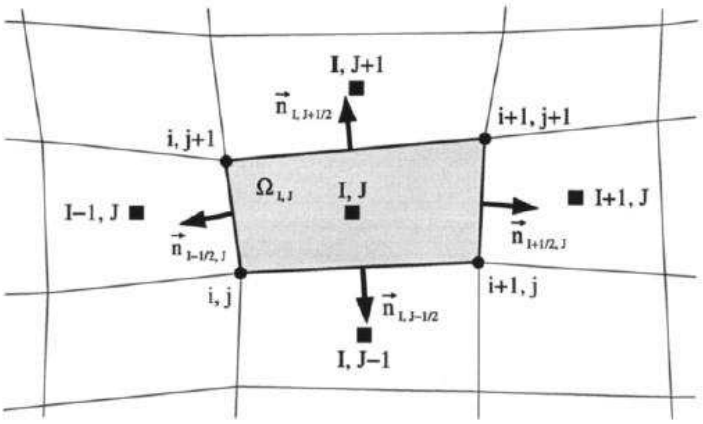


Figure 4.1: Sketch of a grid cell for current LU-SGS scheme (in two dimensions) [39].

If $\beta = \frac{1}{2}$ the scheme remains second-order accurate in time, while for other values of β the time accuracy drops to first order. Following Yoon and Jameson's choice [8], $\beta = 1$ is adopted here. The convective flux of time step $q+1$ can be linearised at time step q as follows

$$\begin{aligned}\mathbf{F}(w_m^{q+1}) &= f(w_m^{q+1})\vec{i} + g(w_m^{q+1})\vec{j} \\ &= [f(w_m^q)\vec{i} + (\frac{\partial f}{\partial w})_m \delta w_m]\vec{i} + [g(w_m^q)\vec{i} + (\frac{\partial g}{\partial w})_m \delta w_m]\vec{j} \\ &= \mathbf{F}(w_m^q) + A_m \delta w_m \vec{i} + B_m \delta w_m \vec{j}\end{aligned} \tag{4.4}$$

where $A_m = (\frac{\partial f}{\partial w})_m$ and $B_m = (\frac{\partial g}{\partial w})_m$ are convective flux Jacobian matrices in x and y directions respectively. Then equation (4.3) can be cast into the following form

$$\Omega_{i,j}\delta w_{i,j} + \Delta t \sum_{m=1}^{4} \mathbf{A}_m \Delta S_m \delta w_m + \Delta t R_{i,j} = 0 \tag{4.5}$$

where $R_{i,j} = \sum_{m=1}^{4} [\mathbf{F}(w_m^q) \cdot \vec{S}_m]$ is the explicit residual estimated at time step q and $\mathbf{A}_m = A_m n_{x,m} + B_m n_{y,m}$ the convective flux Jacobian matrix on cell face m.

To proceed further, firstly, all outward surface vectors should be reoriented so that they all

point in the i increasing or j increasing directions. Then equation (4.5) can be rewritten as

$$\begin{aligned}
&\Omega_{i,j}\delta w_{i,j} - \Delta t[A_{i-\frac{1}{2},j}(-n_{x,i-\frac{1}{2},j}) + B_{i-\frac{1}{2},j}(-n_{y,i-\frac{1}{2},j})]\delta w_{i-\frac{1}{2},j}\Delta S_{i-\frac{1}{2},j} \\
&\quad + \Delta t[A_{i+\frac{1}{2},j}(n_{x,i+\frac{1}{2},j}) + B_{i+\frac{1}{2},j}(n_{y,i+\frac{1}{2},j})]\delta w_{i+\frac{1}{2},j}\Delta S_{i+\frac{1}{2},j} \\
&\quad - \Delta t[A_{i,j-\frac{1}{2}}(-n_{x,i,j-\frac{1}{2}}) + B_{i,j-\frac{1}{2}}(-n_{y,i,j-\frac{1}{2}})]\delta w_{i,j-\frac{1}{2}}\Delta S_{j,i-\frac{1}{2}} \\
&\quad + \Delta t[A_{i,j+\frac{1}{2}}(n_{x,i,j+\frac{1}{2}}) + B_{i,j+\frac{1}{2}}(n_{y,i,j+\frac{1}{2}})]\delta w_{i,j+\frac{1}{2}}\Delta S_{i,j+\frac{1}{2}} \\
&\quad + \Delta t R_{i,j} = 0
\end{aligned} \tag{4.6}$$

The above equation can be simplified using $\tilde{\mathbf{A}}_{i-\frac{1}{2},j} = A_{i-\frac{1}{2},j}(-n_{x,i-\frac{1}{2},j}) + B_{i-\frac{1}{2},j}(-n_{y,i-\frac{1}{2},j})$ and $\hat{\mathbf{A}}_{i+\frac{1}{2},j} = A_{i+\frac{1}{2},j}(n_{x,i+\frac{1}{2},j}) + B_{i+\frac{1}{2},j}(n_{y,i+\frac{1}{2},j})$ in i direction for example. Next, following Yoon and Jameson's approach[8], the matrix $\tilde{\mathbf{A}}_{i-\frac{1}{2},j}$ can be split as

$$\tilde{\mathbf{A}}_{i-\frac{1}{2},j} = \tilde{\mathbf{A}}^{+}_{i-\frac{1}{2},j} + \tilde{\mathbf{A}}^{-}_{i-\frac{1}{2},j} \tag{4.7}$$

where the " $+$ " matrix and the " $-$ " matrix are constructed in the following way such that the eigenvalues of " $+$ " matrix are non-negative and those of " $-$ " matrix are non-positive.

$$\begin{aligned}
\tilde{\mathbf{A}}^{+}_{i-\frac{1}{2},j} &= \frac{1}{2}\{\tilde{\mathbf{A}}_{i-\frac{1}{2},j} + [\left|\bar{u}_{i-\frac{1}{2},j}(-n_{x,i-\frac{1}{2},j}) + \bar{v}_{i-\frac{1}{2},j}(-n_{y,i-\frac{1}{2},j})\right| + c_{i-\frac{1}{2},j}]\mathbf{I}\} \\
\tilde{\mathbf{A}}^{-}_{i-\frac{1}{2},j} &= \frac{1}{2}\{\tilde{\mathbf{A}}_{i-\frac{1}{2},j} - [\left|\bar{u}_{i-\frac{1}{2},j}(-n_{x,i-\frac{1}{2},j}) + \bar{v}_{i-\frac{1}{2},j}(-n_{y,i-\frac{1}{2},j})\right| + c_{i-\frac{1}{2},j}]\mathbf{I}\}
\end{aligned} \tag{4.8}$$

Here c is the local speed of sound on cell face $(i-\frac{1}{2},j)$ and I is the identity matrix. $\left|\bar{u}_{i-\frac{1}{2},j}(-n_{x,i-\frac{1}{2},j}) + \bar{v}_{i-\frac{1}{2},j}(-n_{y,i-\frac{1}{2},j})\right| + c$ is the spectral radius of matrix $\tilde{\mathbf{A}}_{i-\frac{1}{2},j}$. Employing up-winding concept, the " $+$ " matrix at grid side $(i-\frac{1}{2},j)$ can be estimated on its left neighbouring cell center $(i-1,j)$. This is because all eigenvalues of the " $+$ " matrix being non-negative indicate that all information propagates in the i increasing direction. Similarly, the " $-$ " matrix at grid side $(i-\frac{1}{2},j)$ can be calculated on its right neighbouring cell center

(i, j). Having the above ideas in mind, equation (4.6) can be modified as

$$\begin{aligned}
\Omega_{i,j}\delta w_{i,j} &- \Delta t[\tilde{\mathbf{A}}^{+}_{i-1,j}\delta w_{i-1,j} + \tilde{\mathbf{A}}^{-}_{i,j}\delta w_{i,j}]\Delta S_{i-\frac{1}{2},j} \\
&+ \Delta t[\hat{\mathbf{A}}^{+}_{i,j}\delta w_{i,j} + \hat{\mathbf{A}}^{-}_{i+1,j}\delta w_{i+1,j}]\Delta S_{i+\frac{1}{2},j} \\
&- \Delta t[\tilde{\mathbf{A}}^{+}_{i,j-1}\delta w_{i,j-1} + \tilde{\mathbf{A}}^{-}_{i,j}\delta w_{i,j}]\Delta S_{i,j-\frac{1}{2}} \\
&+ \Delta t[\hat{\mathbf{A}}^{+}_{i,j}\delta w_{i,j} + \hat{\mathbf{A}}^{-}_{i,j+1}\delta w_{i,j+1}]\Delta S_{i,j+\frac{1}{2}} \\
&+ \Delta t R_{i,j} = 0
\end{aligned} \tag{4.9}$$

In the above equation, it can be shown that in the i direction

$$\hat{\mathbf{A}}^{+}_{i,j}\Delta S_{i+\frac{1}{2},j}\delta w_{i,j} - \tilde{\mathbf{A}}^{-}_{i,j}\Delta S_{i-\frac{1}{2},j}\delta w_{i,j} = [\left|\bar{u}_{i,j}n^{I}_{x,i,j} + \bar{v}_{i,j}n^{J}_{y,i,j}\right| + c_{i,j}]\Delta S^{I}_{i,j}\delta w_{i,j}\mathbf{I} \tag{4.10}$$

if the following relations are defined in the way specified by Blazek [39]

$$\begin{aligned}
n^{I}_{x,i,j} &= \frac{n_{x,i+\frac{1}{2},j} - n_{x,i-\frac{1}{2},j}}{2} \\
n^{I}_{y,i,j} &= \frac{n_{y,i+\frac{1}{2},j} - n_{y,i-\frac{1}{2},j}}{2} \\
\Delta S^{I}_{i,j} &= \frac{\Delta S_{i+\frac{1}{2},j} + \Delta S_{i-\frac{1}{2},j}}{2}
\end{aligned} \tag{4.11}$$

Similarly, in the j direction

$$\hat{\mathbf{A}}^{+}_{i,j}\Delta S_{i,j+\frac{1}{2}}\delta w_{i,j} - \tilde{\mathbf{A}}^{-}_{i,j}\Delta S_{i,j-\frac{1}{2}}\delta w_{i,j} = [\left|\bar{u}_{i,j}n^{J}_{x,i,j} + \bar{v}_{i,j}n^{J}_{y,i,j}\right| + c_{i,j}]\Delta S^{J}_{i,j}\delta w_{i,j}\mathbf{I} \tag{4.12}$$

with

$$\begin{aligned}
n^{J}_{x,i,j} &= \frac{n_{x,i,j+\frac{1}{2}} - n_{x,i,j-\frac{1}{2}}}{2} \\
n^{J}_{y,i,j} &= \frac{n_{y,i,j+\frac{1}{2}} - n_{y,i,j-\frac{1}{2}}}{2} \\
\Delta S^{J}_{i,j} &= \frac{\Delta S_{i,j+\frac{1}{2}} + \Delta S_{i,j-\frac{1}{2}}}{2}
\end{aligned} \tag{4.13}$$

Finally (4.9) can be rearranged as

$$
\begin{aligned}
&[\Omega_{i,j} + \Delta t(\left|\bar{u}_{i,j}n^I_{x,i,j} + \bar{v}_{i,j}n^I_{y,i,j}\right| + c_{i,j})\Delta S^I_{i,j} + \Delta t(\left|\bar{u}_{i,j}n^J_{x,i,j} + \bar{v}_{i,j}n^J_{y,i,j}\right| + c_{i,j})\Delta S^J_{i,j}]\delta w_{i,j} \\
&+ \Delta t[-\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w_{i-1,j} - \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w_{i,j-1} \\
&+ \hat{\mathbf{A}}^-_{i+1,j}\Delta S_{i+\frac{1}{2},j}\delta w_{i+1,j} + \hat{\mathbf{A}}^-_{i,j+1}\Delta S_{i,j+\frac{1}{2}}\delta w_{i,j+1}] + \Delta t R_{i,j} = 0
\end{aligned}
\tag{4.14}
$$

The discretized system expressed by Equation (4.14) is actually a set of linear equations for solution update δw over all points in the computational domain. The coefficient matrix of this set of linear equations is usually called the implicit operator or system matrix [39]. Clearly, from the above derivation, only convective fluxes are considered in the implicit operator. To make the implicit scheme have the best convergence behaviour, artificial dissipation fluxes should also be treated in the implicit operator. However, if the artificial dissipation term is small compared to convective term (which is usually the case), ignoring it in the implicit operator would not influence the convergence of the scheme too much. For this reason, the artificial dissipation term is not considered in the implicit operator throughout this dissertation to make the scheme simple and easy to implement.

In the implicit operator, it can be observed that the coefficient in front of $\delta w_{i,j}$ shown in (4.14) will appear in diagonal elements. Coefficients in front of $\delta w_{i-1,j}$ and $\delta w_{i,j-1}$ will be in the lower triangular part, while those in front of $\delta w_{i+1,j}$ and $\delta w_{i,j+1}$ will show up in the upper triangular part. Regardless the number of dimensions of a flow problem, the implicit operator matrix is always composed of such diagonal, lower and upper matrices. This is one feature of the present LU-SGS scheme. The "LU" in the name of the method is short for "Lower and Upper", which refers to the above fact. The "SGS" in the method's name is short for "Symmetric Gauss-Seidel" iterations, which involves two "symmetric" sweeping procedures and is described as follows

Forward sweep

$$
\begin{aligned}
&[\Omega_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^I_{x,i,j} + \bar{v}_{i,j} n^I_{y,i,j}\right| + c_{i,j})\Delta S^I_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^J_{x,i,j} + \bar{v}_{i,j} n^J_{y,i,j}\right| + c_{i,j})\Delta S^J_{i,j}]\delta w^*_{i,j} \\
&+ \Delta t[-\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w^*_{i-1,j} - \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w^*_{i,j-1} \\
&+ \hat{\mathbf{A}}^-_{i+1,j}\Delta S_{i+\frac{1}{2},j}\delta w^{(0)}_{i+1,j} + \hat{\mathbf{A}}^-_{i,j+1}\Delta S_{i,j+\frac{1}{2}}\delta w^{(0)}_{i,j+1}] + \Delta t R_{i,j} = 0
\end{aligned}
\tag{4.15}
$$

Backward sweep

$$
\begin{aligned}
&[\Omega_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^I_{x,i,j} + \bar{v}_{i,j} n^I_{y,i,j}\right| + c_{i,j})\Delta S^I_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^J_{x,i,j} + \bar{v}_{i,j} n^J_{y,i,j}\right| + c_{i,j})\Delta S^J_{i,j}]\delta w^{(1)}_{i,j} \\
&+ \Delta t[-\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w^*_{i-1,j} - \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w^*_{i,j-1} \\
&+ \hat{\mathbf{A}}^-_{i+1,j}\Delta S_{i+\frac{1}{2},j}\delta w^{(1)}_{i+1,j} + \hat{\mathbf{A}}^-_{i,j+1}\Delta S_{i,j+\frac{1}{2}}\delta w^{(1)}_{i,j+1}] + \Delta t R_{i,j} = 0
\end{aligned}
\tag{4.16}
$$

where $\delta w^{(0)}$ is initial value of a solution update before forward sweep, δw^* is the intermediate value after forward sweep, and $\delta w^{(1)}$ is the final solution update after backward sweep. When the two sweeps are completed, the solution on each point in the computational domain is updated as

$$
w^{q+1} = w^q + \delta w^{(1)} \tag{4.17}
$$

Usually, for the sake of simplicity, the initial value of solution update $\delta w^{(0)}$ is set to be zero over the entire computational domain including all boundaries. In this way, the evolution of solution update on point (i, j) can be formulated as follows

Forward sweep

$$
\begin{aligned}
\delta w^*_{i,j} = &\frac{\Delta t}{\Omega_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^I_{x,i,j} + \bar{v}_{i,j} n^I_{y,i,j}\right| + c_{i,j})\Delta S^I_{i,j} + \Delta t(\left|\bar{u}_{i,j} n^J_{x,i,j} + \bar{v}_{i,j} n^J_{y,i,j}\right| + c_{i,j})\Delta S^J_{i,j}} \\
&[\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w^*_{i-1,j} + \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w^*_{i,j-1} - R_{i,j}]
\end{aligned}
\tag{4.18}
$$

Backward sweep

$$\begin{aligned}\delta w_{i,j}^{(1)} =& \frac{\Delta t}{\Omega_{i,j} + \Delta t(\left|\bar{u}_{i,j} n_{x,i,j}^{I} + \bar{v}_{i,j} n_{y,i,j}^{I}\right| + c_{i,j})\Delta S_{i,j}^{I} + \Delta t(\left|\bar{u}_{i,j} n_{x,i,j}^{J} + \bar{v}_{i,j} n_{y,i,j}^{J}\right| + c_{i,j})\Delta S_{i,j}^{J}} \\ & [\tilde{\mathbf{A}}_{i-1,j}^{+} \Delta S_{i-\frac{1}{2},j} \delta w_{i-1,j}^{*} + \tilde{\mathbf{A}}_{i,j-1}^{+} \Delta S_{i,j-\frac{1}{2}} \delta w_{i,j-1}^{*} \\ & - \hat{\mathbf{A}}_{i+1,j}^{-} \Delta S_{i+\frac{1}{2},j} \delta w_{i+1,j}^{(1)} - \hat{\mathbf{A}}_{i,j+1}^{-} \Delta S_{i,j+\frac{1}{2}} \delta w_{i,j+1}^{(1)} - R_{i,j}]\end{aligned} \tag{4.19}$$

To guarantee convergence of the Gauss-Seidel iteration method, it is well known that the coefficient matrix of the corresponding linear system should be diagonally dominant. In the present LU-SGS scheme, this requirement is satisfied by introducing the way of constructing the $\mathbf{A}^{+}$ and $\mathbf{A}^{-}$ matrices, which is illustrated in (4.7) and (4.8) already. It can be seen that the spectral radius of local convective flux Jacobian matrix is always in the diagonal elements of the implicit operator. This feature brings another advantage of the scheme. Since the spectral radius is a number, which is easier to be converted than a matrix, the local solution update can be calculated very conveniently.

For a lot of steady inviscid flow problems described by the Euler equations, an infinitely large time step Δt is usually allowed for fast stable convergence. Then the LU-SGS scheme expressed in (4.18) and (4.19) reduces to an approximate Newton iteration method [8] as

Forward sweep

$$\begin{aligned}\delta w_{i,j}^{*} =& \frac{1}{(\left|\bar{u}_{i,j} n_{x,i,j}^{I} + \bar{v}_{i,j} n_{y,i,j}^{I}\right| + c_{i,j})\Delta S_{i,j}^{I} + (\left|\bar{u}_{i,j} n_{x,i,j}^{J} + \bar{v}_{i,j} n_{y,i,j}^{J}\right| + c_{i,j})\Delta S_{i,j}^{J}} \\ & [\tilde{\mathbf{A}}_{i-1,j}^{+} \Delta S_{i-\frac{1}{2},j} \delta w_{i-1,j}^{*} + \tilde{\mathbf{A}}_{i,j-1}^{+} \Delta S_{i,j-\frac{1}{2}} \delta w_{i,j-1}^{*} - R_{i,j}]\end{aligned} \tag{4.20}$$

Backward sweep

$$\begin{aligned}\delta w_{i,j}^{(1)} =& \frac{1}{(\left|\bar{u}_{i,j}n_{x,i,j}^{I}+\bar{v}_{i,j}n_{y,i,j}^{I}\right|+c_{i,j})\Delta S_{i,j}^{I}+(\left|\bar{u}_{i,j}n_{x,i,j}^{J}+\bar{v}_{i,j}n_{y,i,j}^{J}\right|+c_{i,j})\Delta S_{i,j}^{J}} \\ & [\tilde{\mathbf{A}}_{i-1,j}^{+}\Delta S_{i-\frac{1}{2},j}\delta w_{i-1,j}^{*}+\tilde{\mathbf{A}}_{i,j-1}^{+}\Delta S_{i,j-\frac{1}{2}}\delta w_{i,j-1}^{*} \\ & -\hat{\mathbf{A}}_{i+1,j}^{-}\Delta S_{i+\frac{1}{2},j}\delta w_{i+1,j}^{(1)}-\hat{\mathbf{A}}_{i,j+1}^{-}\Delta S_{i,j+\frac{1}{2}}\delta w_{i,j+1}^{(1)}-R_{i,j}]\end{aligned} \tag{4.21}$$

In order to avoid explicit calculation and storage of the convective flux Jacobian matrices in the LU-SGS scheme, the products $\mathbf{A}^{\pm}\delta w\Delta S$ can be approximately evaluated by calculating the flux difference [39]. For example, during forward sweep

$$\begin{aligned}\tilde{\mathbf{A}}_{i-1,j}^{+}\Delta S_{i-\frac{1}{2},j}\delta w_{i-1,j}^{*} = \frac{1}{2}\{&[\mathbf{F}(w_{i-1,j}^{q}+\delta w_{i-1,j}^{*})-\mathbf{F}(w_{i-1,j}^{q})]\cdot(-\vec{S}_{i-\frac{1}{2},j}) \\ &+[\left|\bar{u}_{i-1,j}(-n_{x,i-\frac{1}{2},j})+\bar{v}_{i-1,j}(-n_{y,i-\frac{1}{2},j})\right|+c_{i-1,j}]\Delta S_{i-\frac{1}{2},j}\}\end{aligned} \tag{4.22}$$

note that product $\tilde{\mathbf{A}}_{i-1,j}^{+}\Delta S_{i-\frac{1}{2},j}\delta w_{i-1,j}^{*}$ is used for updating $\delta w_{i,j}^{*}$. Point $(i-1,j)$ is swept before point (i,j) during forward sweep. So $\delta w_{i-1,j}^{*}$ has already become available when $\delta w_{i,j}^{*}$ is calculated. The use of (4.22) utilizes this fact.

Another important feature of the LU-SGS scheme is the diagonal sweeping order. For the forward sweep, this special sweeping order is illustrated in Figure 4.2. • denotes the points that are currently swept(on line $i+j=$ constant), while $\times$ denotes the points that have been swept already. During backward sweep, the sweeping order is reversed, in other words, from top right to bottom left. For every diagonal line, solution update δw can be calculated independently on each point of it. It indicates the scheme can be easily vectorised [39]. This is why the diagonal sweeping order is frequently used.

When viscous effect is considered, the above LU-SGS scheme needs to be revised. First of all, the explicit residual should include viscous fluxes. Besides, the viscous fluxes should be treated in the implicit operator of LU-SGS scheme. This requirement has to be fulfilled especially for the viscous flows where viscous effect is also significant in large area, such as

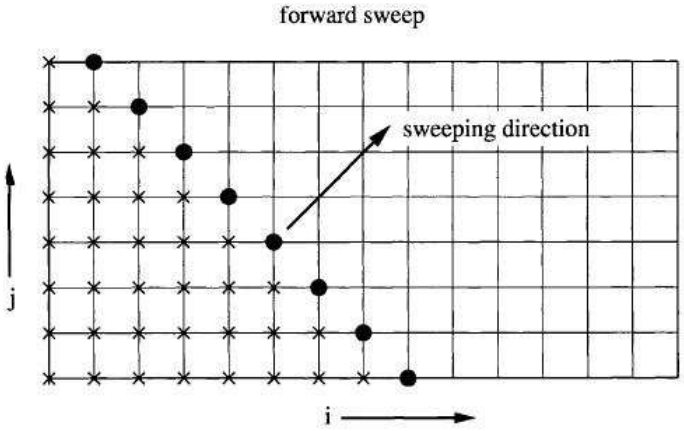


Figure 4.2: Sweeping order of the LU-SGS scheme in computational domain [39].

the low Reynolds flow over a circular cylinder. However, the complete treatment of viscous term in the implicit operator is very complicated. To balance the accuracy of the scheme and the implementation difficulty, the way suggested by Tysinger and Caughey [40] [41] is employed in this dissertation. That is to add the following term to the diagonal elements in the implicit operator

$$\frac{\gamma\mu}{\rho Pr}\frac{(\Delta S^I_{i,j})^2+(\Delta S^J_{i,j})^2}{\Omega_{i,j}} \tag{4.23}$$

where γ is the ratio of specific heats, ρ the local density, Pr the Prandtl number and Ω the volume of grid cell. $\Delta S^I_{i,j}$ and $\Delta S^J_{i,j}$ are length defined in (4.11) and (4.13) respectively. This term is actually the largest eigenvalue of the viscous Jacobian matrix. It has the same dimension as the spectral radius of the convective Jacobian matrix. The contribution of viscous fluxes to the lower and upper parts of the implicit operator is ignored. These treatments still maintain the implicit operator being diagonally dominant, hence guarantee the convergence of the Guass-Seidel iterations in LU-SGS scheme. And the new diagonal coefficient in the implicit operator remains a number, which still can be easily converted. Finally, the forward and backward sweep formula should be revised accordingly. For example,

equation (4.18) and (4.19) become

Forward sweep

$$\delta w^*_{i,j} = \frac{\Delta t}{D}[\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w^*_{i-1,j} + \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w^*_{i,j-1} - R_{i,j}] \tag{4.24}$$

Backward sweep

$$\begin{aligned}\delta w^{(1)}_{i,j} = \frac{\Delta t}{D}[&\tilde{\mathbf{A}}^+_{i-1,j}\Delta S_{i-\frac{1}{2},j}\delta w^*_{i-1,j} + \tilde{\mathbf{A}}^+_{i,j-1}\Delta S_{i,j-\frac{1}{2}}\delta w^*_{i,j-1} \\ &- \hat{\mathbf{A}}^-_{i+1,j}\Delta S_{i+\frac{1}{2},j}\delta w^{(1)}_{i+1,j} - \hat{\mathbf{A}}^-_{i,j+1}\Delta S_{i,j+\frac{1}{2}}\delta w^{(1)}_{i,j+1} - R_{i,j}]\end{aligned} \tag{4.25}$$

where

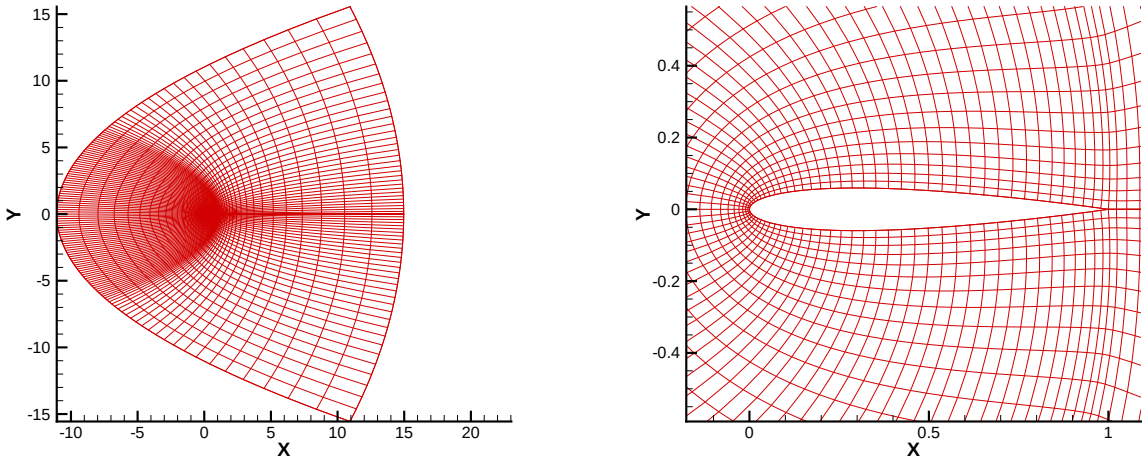
$$\begin{aligned}D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j}n^I_{x,i,j} + \bar{v}_{i,j}n^I_{y,i,j}\right| + c_{i,j})\Delta S^I_{i,j} \\ &+ \Delta t(\left|\bar{u}_{i,j}n^J_{x,i,j} + \bar{v}_{i,j}n^J_{y,i,j}\right| + c_{i,j})\Delta S^J_{i,j} \\ &+ \Delta t\frac{\gamma\mu}{\rho Pr}\frac{(\Delta S^I_{i,j})^2 + (\Delta S^J_{i,j})^2}{\Omega_{i,j}}\end{aligned} \tag{4.26}$$

4.2 Steady Flow Test Cases Using the LU-SGS Scheme

To validate the present LU-SGS solver and show improvement of computational efficiency by replacing the explicit Runge-Kutta solver with this solver, computation results for the following test cases are shown and discussed.

4.2.1 Transonic Inviscid Flow over a NACA0012 Airfoil

Inviscid transonic flow over NACA0012 airfoil at $M_\infty = 0.8$, angle of attack $\alpha = 1.25°$ is a well-studied case for testing transonic flow solvers. Computational results using explicit Runge-Kutta solver and LU-SGS solver can be found in [8]. It is also a test case in the earliest references reporting progress on LU-SGS scheme. For these reasons, this test case is



(a) 128×32 C mesh

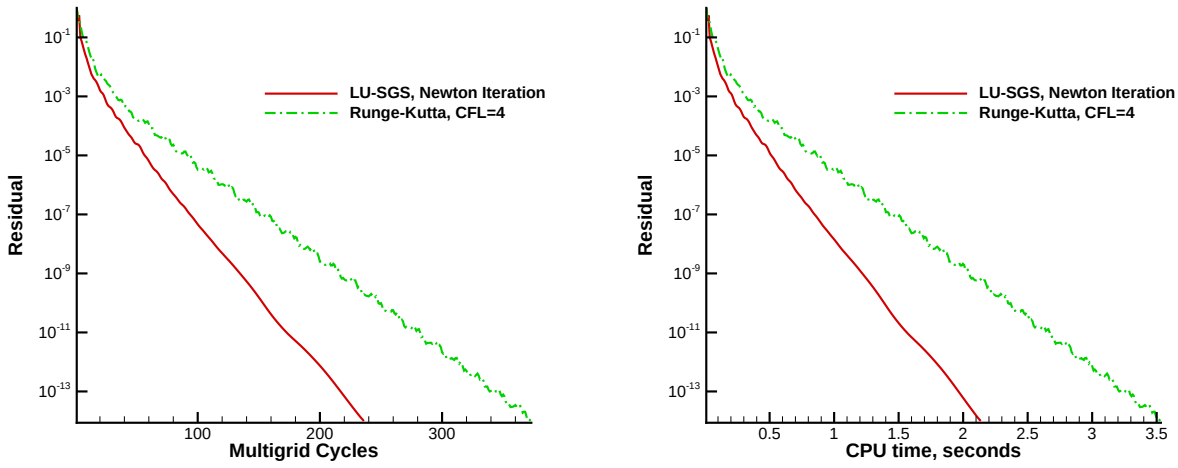
(b) Close view near airfoil surface

Figure 4.3: NACA0012 airfoil,body-fitted mesh.

chosen to be the first one for validating present LU-SGS solver.

A 129×33 C-type body-fitted mesh is used here. The mesh is shown in figure 4.3(a), the close view near airfoil surface is shown in figure 4.3(b).

Convergence history using LU-SGS scheme and explicit Runge-Kutta scheme is shown in Figure 4.4. The convergence criteria is set to be 1×10^{-14}. In Figure 4.4(a), it can be found that the explicit Runge-Kutta solver takes 376 multigrid cycles to reach convergence, which is approximately 1.61 times of that for the LU-SGS solver to reach the same residual level. When consumed CPU time is compared, the ratio rises a little to 1.67. This suggests the current LU-SGS solver consumes less CPU time in each multigrid cycle. It is mainly because the 5-stage explicit Runge-Kutta solver needs to calculate convective fluxes five times, while present LU-SGS solver just do it three times(once in the explicit residual, the other two in the implicit operator). So this test case reveals two desired features of LU-SGS scheme compared to the explicit Runge-Kutta solver. Firstly, the LU-SGS solver saves CPU time during each multigrid cycle. Secondly, less multigrid cycles are required for the LU-SGS solver to reach the same convergence criteria.



(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.4: NACA0012 airfoil, steady inviscid flow, convergence history.

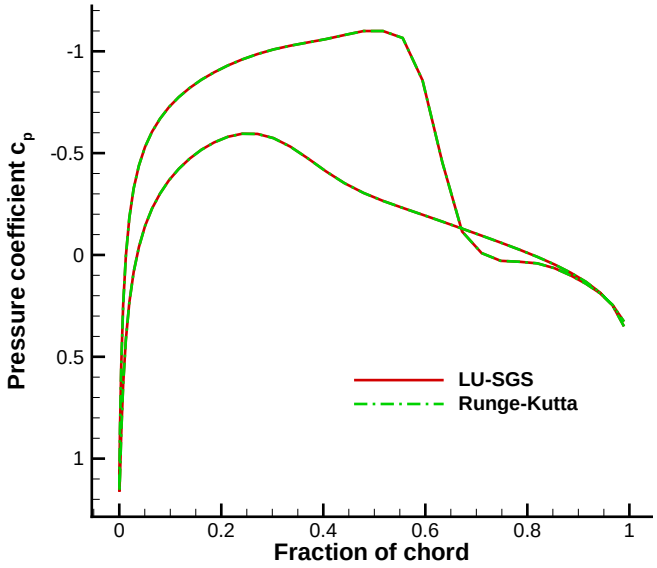


Figure 4.5: NACA0012 airfoil, steady inviscid flow, pressure coefficient.

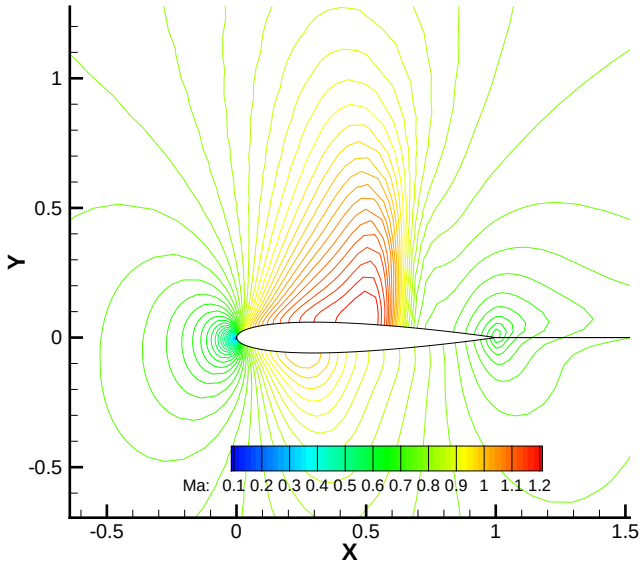


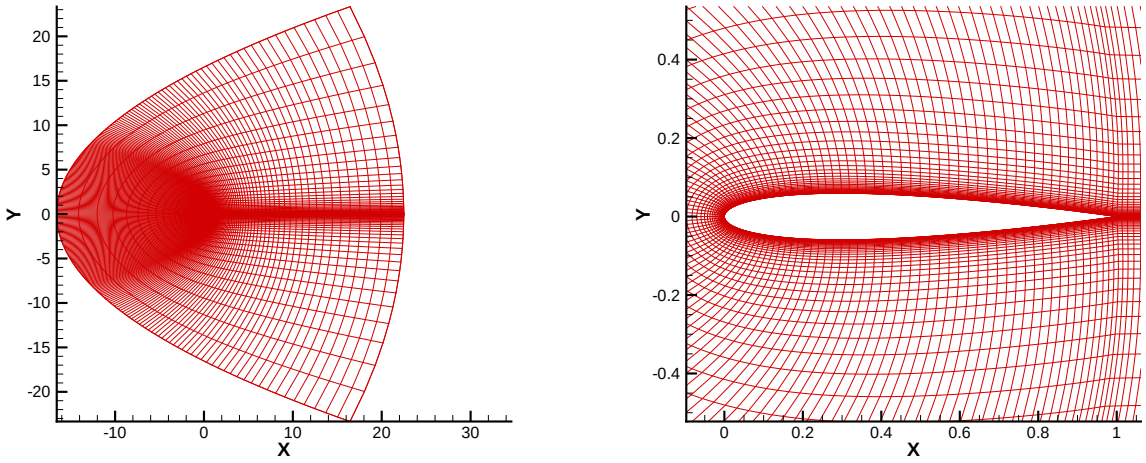
Figure 4.6: NACA0012 airfoil, steady inviscid flow, Mach number contours.

The comparison on computed surface pressure coefficient distribution is shown in Figure 4.5. The result using the LU-SGS solver matches the one using explicit Runge-Kutta solver very well. So both computations converge to the same solution.

The Mach number contours using LU-SGS solver are shown in Figure 4.6. It can be seen that a strong shock wave occurs on the upper airfoil surface, while a very week shock wave can be found on the lower surface. These features are very similar to those reported in [8].

4.2.2 Subsonic Laminar Flow over a NACA0012 Airfoil

A purely subsonic laminar flow over NACA0012 airfoil at $M_\infty = 0.5$, angle of attack $\alpha = 0°$ and Reynolds number $Re = 5000$ is computed as the second test case. A 257×65 C-type mesh is used for both the LU-SGS and the explicit Runge-Kutta solver. The distance in the surface normal direction from the first grid point to the surface is 1×10^{-4}. The mesh is



(a) 256×64 C mesh

(b) Close view near airfoil surface

Figure 4.7: NACA0012 airfoil,body-fitted mesh.

shown in Figure 4.7(a) and the close view near the airfoil surface is shown in Figure 4.7(b).

The Mach number contours using LU-SGS solver are shown in Figure 4.8. Separation occurs close to the trailing edge of the airfoil. It can be observed that viscous effects are only significant in a small region near the airfoil, as well as in the wake region since local Mach number in these these areas is pretty low.

The computed surface pressure coefficient and skin friction distribution are shown in Figure 4.9. Computations using the present LU-SGS solver and the explicit Runge-Kutta solver both converge to the same solution.

The convergence history vesus multigrid cycle is shown in Figure 4.10(a), while the result versus CPU time is shown in Figure 4.10(b). The convergence criteria is set to be 1×10^{-5}. It can be seen that, to reach convergence, explicit Runge-Kutta solver needs approximately 1.75 times the number of multigrid cycles that are required by LU-SGS solver. When CPU time is considered, this ratio increases to 2.59. Compared to the previous inviscid test case, LU-SGS solver saves more CPU time during each multigrid cycle(relative to explicit Runge-

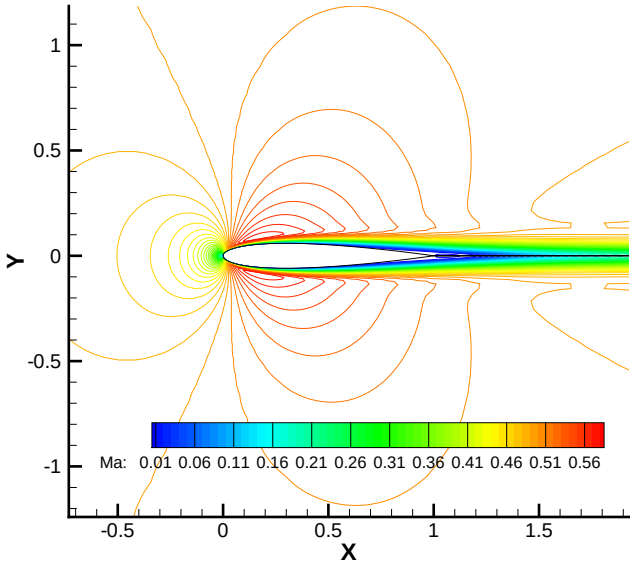
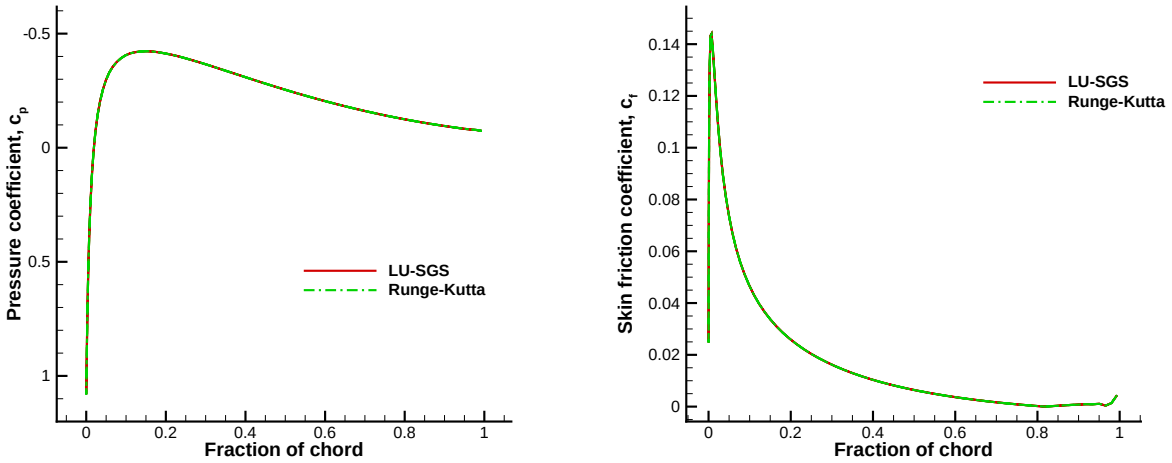


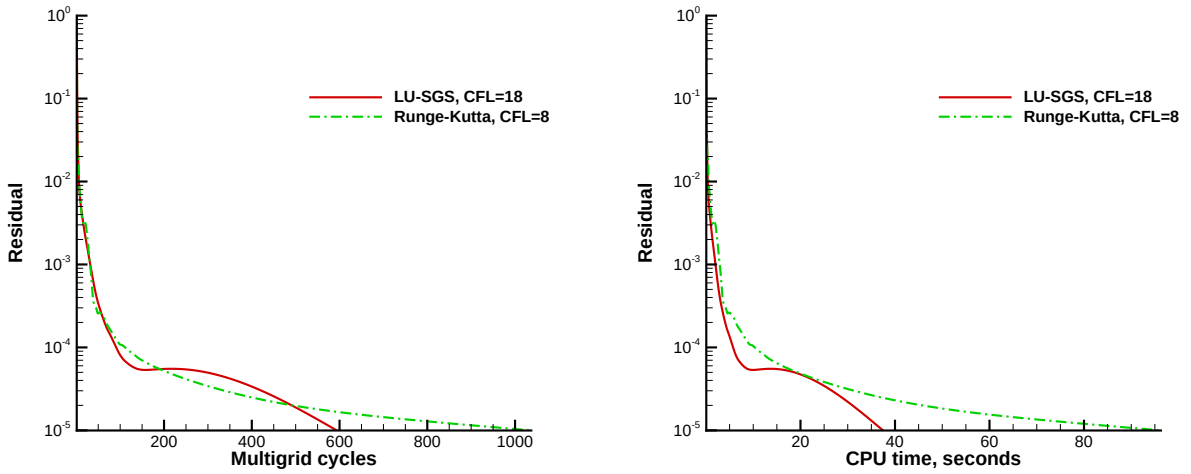
Figure 4.8: NACA0012 airfoil, steady laminar flow, Mach number contours.



(a) Pressure coefficient distribution

(b) Skin friction coefficient distribution

Figure 4.9: NACA0012 airfoil, steady laminar flow, surface force coefficient.



(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.10: NACA0012 airfoil, steady laminar flow, convergence history.

Kutta solver) in laminar flow computations. This is due to the fact that the viscous fluxes are evaluated only once in LU-SGS solver on every grid level, while the explicit Runge-Kutta solver usually does it at least three times. The convergence history of this test case also shows that the modification on the original LU-SGS scheme for Euler equations to include viscous effects is successful. The improvement on computational efficiency using LU-SGS solver is maintained.

4.2.3 Laminar Flow over a Circular Cylinder

The laminar flow over a circular cylinder at $Re = 40$ has been extensively studied in experiments and computations. It serves as a good test case for numerical methods, and hence is selected as the third test case for the present LU-SGS solver.

The sketch of the flow field is shown in Figure 4.11. Key parameters that quantitatively describing flow features, such as the length of the wake L, distance between two vortices b, distance from any vortex core to the trailing edge of the cylinder a and the separation angle

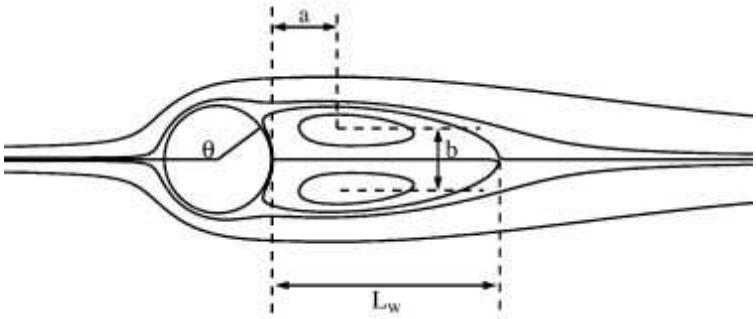
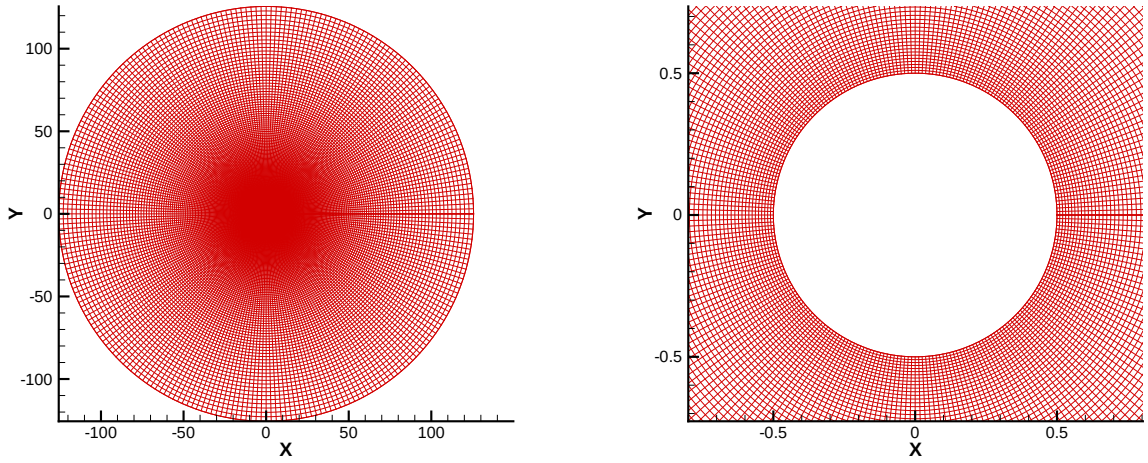


Figure 4.11: Sketch of laminar flow over a circular cylinder.



(a) 256×256 O mesh

(b) close view near airfoil surface

Figure 4.12: Circular cylinder,body-fitted mesh.

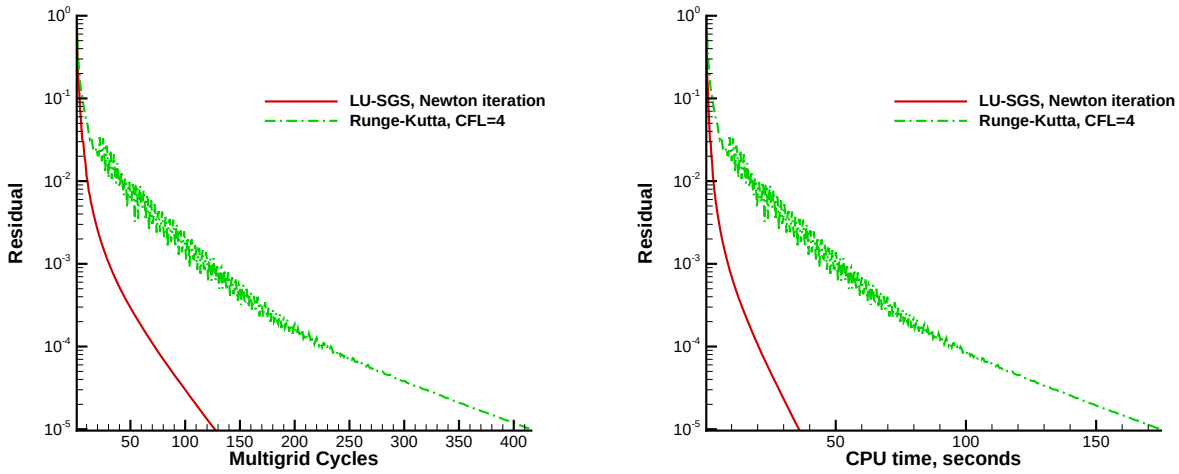
θ are labelled in the sketch. At this low Reynolds number, the laminar flow is steady.

The free-stream Mach number is set to be $M_\infty = 0.1$, which indicates the flow is essentially incompressible. However, the flow can still be computed using the present compressible flow solver.

A 257×257 O-type mesh is used, which is shown in Figure 4.12(a), while the close view near the surface is shown in Figure 4.12(b). The distance in the surface normal direction from the first grid point to the cylinder is 1×10^{-2}. The far field boundary of the mesh extends to approximately 100 times of the cylinder diameter away from the cylinder center. The

Table 4.1: Flow over circular cylinder, LU-SGS results

	C_d	θ	L_w/D	a/D	b/D
Tritton [42]	1.48				
Coutanceau & Bouard [43]		126.2°	2.13	0.76	0.59
Dennis & Chang [44]	1.52	126.2°	2.35		
Fornberg [45]	1.50	124.4°	2.24		
Gautier et al. [46]	1.49	126.4°	2.24	0.71	0.59
Present LU-SGS	1.51	126.3°	2.21	0.71	0.60
explicit Runge-Kutta	1.51	126.3°	2.21	0.71	0.60



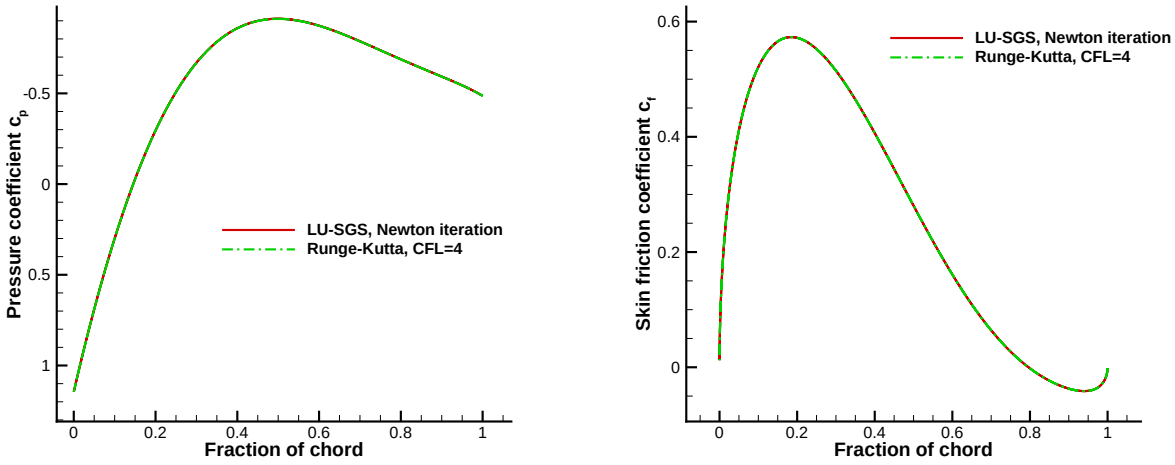
(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.13: Circular cylinder, steady laminar flow, convergence history.

computational results using the present LU-SGS scheme are compared to those using the explicit Runge-Kutta solver, as well as results of experiments and computations in Table 4.1. It can be seen that the present computations can precisely capture key flow field features and match results in literature very well.

The convergence history using the present LU-SGS solver is compared with that for the explicit Runge-Kutta solver in Figure 4.13. It can be observed that computation using LU-SGS solver converges to the criteria of 1×10^{-5} much faster than that using Runge-Kutta solver. The number of multigrid cycles for the latter to converge is about 3.26 times of that



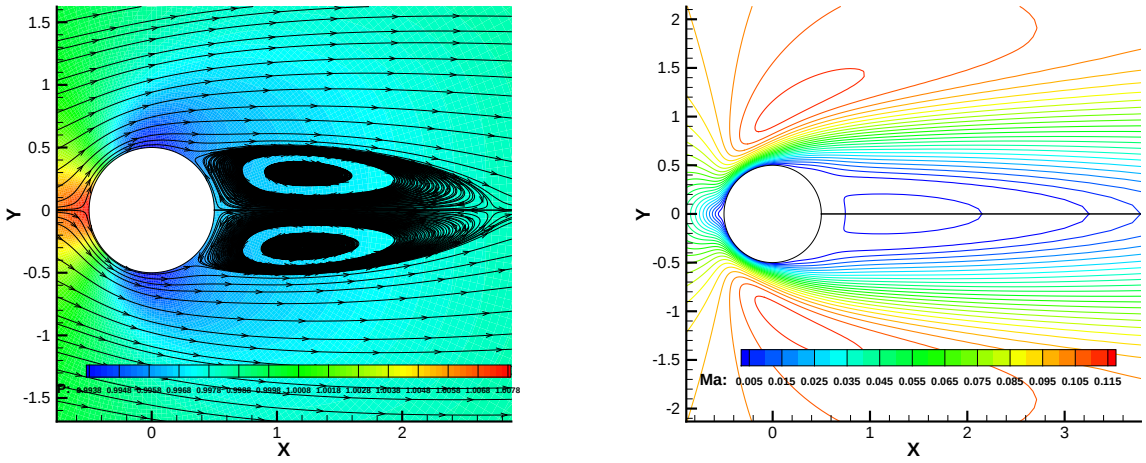
(a) Pressure coefficient distribution

(b) Skin friction coefficient distribution

Figure 4.14: Circular cylinder, steady laminar flow, surface force coefficient.

needed for the former. When CPU time is compared, this ratio goes up to approximately 5, which means LU-SGS solver still saves considerable CPU time during each multigrid cycle. The computational results confirm the conclusion that is made in the previous laminar flow test case. The use of the present LU-SGS solver can greatly reduce computational cost relative to explicit Runge-Kutta solver in laminar flow computations. Even for the flows at low Reynolds number, the efficiency advantage of using present LU-SGS is still obvious.

The computed surface pressure coefficient and skin friction coefficient distributions are shown in Figure 4.14. It can be observed that computations using the two solvers both converge to the same solution. Contours of the flow field and streamlines are shown in Figure 4.15. Two symmetric attached vortices can be seen behind the cylinder. This could explain why in Figure 4.14(b), the surface skin friction near the trailing edge is negative. This is because reverse flows occur in that area due to the attached vortices.



(a) Pressure contours and streamlines

(b) Mach number contours

Figure 4.15: Circular cylinder, steady laminar flow, flow field contours.

4.3 The Block-Jacobi Implicit Algorithm

For the unsteady problems, implicit schemes based on the original LU-SGS scheme can be developed to work with the Time Spectral Method. The block-Jacobi implicit algorithm proposed by Frederic Sicot et al. [24] is a representative one of this kind. The algorithm was applied with the Fourier time spectral method and reported to be very efficient in solving periodic flow problems, such as three-dimensional turbulent flow over a pitching wing.

To derive the Block-Jacobi Implicit algorithm, the unsteady flow governing equations with application of Fourier time spectral method are written in semi-discrete form as

$$\Omega \frac{\Delta W_n}{\Delta t_n} = -[R(W_n^{q+1}) + \Omega D_n(\vec{W}^{q+1})], 0 \leq n < 2N+1 \tag{4.27}$$

where $\Delta W_n = W_n^{q+1} - W_n^q$ is the solution update on the nth real time level, $\vec{W}$ is the solution vector composed of solutions on all real time levels and N the number of resolved Fourier modes. Following the original derivation, odd number of real time levels $(2N+1)$ is used. D_n is the Fourier time spectral operator for the nth real time level and can be specified

through

$$D_n(\vec{W}^{q+1}) = \sum_{m=0}^{2N+1} F_{n,m} W_m^{q+1} \tag{4.28}$$

where $F_{n,m}$ represents element of the Fourier time spectral operator matrix.

In equation (4.27), the steady residual evaluated at step $q+1$ can be linearised at step q as

$$R(W_n^{q+1}) = R(W_n^q) + J_n \Delta W_n + O((\Delta W_n)^2) \tag{4.29}$$

Since the Fourier time spectral operator is linear, the real time derivative term approximated at step q+1 and be calculated at step q as

$$D_n(\vec{W}^{q+1}) = D_n(\vec{W}^q) + D_n(\Delta \vec{W}) \tag{4.30}$$

Then equation (4.27) can be rewritten as

$$(\frac{\Omega}{\Delta t_n} I + J_n)\Delta W_n + \Omega D_n(\Delta \vec{W}) = -R_{TSM}(W_n^q), 0 \leq n < 2N+1 \tag{4.31}$$

where $R_{TSM}(W_n^q)$ is the explicit unsteady residual evaluated at step q. Equation (4.31) for all real time levels can be listed together and written in the form of $\mathbf{Ax} = \mathbf{b}$,where $\mathbf{x}$ and $\mathbf{b}$ are given by

$$\begin{aligned} \mathbf{x} &= [\Delta W_0, \ldots, \Delta W_{n-1}, \Delta W_n, \Delta W_{n+1}, \ldots, \Delta W_{2N+1}]^T \\ \mathbf{b} &= [R_{TSM}(W_0^q), \ldots, R_{TSM}(W_{n-1}^q), R_{TSM}(W_n^q), R_{TSM}(W_{n+1}^q), \ldots, R_{TSM}(W_{2N+1}^q)]^T \end{aligned} \tag{4.32}$$

The matrix $\mathbf{A}$ can be assembled as

$$\mathbf{A} = \begin{pmatrix} \frac{\Omega}{\Delta t_0}I + J_0 & \dots & \Omega F_{0,n-1} & \Omega F_{0,n} & \Omega F_{0,n+1} & \dots & \Omega F_{0,2N+1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \Omega F_{n-1,0} & \dots & \frac{\Omega}{\Delta t_{n-1}}I + J_{n-1} & \Omega F_{n-1,n} & \Omega F_{n-1,n+1} & \dots & \Omega F_{n-1,2N+1} \\ \Omega F_{n,0} & \dots & \Omega F_{n,n-1} & \frac{\Omega}{\Delta t_n}I + J_n & \Omega F_{n,n+1} & \dots & \Omega F_{n,2N+1} \\ \Omega F_{n+1,0} & \dots & \Omega F_{n+1,n-1} & \Omega F_{n+1,n} & \frac{\Omega}{\Delta t_{n+1}}I + J_{n+1} & \dots & \Omega F_{n+1,2N+1} \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Omega F_{2N+1,0} & \dots & \Omega F_{2N+1,n-1} & \Omega F_{2N+1,n} & \Omega F_{2N+1,n+1} & \dots & \frac{\Omega}{\Delta t_{2N+1}}I + J_{2N+1} \end{pmatrix} \tag{4.33}$$

It can be seen this matrix is block full and it could be split in three parts as $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$. Then a standard block-Jacobi iteration algorithm can be constructed as

$$\mathbf{D}\mathbf{x}^{l+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{l} \tag{4.34}$$

Specifying this equation on each real time level and notice $F_{n,n} = 0$, the following equation can be obtained

$$(\frac{\Omega}{\Delta t_n}I + J_n)\Delta W_n^{l+1} = -R_{TSM}(W_n^q) - \Omega D_n(\Delta\vec{W}^l), 0 \leq n < 2N+1 \tag{4.35}$$

where $\Delta\vec{W}^l = [\Delta W_0^l, \dots, \Delta W_{n-1}^l, \Delta W_n^l, \Delta W_{n+1}^l, \dots, \Delta W_{2N+1}^l]^T$

In equation 4.35, the solution update at block-Jacobi step l is known, while that at step $l+1$ is unknown and needs to be solved. Though the real time derivative term approximated by Fourier time spectral operator is initially included in the original full block implicit operator, its contribution is removed from the new reduced implicit operator in the block-Jacobi algorithm. The term representing this contribution, which is $\Omega D_n(\Delta\vec{W}^l)$ in equation (4.35), serves as an known source term. In this case, at every block-Jacobi step, equation (4.35) reduces in the form of classical implicit scheme for steady problems. At this point

the original LU-SGS scheme for steady problems can be applied at each real time level of a block-Jacobi step. To ensure the strongest coupling through updating $\Omega D_n(\Delta\vec{W}^l)$, Sicot [24] proposed to carry out only one Gauss-Seidel sweep (on each real time level) at a block-Jacobi step and update the implicit coupling term $\Omega D_n(\Delta\vec{W}^l)$ before each Gauss-Seidel sweep. However, a block-Jacobi step with forward sweep should be followed by one with backward sweep to balance the successive sweeping procedure. So an even number of block-Jacobi steps $l_{max}(1 \leq l \leq l_{max})$ is suggested to be used. Then the successive sweeping procedure at each block-Jacobi step is given by forward sweep

$$(L_n + D_n)W_n^{l+1} = -U_n W_n^l - R_{TSM}(W_n^q) - \Omega D_n(\Delta\vec{W}^l), 0 \leq n < 2N+1 \tag{4.36}$$

if l is odd and backward sweep

$$(D_n + U_n)W_n^{l+1} = -L_n W_n^l - R_{TSM}(W_n^q) - \Omega D_n(\Delta\vec{W}^l), 0 \leq n < 2N+1 \tag{4.37}$$

if l is even. Finally, after l_{max} block-Jacobi steps, solution at pseudo-time step q is updated as $W_n^{q+1} = W_n^q + \Delta W_n^{l_{max}}$.

Though fast convergence and improved efficiency of unsteady computations were achieved using the Block-Jacobi implicit algorithm, the algorithm itself has the following drawbacks. First, the algorithm is constructed based on the fact that diagonal elements of Fourier time spectral operator $F_{n,n}$ are zero. This is only true for Fourier time spectral method. If a time spectral method does not use Fourier expansion, the diagonal elements of the corresponding time spectral operator are not necessarily all zero, such as in Chebyshev time spectral method. If that is the case, the Block-Jacobi implicit method may fail or at least cannot be applied in its original form. This means the Block-Jacobi implicit method does not work for different time spectral methods in a unified form.

Second, when solving the implicit system resulting from time spectral method, treatments in

space and time are separated for the Block-Jacobi implicit algorithm. This leads to the fact that equations and solutions in delta form in the implicit system must be listed according to different real time levels. Then the implicit operator is split into blocks for different real time levels; hence block-matrix operations have to be adopted in the algorithm. So it can be observed that distinguishing space and time in solving the implicit system makes the algorithm not simple and not straightforward.

Third, the use of LU-SGS sweeps is only confined on each real time level, or in other words, only in space. So the algorithm only guarantees the just updated information in space can be used as soon as it becomes available. The just updated information in time direction is not used in the same way. Actually the time spectral coupling terms of the solution update on all real time levels are calculated once at the beginning of each Block-Jacobi step and are frozen during that Block-Jacobi step. So within a Block-Jacobi step, solution update that is just obtained on real time level n and the earlier ones is not used immediately to calculate the time spectral coupling term on real time level $n+1$. This is the well-known feature of a Jacobi iteration method. Clearly the Jacobi iteration method does not offer the fastest convergence.

4.4 The Space-Time LU-SGS Scheme for the Fourier Time Spectral Method and its Formulation

To make improvements on the drawbacks of Block-Jacobi implicit algorithm, a new space-time LU-SGS (ST-LU-SGS) scheme for time spectral method is proposed in this dissertation. This method is not based on specific features of a particular time spectral method, such as the diagonal elements of a Fourier time spectral operator being all zero. Therefore it can be applied to any time spectral method in a unified form. Most importantly, time is regarded as

an additional dimension in space in the ST-LU-SGS scheme; hence treatments in time and space for solving the implicit system resulting from time spectral method are not separated. The original LU-SGS scheme for a steady-state problem is modified such that the Gauss-Seidel sweeps are implemented not only in space but also in time. In this way the just updated information in both space and time can be used immediately as soon as it becomes available. This newly proposed scheme is simpler in algorithm and has the potential to make computation more stable and converge not slower than the Block-Jacobi implicit algorithm.

To develop the ST-LU-SGS scheme, the unsteady Euler equations are used for the sake of simplicity. After integration in two-dimensional space and applying the Fourier time spectral method, equation (4.3) can be modified as

$$\Omega_{i,j}\frac{w_{i,j,l}^{q+1}-w_{i,j,l}^{q}}{\Delta t}+\sum_{m=1}^{4}[\mathbf{F}(w_{m,l}^{q+1})\cdot\vec{S}_m]+\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}w_{i,j,n}^{q+1}=0 \tag{4.38}$$

where $F_{l,n}$ denotes element of the Fourier time spectral operator with explicit flow frequency inside, l the index number of real time level and N the total number of real time levels used in time spectral method. In equation (4.38), since the time spectral operator is a linear, the real time derivative approximated by it at pseudo-time step $q+1$ can be easily evaluated at step q as

$$\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}w_{i,j,n}^{q+1}=\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}w_{i,j,n}^{q}+\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}\delta w_{i,j,n} \tag{4.39}$$

Substituting the above relation back and let $\beta=1$ as usual, equation (4.5) is revised as

$$\Omega_{i,j}\delta w_{i,j,l}+\Delta t\sum_{m=1}^{4}\mathbf{A}_{m,l}\Delta S_m\delta w_{m,l}+\Delta t\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}\delta w_{i,j,n}+\Delta t\tilde{R}_{i,j,l}=0 \tag{4.40}$$

where $\tilde{R}_{i,j,l}=\sum_{m=1}^{4}[\mathbf{F}(w_{m,l}^{q})\cdot\vec{S}_m]+\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}w_{i,j,n}^{q}$ is the explicit unsteady residual estimated at pseudo-time step q. In equation (4.40), the third term can be split into three parts, one for the real time levels in the past ($n<l$), one for the current real time level ($n=l$) and

the last one for the real time levels in the future ($n > l$) as

$$\Delta t\Omega_{i,j}\sum_{n=0}^{N-1}F_{l,n}\delta w_{i,j,n} = \Delta t\Omega_{i,j}\sum_{n=0}^{l-1}F_{l,n}\delta w_{i,j,n}+\Delta t\Omega_{i,j}F_{l,l}\delta w_{i,j,l}+\Delta t\Omega_{i,j}\sum_{n=l+1}^{N-1}F_{l,n}\delta w_{i,j,n} \quad (4.41)$$

In the revised implicit operator, combining equation (4.41) and the original treatment for space in the LU-SGS scheme for steady problem, equation (4.14) can be rewritten as

$$\begin{aligned} &D\delta w_{i,j,l} + \Delta t[-\tilde{\mathbf{A}}^{+}_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w_{i-1,j,l} - \tilde{\mathbf{A}}^{+}_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w_{i,j-1,l} + \Omega_{i,j}\sum_{n=0}^{l-1}F_{l,n}\delta w_{i,j,n} \\ &+ \hat{\mathbf{A}}^{-}_{i+1,j,l}\Delta S_{i+\frac{1}{2},j,l}\delta w_{i+1,j,l} + \hat{\mathbf{A}}^{-}_{i,j+1,l}\Delta S_{i,j+\frac{1}{2},l}\delta w_{i,j+1,l} + \Omega_{i,j}\sum_{n=l+1}^{N-1}F_{l,n}\delta w_{i,j,n}] + \Delta t\tilde{R}_{i,j,l} = 0 \end{aligned} \quad (4.42)$$

where

$$\begin{aligned} D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l}n^{I}_{x,i,j,l} + \bar{v}_{i,j,l}n^{I}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{I}_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l}n^{J}_{x,i,j,l} + \bar{v}_{i,j,l}n^{J}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{J}_{i,j,l} \\ &+ \Delta t\Omega_{i,j}F_{l,l} \end{aligned} \quad (4.43)$$

In equation (4.43), terms due to Fourier time spectral operator are added to the implicit operator. Among these terms, the coefficient in front of $\delta w_{i,j,l}$ appears in the diagonal elements of the modified implicit operator. Coefficients of $\delta w_{i,j,n}(0 < n < l)$ will be found in the lower triangular part, while those for $\delta w_{i,j,n}(l < n < N-1)$ will be in the upper triangular part. It can be shown that if the modified implicit operator matrix is assembled, it will be an augmented matrix corresponding to solution update on all spatial points and all real time levels. Unlike the banded matrix for steady problem, this matrix is full.

Based on equation (4.43), a symmetric Gauss-Seidel sweeping procedure similar to that for steady problems can be constructed as follows.

Forward sweep

$$\begin{aligned}\delta w^*_{i,j,l} = \frac{\Delta t}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \Omega_{i,j}\sum_{n=0}^{l-1} F_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.44}$$

Backward sweep

$$\begin{aligned}\delta w^{(1)}_{i,j,l} = \frac{\Delta t}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \hat{\mathbf{A}}^-_{i+1,j,l}\Delta S_{i+\frac{1}{2},j,l}\delta w^{(1)}_{i+1,j,l} - \hat{\mathbf{A}}^-_{i,j+1,l}\Delta S_{i,j+\frac{1}{2},l}\delta w^{(1)}_{i,j+1,l} \\ &- \Omega_{i,j}\sum_{n=l+1}^{N-1} F_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.45}$$

where

$$\begin{aligned}D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l}n^I_{x,i,j,l} + \bar{v}_{i,j,l}n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l}n^J_{x,i,j,l} + \bar{v}_{i,j,l}n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Delta t\Omega F_{l,l}\end{aligned} \tag{4.46}$$

When Δt goes to infinity, the above scheme reduces to Newton iteration as:

Forward sweep

$$\begin{aligned}\delta w^*_{i,j,l} = \frac{1}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \Omega_{i,j}\sum_{n=0}^{l-1} F_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.47}$$

Backward sweep

$$\begin{aligned}\delta w^{(1)}_{i,j,l} = \frac{1}{D}[&\tilde{\mathbf{A}}^{+}_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^{*}_{i-1,j,l} + \tilde{\mathbf{A}}^{+}_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^{*}_{i,j-1,l} \\ &- \hat{\mathbf{A}}^{-}_{i+1,j,l}\Delta S_{i+\frac{1}{2},j,l}\delta w^{(1)}_{i+1,j,l} - \hat{\mathbf{A}}^{-}_{i,j+1,l}\Delta S_{i,j+\frac{1}{2},l}\delta w^{(1)}_{i,j+1,l} \\ &- \Omega_{i,j}\sum_{n=l+1}^{N-1} F_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.48}$$

where

$$\begin{aligned}D =&(\left|\bar{u}_{i,j,l}n^{I}_{x,i,j,l} + \bar{v}_{i,j,l}n^{I}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{I}_{i,j,l} \\ &+ (\left|\bar{u}_{i,j,l}n^{J}_{x,i,j,l} + \bar{v}_{i,j,l}n^{J}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{J}_{i,j,l} \\ &+ \Omega F_{l,l}\end{aligned} \tag{4.49}$$

When the ST-LU-SGS scheme is applied to viscous flows, treatment on viscous fluxes in implicit operator for steady problems is adopted in this dissertation. In this way the implicit operator still remains diagonally dominant. Hence convergence of the symmetric Gauss-Seidel iterations is guaranteed. So similar to equation (4.26), equations (4.46) and (4.49) can be revised as

$$\begin{aligned}D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l}n^{I}_{x,i,j,l} + \bar{v}_{i,j,l}n^{I}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{I}_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l}n^{J}_{x,i,j,l} + \bar{v}_{i,j,l}n^{J}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{J}_{i,j,l} \\ &+ \Delta t\Omega F_{l,l} + \Delta t\frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^{I}_{i,j,l})^2 + (\Delta S^{J}_{i,j,l})^2}{\Omega_{i,j}}\end{aligned} \tag{4.50}$$

for the general form of the full LU-SGS scheme and

$$\begin{aligned}D =&(\left|\bar{u}_{i,j,l}n^{I}_{x,i,j,l} + \bar{v}_{i,j,l}n^{I}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{I}_{i,j,l} \\ &+ (\left|\bar{u}_{i,j,l}n^{J}_{x,i,j,l} + \bar{v}_{i,j,l}n^{J}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{J}_{i,j,l} \\ &+ \Omega F_{l,l} + \frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^{I}_{i,j,l})^2 + (\Delta S^{J}_{i,j,l})^2}{\Omega_{i,j}}\end{aligned} \tag{4.51}$$

for the full LU-SGS scheme in Newton iteration form if the scheme is still stable using

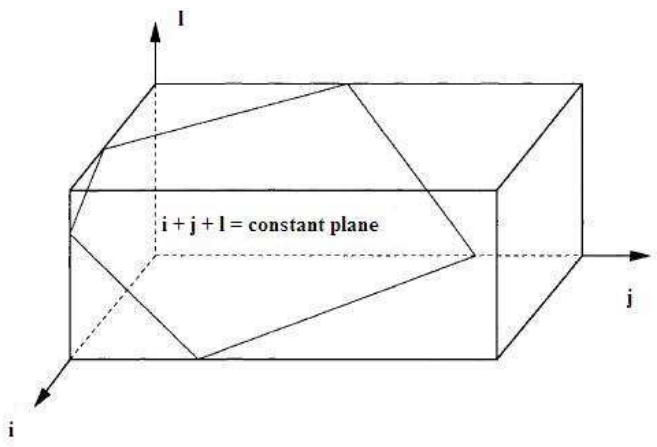


Figure 4.16: Sweeping plane of the ST-LU-SGS scheme in computational domain[39].

infinitely large pseudo-time step.

In the newly proposed ST-LU-SGS scheme, the sweeping order is similar to that in the standard LU-SGS scheme for three-dimensional steady problems. This is a natural result of regarding the time domain as an additional dimension in space. In the ST-LU-SGS scheme, the points that are currently swept are all on a diagonal plane($i + j + l =$ constant) other than a diagonal line. One of such a diagonal sweeping plane is illustrated in Figure 4.16. Then computations of solution update δw on all points over the same diagonal plane can be easily vectorised.

Another issue about the newly proposed ST-LU-SGS scheme is its performance in modelling high frequency periodic flows. It has been reported, when applied with explicit Runge-Kutta solver, the convergence of time spectral method is significantly influenced if the flow frequency is high [19]. Since the ST-LU-SGS scheme is constructed on time spectral method, whether or not the convergence of this scheme is maintained for high frequency flows remains a problem. Actually, computational experiments show that if the frequency of a flow is sufficiently high, computations using the ST-LU-SGS scheme in the form of equations (4.44),(4.45) and (4.46) cannot converge. One possible reason for this problem is that, at high frequency, the implicit

operator of the ST-LU-SGS scheme is no longer diagonally dominant. To investigate further, it is worthwhile to examine the Fourier time spectral operator first. For the sake of simplicity, the Fourier time spectral operator using four real time levels is

$$F = \omega \begin{pmatrix} 0 & 0.5 & 0 & -0.5 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 \\ 0.5 & 0 & -0.5 & 0 \end{pmatrix} \tag{4.52}$$

where ω is angular frequency. It can be seen that in this operator, all diagonal elements $F_{l,l}$ are zero. When incorporated in the implicit operator of the ST-LU-SGS scheme, these zero coefficients are added to the diagonal elements. Those non-zero coefficients of the Fourier time spectral operator will be added to the off-diagonal elements in the implicit operator. Note that these non-zero coefficients are directly proportional to the frequency of the flow field. So when the flow frequency is sufficiently high, those off-diagonal elements in the implicit operator could become equal or even larger than the diagonal elements, which are summations of spectral radius of local convective Jacobian matrix and viscous Jacobian matrix. Hence, the convergence of Gauss-Seidel iterations is no longer guaranteed. Inspired by the way how contribution of viscous fluxes is added to the implicit operator for viscous flow. A possible solution to the convergence problem at high flow frequency could be replacing $F_{l,l}$ with the following term to the diagonal elements of the implicit operator:

$$F_{l,max} = max(F(l,n)), 0 \leq n \leq N-1 \tag{4.53}$$

It can be shown that the modified implicit operator is still diagonally dominant even at high flow frequency. So the convergence of the ST-LU-SGS scheme at high frequency could be improved. And at the same time the accuracy of the flow solution is not influenced since the it completely depends on the explicit residual other than the implicit operator. This

treatment is called high frequency modification in this dissertation. With this treatment, only equation (4.50) needs to be revised as

$$\begin{aligned} D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l}n^I_{x,i,j,l} + \bar{v}_{i,j,l}n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l}n^J_{x,i,j,l} + \bar{v}_{i,j,l}n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Delta t\Omega F_{l,max} + \Delta t\frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^I_{i,j,l})^2 + (\Delta S^J_{i,j,l})^2}{\Omega_{i,j}} \end{aligned} \tag{4.54}$$

in the original full LU-SGS scheme of the general form and only equation (4.51) need to be revised as

$$\begin{aligned} D = &(\left|\bar{u}_{i,j,l}n^I_{x,i,j,l} + \bar{v}_{i,j,l}n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ (\left|\bar{u}_{i,j,l}n^J_{x,i,j,l} + \bar{v}_{i,j,l}n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Omega F_{l,max} + \frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^I_{i,j,l})^2 + (\Delta S^J_{i,j,l})^2}{\Omega_{i,j}} \end{aligned} \tag{4.55}$$

for the Newton iteration form if infinitely large pseudo-time step is allowed. This treatment will be tested in high frequency flow test case in next section.

4.5 Periodic Flow Test Cases Using the ST-LU-SGS Scheme for the Fourier Time Spectral Method

To validate the newly proposed ST-LU-SGS scheme for time spectral method and show its advantage in improving computational efficiency, results of the following test cases are presented and discussed.

4.5.1 Inviscid Flow over a Pitching NACA0012 Airfoil

The periodic flow over a pitching NACA0012 airfoil at transonic flow conditions is frequently used as a test case for unsteady solvers. In this test case, a NACA0012 airfoil is forced to pitch around its quarter chord at $M_\infty = 0.755, Re = 5.5 \times 10^6$. The forced pitching movement is given by

$$\alpha = \alpha_0 + \alpha_m sin\omega t \tag{4.56}$$

where α_0 is mean angle of attack, α_m the pitching amplitude. ω is the angular frequency defined as

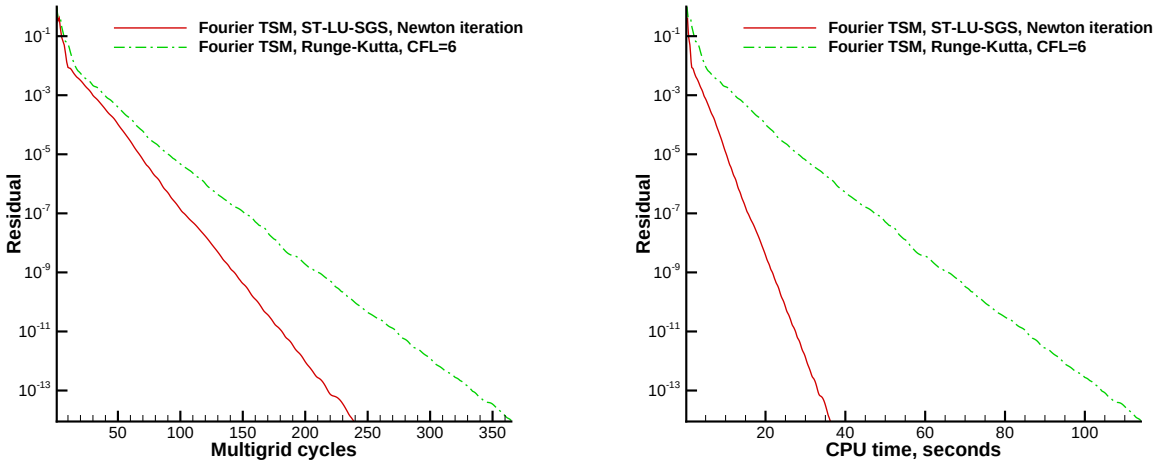
$$\kappa = \frac{\omega c}{2U_\infty} \tag{4.57}$$

where κ is the reduced frequency, c the chord length and U_∞ the free stream velocity. In this case, $\alpha_0 = 0.0°$, $\alpha_m = 2.51°$ and $\kappa = 0.0814$.

Although this test case is perfect to test turbulent flow solvers, it also serves as an good test case for numerical methods solving the Euler equations. This is because strong shock waves occur in this case and then can be simulated by solving Euler equations.

The same 128×32 C-type grid as in the previous steady inviscid flow problem is used in this test case again. To implement the time spectral method, eight real time levels are used. Computational results using the ST-LU-SGS scheme are compared with those of the Runge-Kutta solver. The convergence criteria is set to be 1×10^{-14}. Computational results show that the ST-LU-SGS scheme can be carried out in the way of Newton iteration method, which is equivalent to using infinitely large CFL number (hence infinitely large pseudo-time step) in the scheme for this inviscid case. This means the stability of the new scheme is well preserved when regarding the time domain as an additional dimension of space in the basic LU-SGS scheme.

The residual history with respect to multigrid cycles and CPU time is shown in Figure 4.17(a).



(a) Residual versus multigrid Cycles

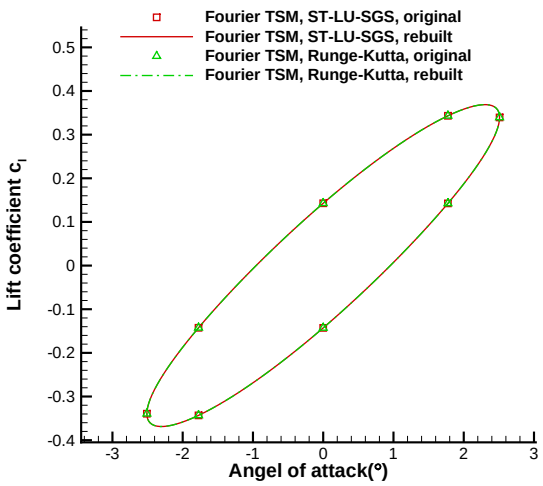
(b) Residual versus CPU time

Figure 4.17: Pitching NACA0012 airfoil, periodic inviscid flow, convergence history.

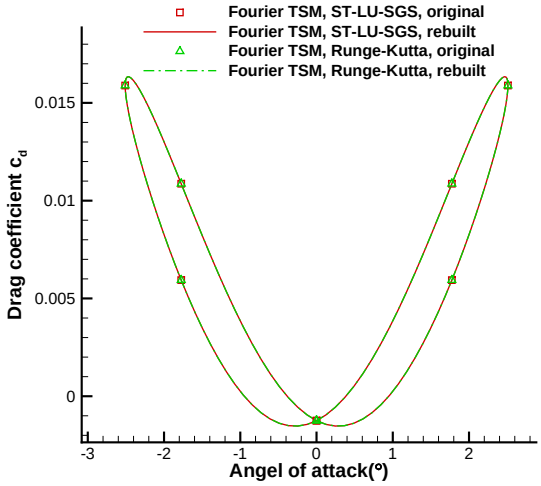
The number of multigrid cycles for the explicit Runge-Kutta solver to reach convergence is about 1.52 times of that for the ST-LU-SGS scheme. When CPU time is compared, this ratio goes up to 3.51. This can be observed in Figure 4.17(b). So nearly a half of the CPU time spent in a multigrid cycle can be saved if the explicit Runge-Kutta solver is replaced by the ST-LU-SGS scheme. This improvement on computational efficiency is obviously greater than that in the previous steady test case. The main reason is as follows. On each grid level in a multigrid cycle, the time spectral operator is applied five times in the explicit Runge-Kutta solver, whereas in the ST-LU-SGS scheme it is only applied twice, one in the explicit residual and the other in the implicit operator. This makes a multigrid cycle much less time-consuming in the ST-LU-SGS scheme.

The force coefficient variation with respect to angle of attack is shown in Figure 4.18. The symbols denote the original results on the 8 real time levels, while the lines represent the rebuilt results by adding the resolved Fourier modes together. It can be seen computations using the ST-LU-SGS scheme and explicit Runge-Kutta both converge to the same solution.

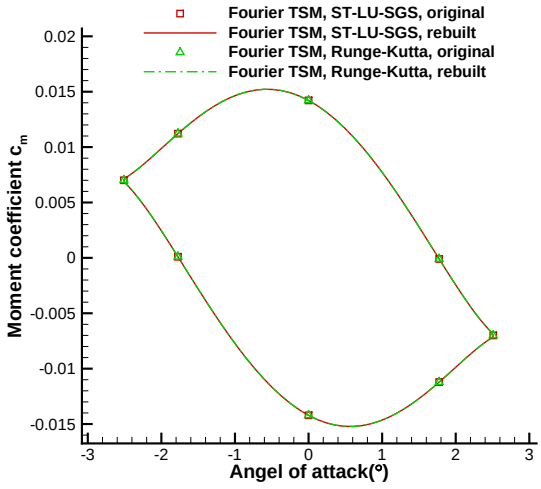
The temporal variation of surface pressure coefficient distribution is extensively shown in



(a) Lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 4.18: Pitching NACA0012 airfoil, periodic inviscid flow, force and moment coefficients versus angle of attack.

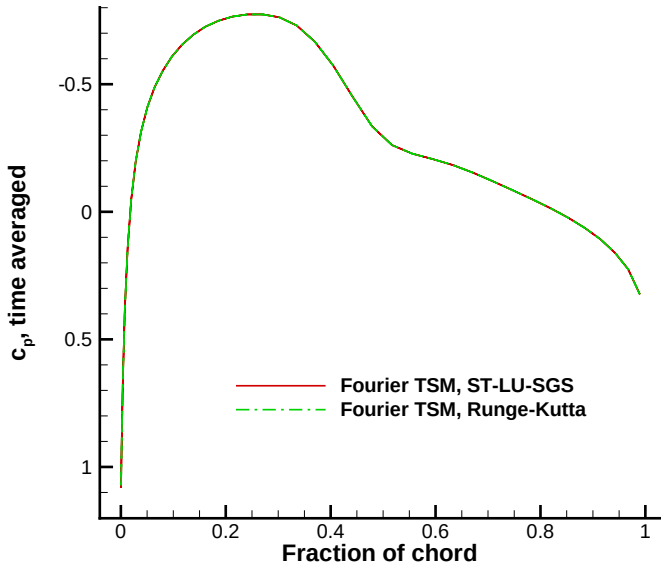
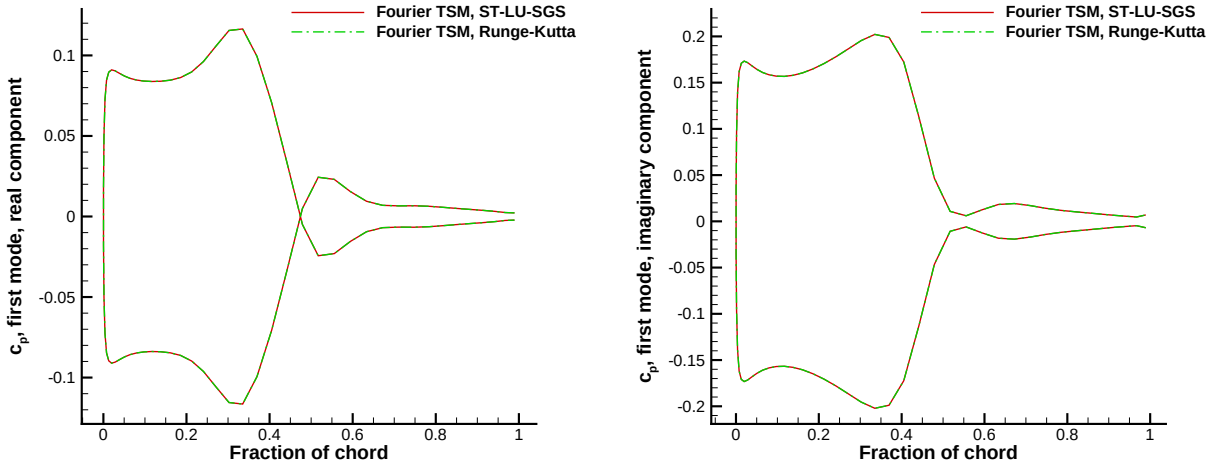


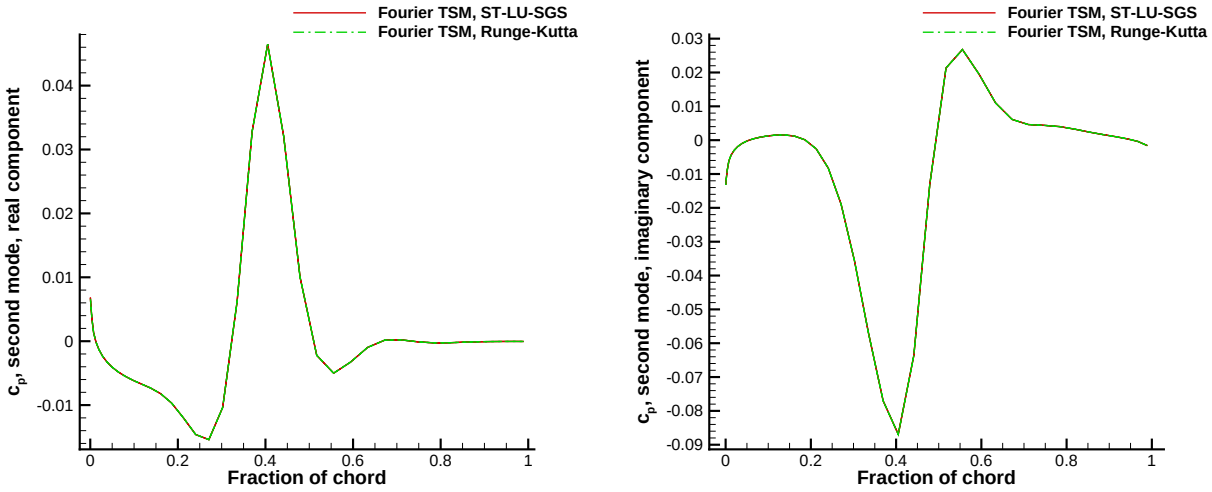
Figure 4.19: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, time averaged.



(a) real component

(b) imaginary component

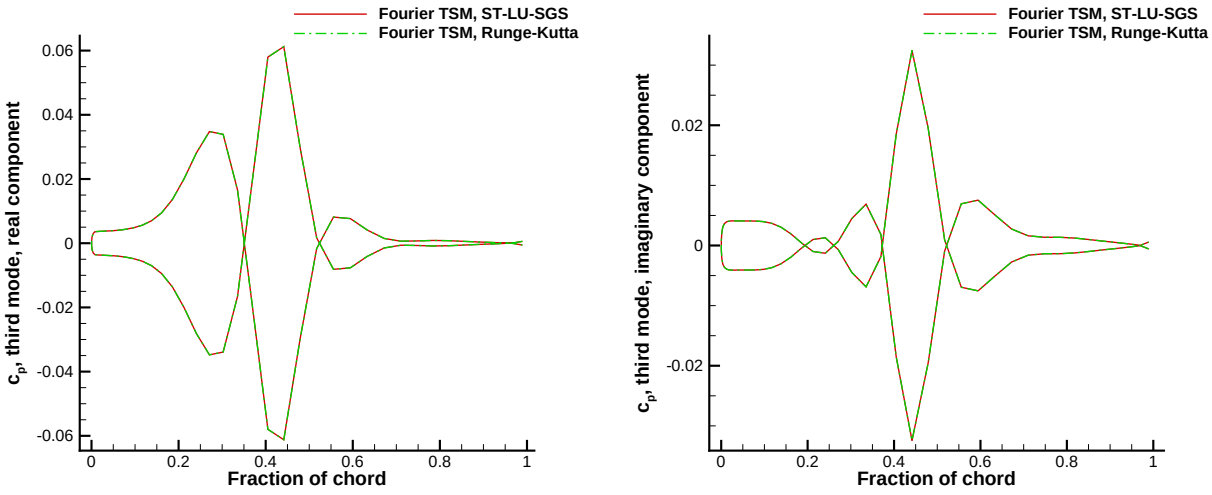
Figure 4.20: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, first mode.



(a) real component

(b) imaginary component

Figure 4.21: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, second mode.



(a) real component

(b) imaginary component

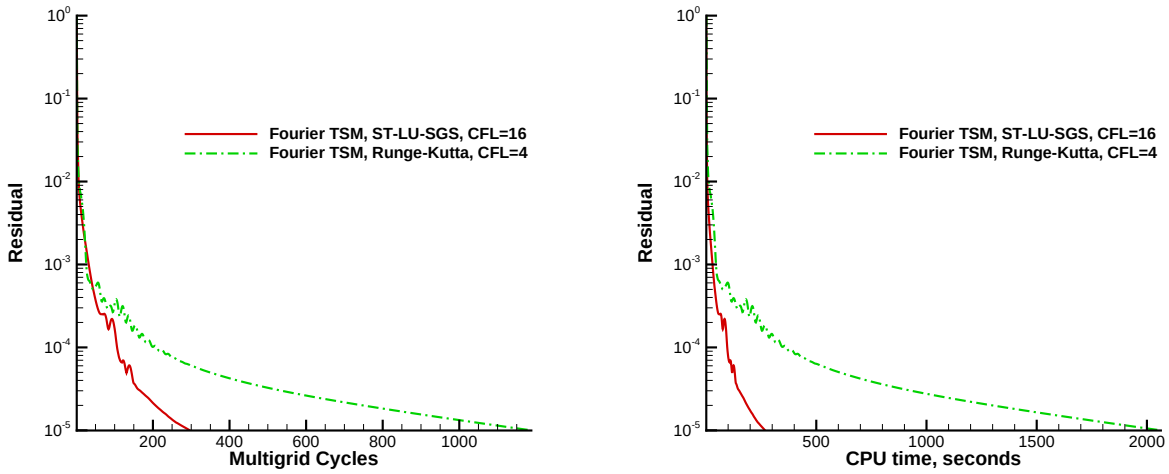
Figure 4.22: Pitching NACA0012 airfoil, periodic inviscid flow, surface pressure coefficient, third mode.

frequency domain. The Fourier transformed results of the time averaged component and the first three modes are shown in Figures 4.19, 4.20, 4.21 and 4.22. The results using ST-LU-SGS scheme match those by the explicit Runge-Kutta solver very well. Though the Mach number of this test case is not very high, strong shock waves occur due to the thick airfoil. As the airfoil pitches up and down, shock waves on both upper and lower surfaces move back and forth. This phenomenon indicates strong non-linearity. The high efficiency of the new ST-LU-SGS scheme is verified when simulating this kind of unsteady flows.

4.5.2 Laminar Flow over a Pitching NACA0012 Airfoil

Next the newly proposed ST-LU-SGS scheme is tested in cases where viscous effects are significant. The second test case is chosen to be the laminar flow over a pitching NACA0012 airfoil. In this case, the airfoil still pitches around its quarter chord with a smaller pitching amplitude $\alpha_m = 1.51^\circ$ and a smaller reduced frequency $\kappa = 0.05$. The mean angle of attack is still zero. The flow conditions are $M_\infty = 0.5$ and $Re = 5000$. At these conditions, the flow is laminar and shock free. The same 257×65 C-type grid in the previous steady laminar flow example is used. To accurately resolve the laminar boundary layer in this unsteady flow, the normal distance between the first grid point and the airfoil surface is set to be 1×10^{-4} as usual. When implementing the time spectral method, 8 real time levels are used. Computational results are still compared between the ST-LU-SGS solver and the Runge-Kutta solver. It is similar to the previous steady laminar flow that the CFL number can only achieve a finite large one, which is 16 in this case. This is because viscous fluxes only make contribution to the diagonal elements in the implicit operator. Since this is a viscous computation, the convergence criteria is set to be 1×10^{-5}.

The convergence history versus multigrid cycles is shown in Figure 4.23(a). To reach convergence, the explicit Runge-Kutta solver uses almost four times the multigrid cycles needed



(a) Residual versus Multigrid cycles

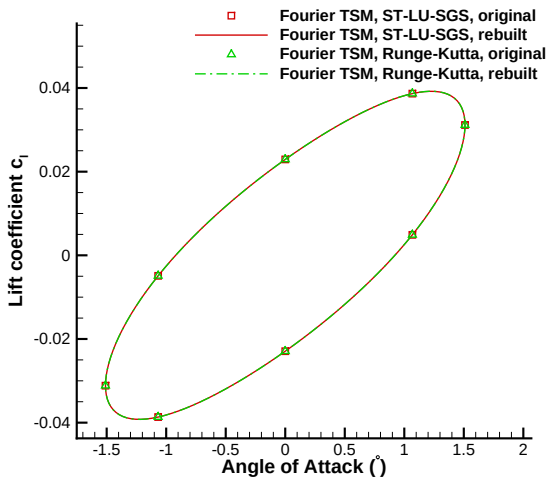
(b) Residual versus CPU time

Figure 4.23: Pitching NACA0012 airfoil, periodic laminar flow, convergence history.

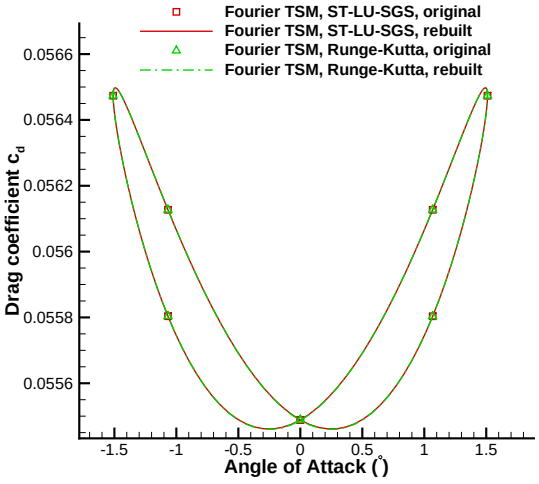
by the ST-LU-SGS scheme. Figure 4.23(b) shows this ratio is enlarged to over eight when CPU time is compared. In terms of CPU time consumption, the acceleration effect of considering time domain as an additional dimension of space in the basic LU-SGS scheme can be accumulated on the acceleration effect of properly involving viscous fluxes in the implicit operator.

The variation of the force coefficients with respect to angle of attack is shown in Figure 4.24. As before, symbols denote original results on the 8 real time levels, while the lines represent the rebuilt results by adding resolved Fourier modes together. It can be seen that computations using ST-LU-SGS scheme and explicit Runge-Kutta solver both converge to the same solution.

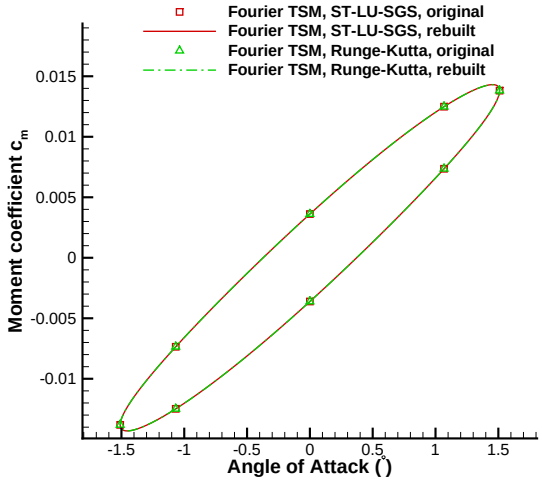
The temporal variation of surface pressure coefficient and skin friction coefficient distribution are Fourier transformed and shown in Figures 4.25- 4.32 for several Fourier modes. For every resolved Fourier mode, the solver using ST-LU-SGS scheme and the explicit Runge-Kutta solver reach the same result.



(a) Lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 4.24: Pitching NACA0012 airfoil, periodic laminar flow, force and moment coefficients versus angle of attack.

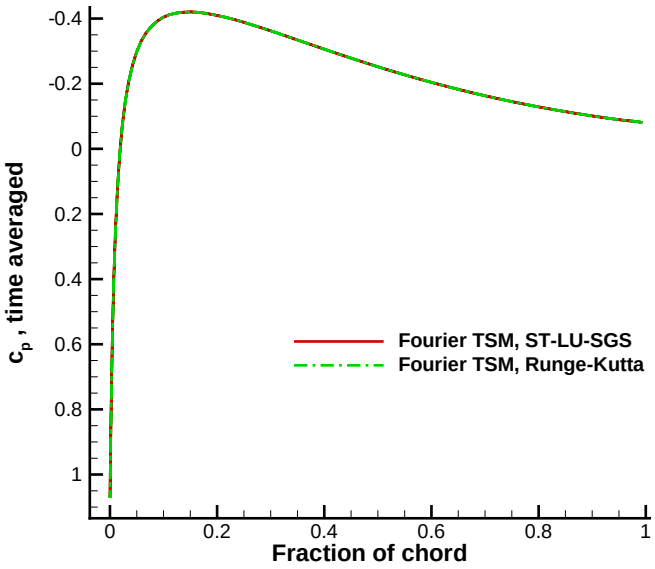
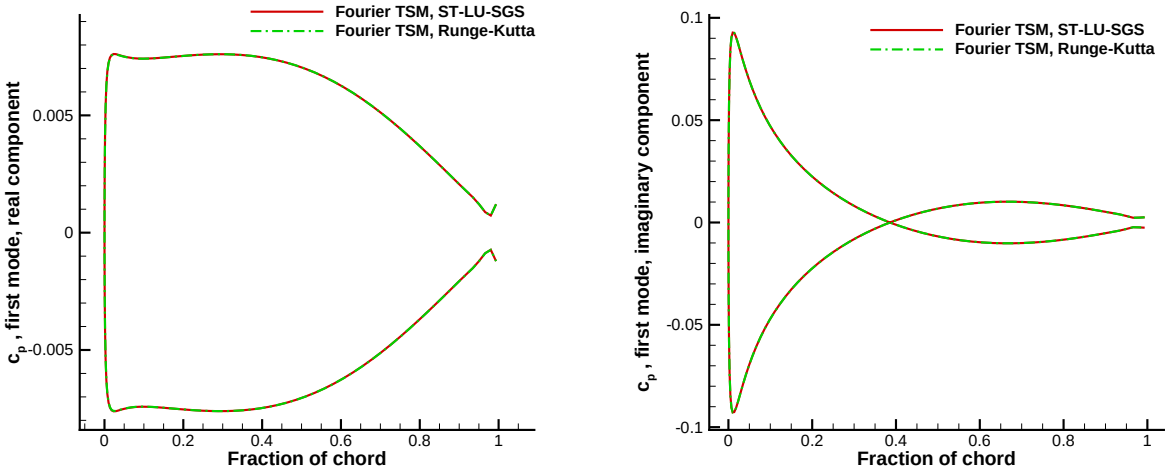


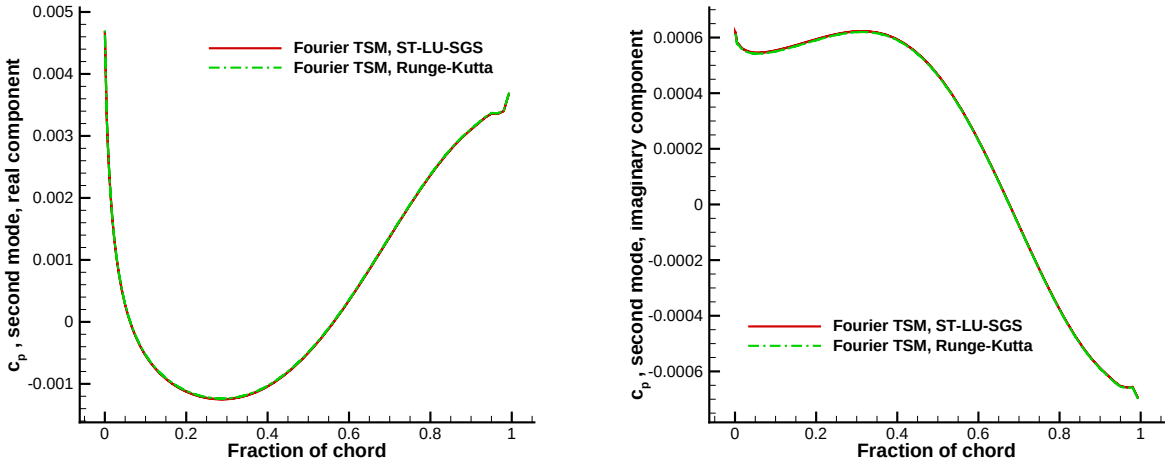
Figure 4.25: Pitching NACA0012 airfoil, periodic laminar flow, surface pressure coefficient, time averaged.



(a) real component

(b) imaginary component

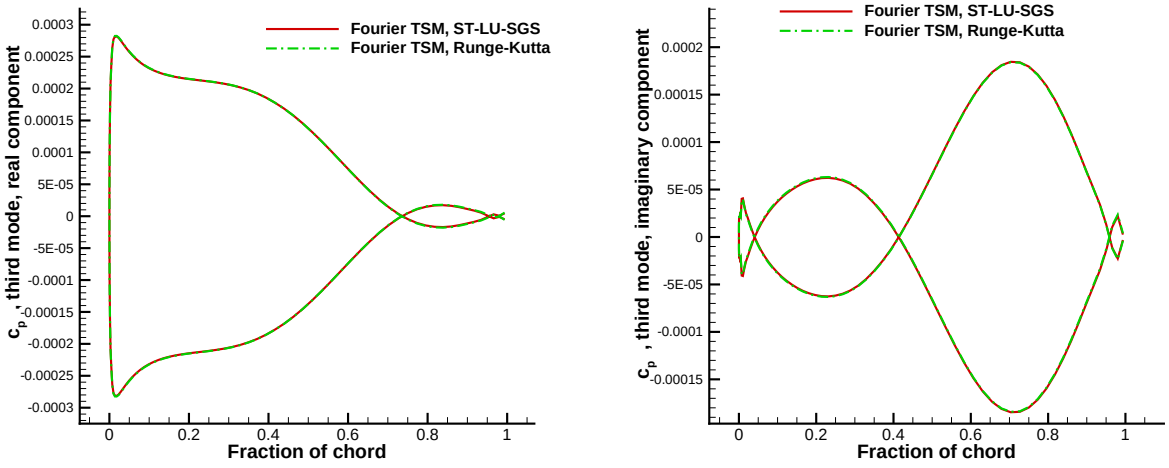
Figure 4.26: Pitching NACA0012 airfoil, periodic laminar flow, surface pressure coefficient, first mode.



(a) real component

(b) imaginary component

Figure 4.27: Pitching NACA0012 airfoil, periodic laminar flow, surface pressure coefficient, second mode.



(a) real component

(b) imaginary component

Figure 4.28: Pitching NACA0012 airfoil, periodic laminar flow, surface pressure coefficient, third mode.

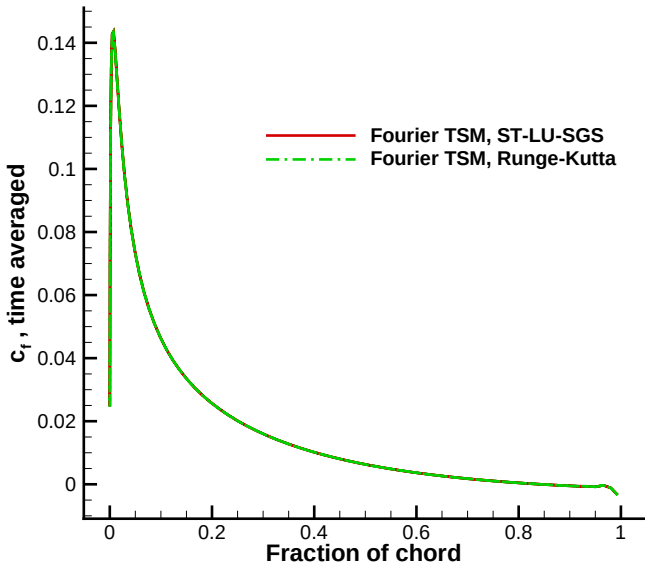
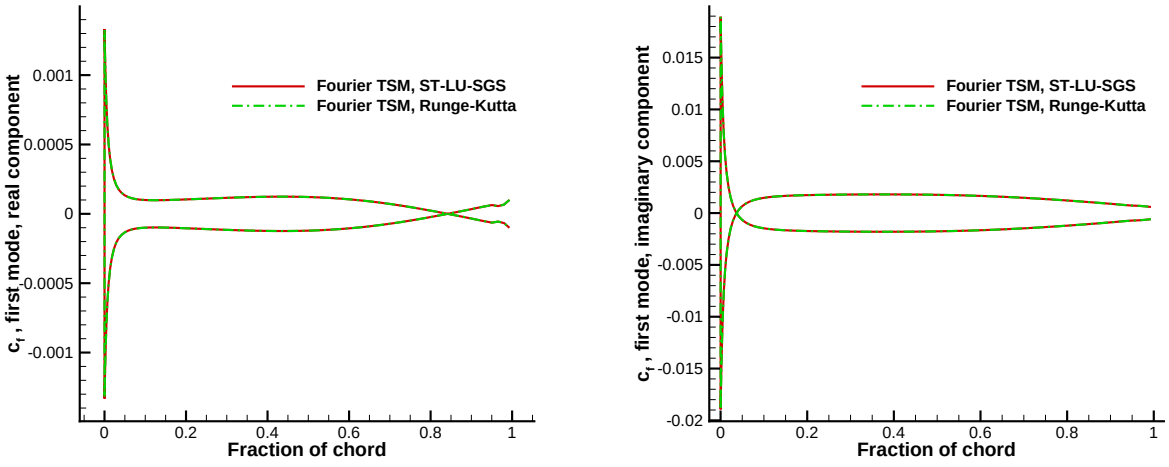


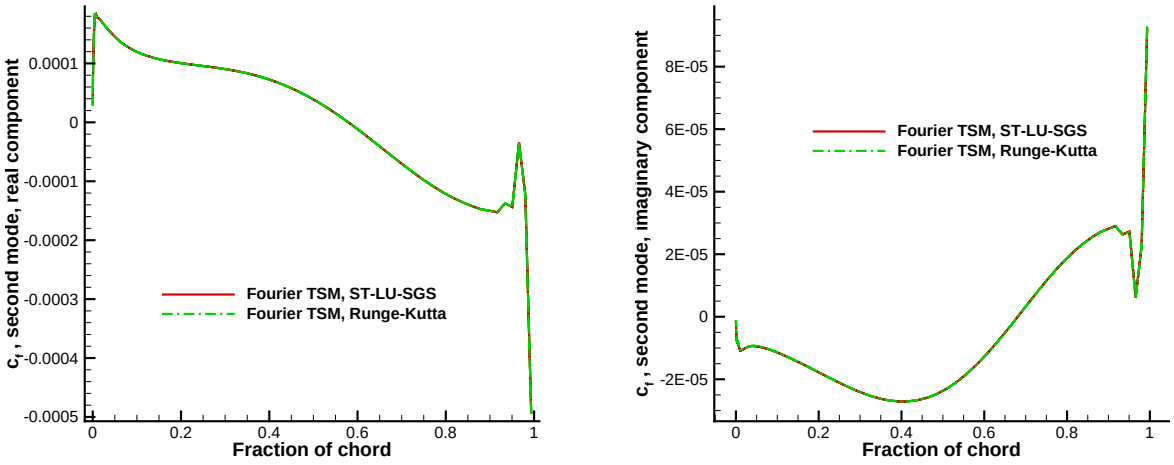
Figure 4.29: Pitching NACA0012 airfoil, periodic laminar flow, surface skin friction coefficient, time averaged.



(a) real component

(b) imaginary component

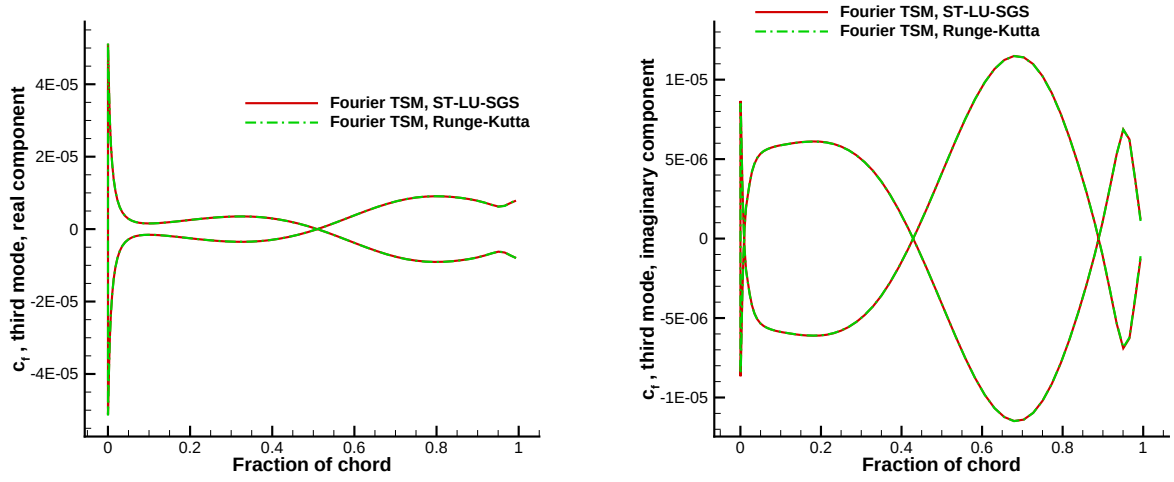
Figure 4.30: Pitching NACA0012 airfoil, periodic laminar flow, surface skin friction coefficient, first mode.



(a) real component

(b) imaginary component

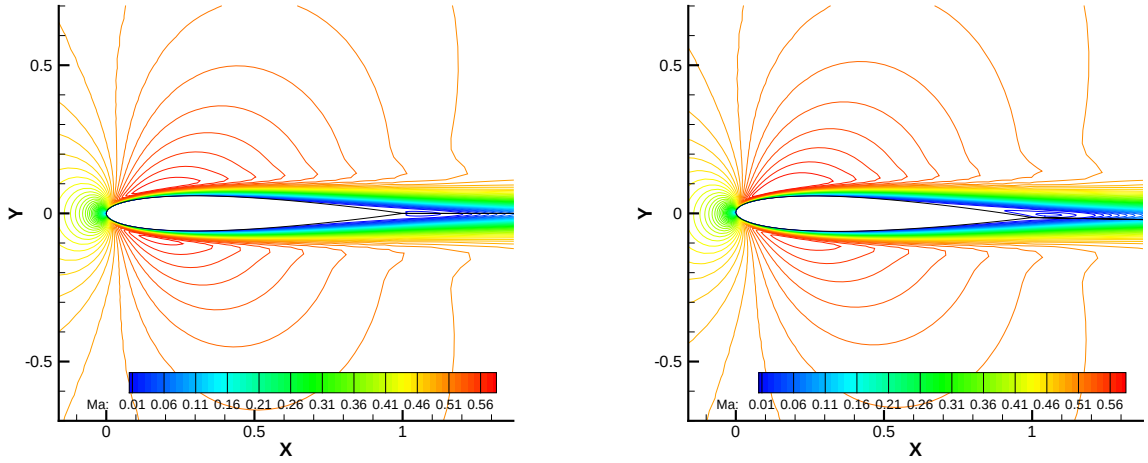
Figure 4.31: Pitching NACA0012 airfoil, periodic laminar flow, surface skin friction coefficient, second mode.



(a) real component

(b) imaginary component

Figure 4.32: Pitching NACA0012 airfoil, periodic laminar flow, surface skin friction coefficient, third mode.



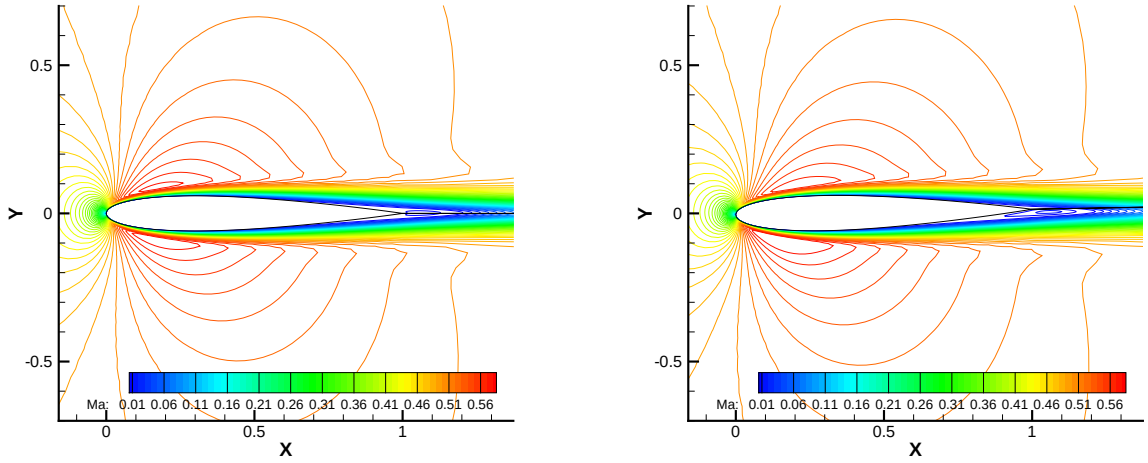
(a) $t = 0$

(b) $t = \frac{1}{8}T$

(c) $t = \frac{2}{8}T$

(d) $t = \frac{3}{8}T$

Figure 4.33: Pitching NACA0012 airfoil, periodic laminar flow, Mach number contours in the first half period.



(a) $t = \frac{4}{8}T$

(b) $t = \frac{5}{8}T$

(c) $t = \frac{6}{8}T$

(d) $t = \frac{7}{8}T$

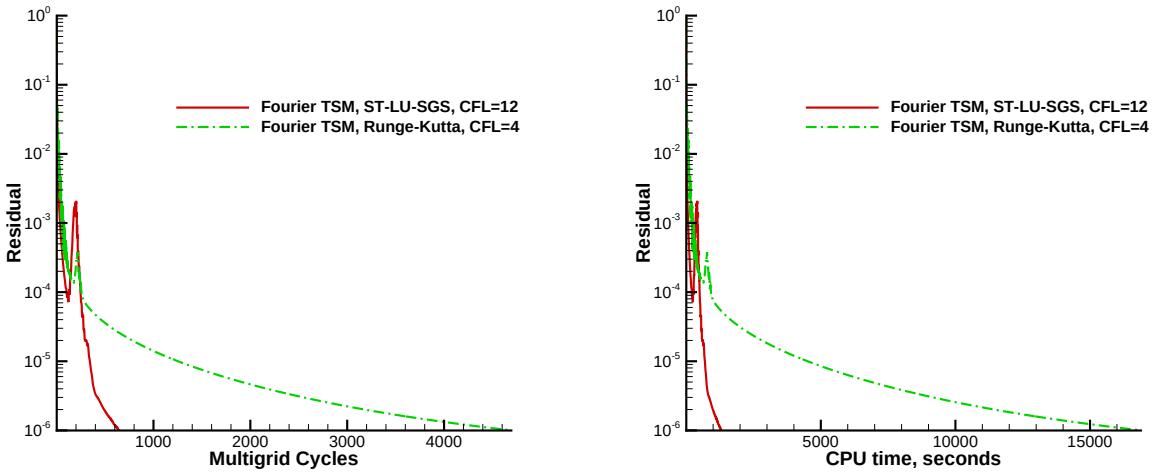
Figure 4.34: Pitching NACA0012 airfoil, periodic laminar flow, Mach number contours in the second half period.

The Mach number contours computed by the ST-LU-SGS scheme at the 8 real time levels are shown in Figure 4.33 for the first half period and Figure 4.34 for the second half period. It can be seen that under the given flow conditions, viscous effects are confined in a small region close to the airfoil surface as well as in the wake region. Close to the trailing edge, small vortices occur alternately on upper surface and lower surface as the airfoil pitches up and down. In the forced pitching movement, the instantaneous angle of attack achieves positive maximum at $t = \frac{2}{8}T$ and negative maximum at $t = \frac{6}{8}T$. At $t = 0$ and $t = \frac{4}{8}T$ zero angle of attack is arrived. It can be seen that the flow field is not symmetric even at these zero angle of attack instants. This is due to the unsteady effect. The flow field at current instant inherit features of that in the past and it imposes influence on the flow field in the future.

4.5.3 Laminar Vortex Shedding Flow Behind a Circular Cylinder

In the preceding test case, the high efficiency of the ST-LU-SGS scheme is verified for the laminar flow at a moderate Reynolds number. It is interesting to see if high efficiency using the scheme can be achieved for low Reynolds number flows. The laminar vortex shedding flow behind a circular cylinder is chosen to be the third test case. Though this flow is known to be periodic, the frequency is not a priori. Only a rough result can be estimated empirical formula using Strouhal number. To obtain the accurate frequency and the unsteady flow field in computation, a frequency searching process must be involved. As before, this process is carried out by firstly finding out the most significant Fourier mode of the lift coefficient and then using the gradient based method. Whether or not the ST-LU-SGS scheme works well with the above frequency searching process is also tested.

In this test case, a circular cylinder is fixed in the laminar flow at $M_\infty = 0.2$ and $Re = 180$. A 257×129 O-type mesh is used. The normal distance from the first grid point to the



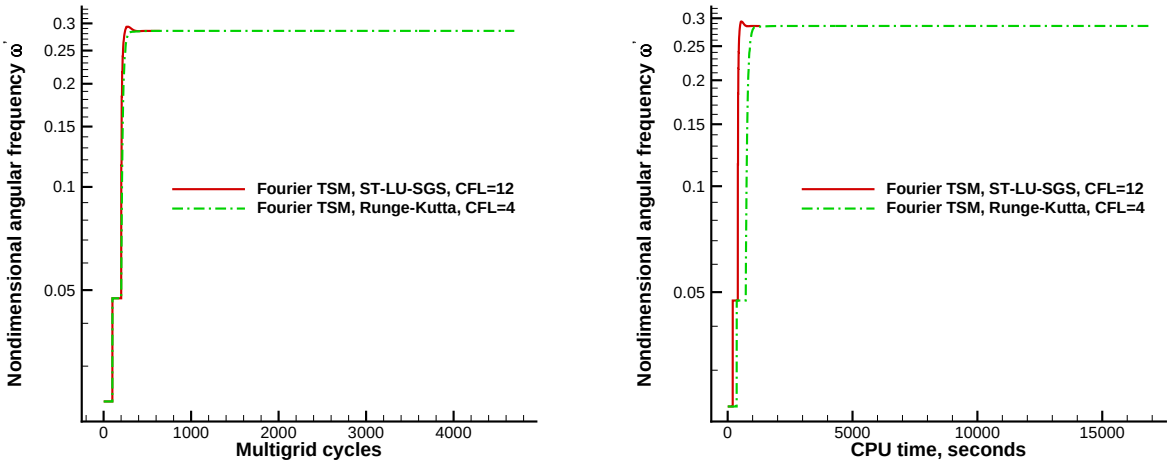
(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.35: Circular cylinder, laminar vortex shedding flow, convergence history.

surface is 1×10^{-3} of the cylinder diameter. The far field boundary of the mesh extends to about 200 diameters away from the cylinder. 8 real time intervals are still employed when implementing the time spectral method. Convergence criteria for the unsteady flow field is set to be 1×10^{-6}. The frequency search process does not stop until the flow field converges. Computational results using the ST-LU-SGS scheme are compared with those of explicit Runge-Kutta solver as well as experiments [35] [36] [37] [38].

The convergence histories are compared in Figure 4.35. The maximum CFL number that can be used in the ST-LU-SGS scheme is 12, which is lower than that in preceding laminar flow test cases. A possible reason for this is that during the frequency search process, the flow frequency change could be significant. This imposes difficulty in using large pseudo-time steps (hence large CFL number). From Figure 4.35(a), it can be seen that the explicit Runge-Kutta solver needs about 4600 multigrid cycles to converge, while only around 640 cycles are required for the ST-LU-SGS scheme. The ratio of the number of multigrid cycles is about 7.2. When CPU time is considered, that ratio increases to around 14. Though the maximum allowable CFL number can only reach 12, the ST-LU-SGS scheme is still much



(a) ω' versus Multigrid Cycles

(b) ω' vs. CPU time

Figure 4.36: Circular cylinder, laminar vortex shedding flow, ω' updating history.

more efficient than the explicit Runge-Kutta solver.

The frequency updating history with respect to Multigrid Cycles as well as CPU time is shown in Figure 4.36. The nondimensionalized frequency is defined as

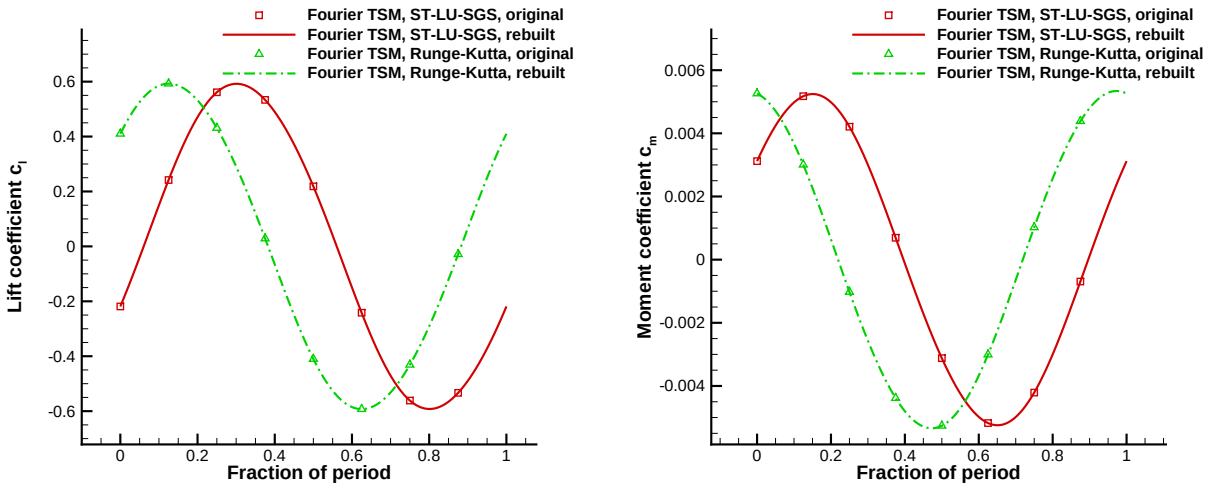
$$\omega' = \frac{\omega d}{\sqrt{\frac{p_\infty}{\rho_\infty}}} \tag{4.58}$$

where d is diameter of cylinder, p_∞ the free stream pressure and ρ_∞ the free stream density. It can be observed that the updated frequency approaches the accurate value faster if the explicit Runge-Kutta solver is replaced by the ST-LU-SGS scheme. This ensures high efficiency of the ST-LU-SGS scheme in the current test case. So obviously the ST-LU-SGS scheme also work very well with the frequency search method.

The time averaged drag coefficient and the Strouhal number of the present computations are compared with experimental data in Table 4.2. For the time averaged drag coefficient, results using both ST-LU-SGS scheme and Runge-Kutta solver match Henderson's data [38] very well. And for the Strouhal number, the present computational results are all very close

Table 4.2: Vortex shedding flow behind circular cylinder

	time averaged C_d	St
Williamson [35] [36]		0.1919
Roshko [37]		0.185
Henderson [38]	1.336	
full LU-SGS	1.336	0.1918
explicit Runge-Kutta	1.337	0.1919



(a) Lift coefficient

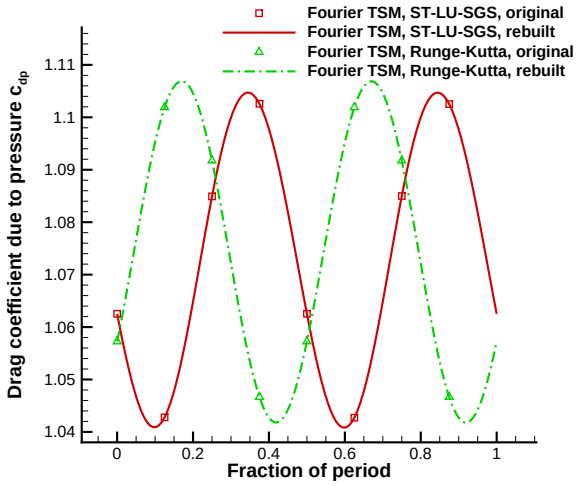
(b) Moment coefficient

Figure 4.37: Circular cylinder, laminar vortex shedding flow, lift and moment coefficients variation in a period

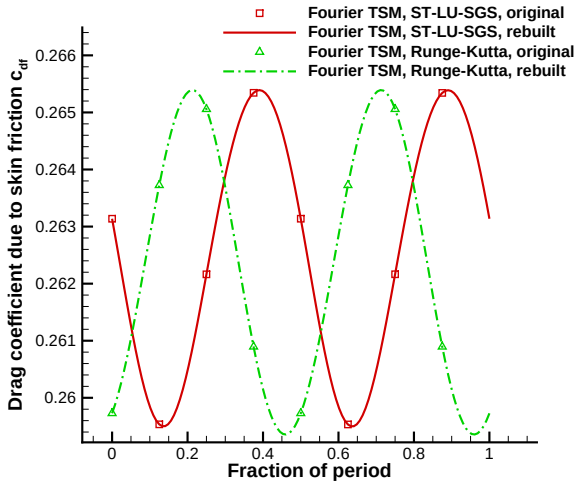
to Williamson's data [35] [36].

The variation of lift and moment coefficient in the finally searched period is plotted in Figure 4.37. For each force coefficient, the amplitude and the frequency are almost the same using the ST-LU-SGS scheme or explicit Runge-Kutta solver. However, in each figure, a phase difference can be seen. This is understandable since the phase is drifting during the frequency search process.

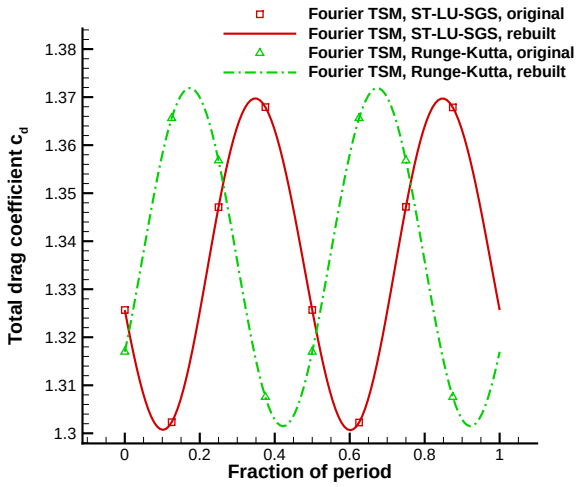
The temporal variation of total drag coefficient and its component due to pressure and skin friction in the finally searched period is shown in Figure 4.38. Similarly to Figure 4.37, there is phase difference between the results using ST-LU-SGS scheme and those of explicit



(a) Drag coefficient due to pressure

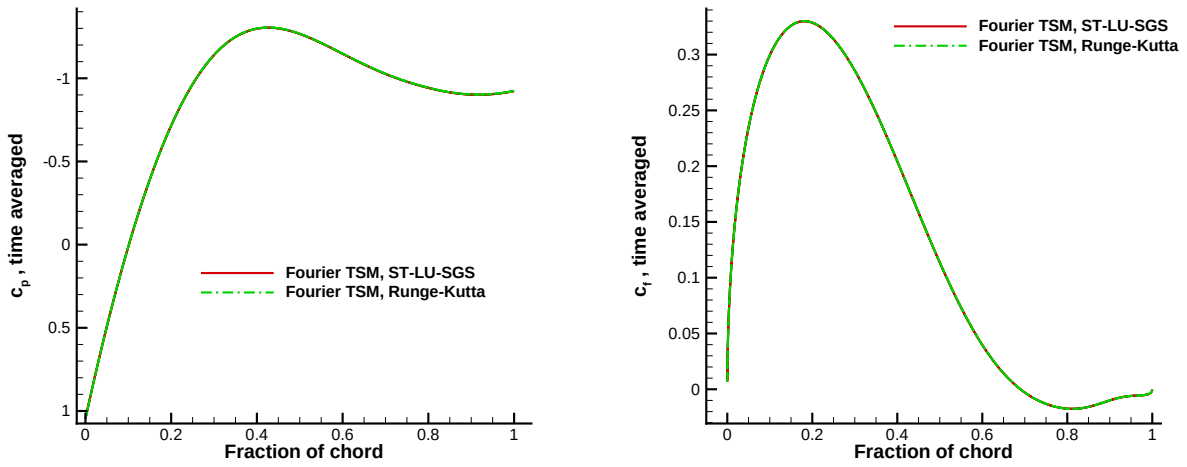


(b) Drag coefficient due to skin friction



(c) Total drag coefficient

Figure 4.38: Circular cylinder, laminar vortex shedding flow, drag coefficient variation in a period.



(a) Surface pressure coefficient

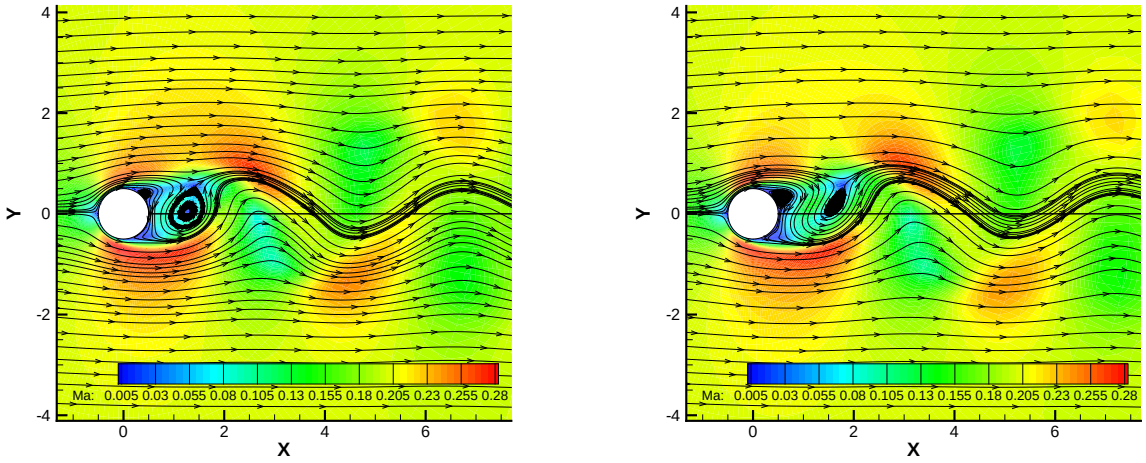
(b) Surface skin friction coefficient

Figure 4.39: Circular cylinder, laminar vortex shedding flow, surface force coefficient distribution, time averaged.

Runge-Kutta solver. In addition, there are slight differences in amplitude between the two group of results. This indicates the result of drag variation is more sensitive to the simulated vortex shedding than those of lift and moment coefficient.

The distribution of the time averaged surface force coefficient is shown in Figure 4.39. Results of using ST-LU-SGS scheme match those of explicit Runge-Kutta solver very well. In Figure 4.39(b), it can be seen that the time averaged skin friction coefficient is negative near the trailing edge. This is evidence that reverse flows exist on the surface in that area. Actually during the vortex shedding process, vortices are generated on the surface near trailing edge due to instability of the laminar boundary layer.

Figures 4.40 and 4.41 vividly show the vortex shedding process during the first and the second half period. Mach number contours in flood type and streamlines are computed using the ST-LU-SGS scheme. Every vortex in this flow is firstly generated on the surface near trailing edge, then it grows progressively bigger. When it is in about the size of the cylinder, it detaches. After that the detached vortex sheds away from the cylinder and



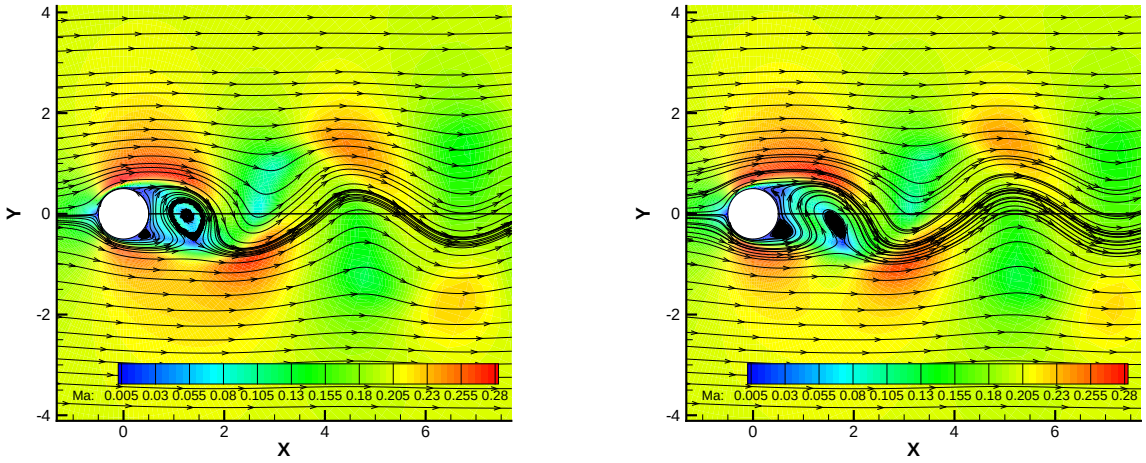
(a) $t = 0$

(b) $t = \frac{1}{8}T$

(c) $t = \frac{2}{8}T$

(d) $t = \frac{3}{8}T$

Figure 4.40: Circular cylinder, laminar vortex shedding flow, Mach number contours and steam lines in the first half period.



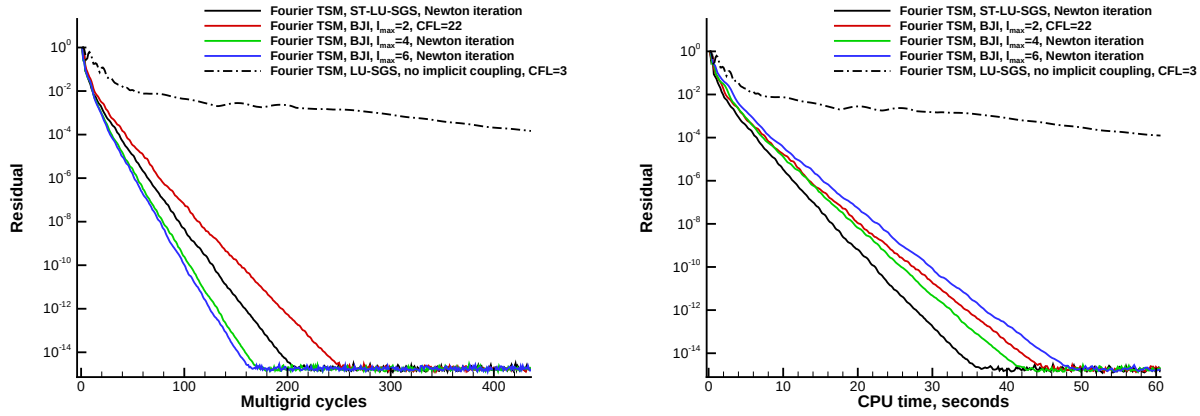
(a) $t = \frac{4}{8}T$

(b) $t = \frac{5}{8}T$

(c) $t = \frac{6}{8}T$

(d) $t = \frac{7}{8}T$

Figure 4.41: Circular cylinder, laminar vortex shedding flow, Mach number contours and steam lines in the second half period.



(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.42: Pitching NACA0012 airfoil, periodic inviscid flow, convergence history.

dissipates very fast in the wake. This process is seen to occur alternately on upper and lower surface near the trailing edge.

4.6 Test Cases for Comparing the ST-LU-SGS Scheme with the Block-Jacobi Implicit Algorithm

Since both schemes are developed for Fourier time spectral method and based on the original LU-SGS scheme for steady problems, it is interesting to compare their performance in solving the same periodic flow problem. The computational efficiency of using it is compared with that using present ST-LU-SGS scheme in several test cases. Results will be discussed.

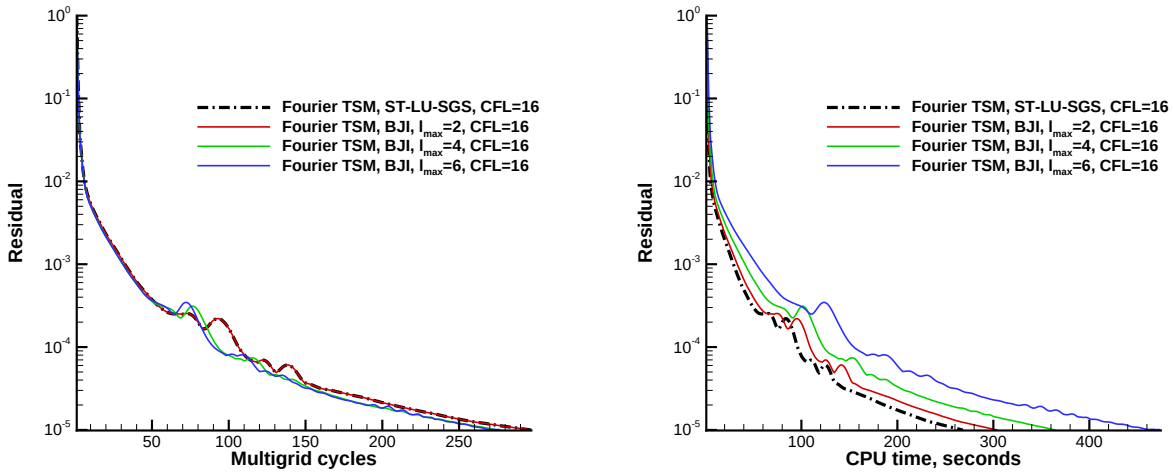
First of all, the inviscid flow over a pitching NACA0012 airfoil, which is the first test case in Section 4.5 is used for comparison.

The convergence history using the Block-Jacobi Implicit algorithm is compared with that of the ST-LU-SGS in Figure 4.42. In computations applying Block-Jacobi Implicit algorithm, different l_{max} is used. When $l_{max} = 2$, maximum allowable CFL number is 22. For $l_{max} > 2$,

Newton iteration is stable. As l_{max} increases, the computation converge faster and become more stable. This is understandable since as more Block-Jacobi steps are implemented, the implicit coupling term $\Omega D_n(\Delta\vec{W}^l)$ is updated more frequently. So the coupling between real time steps in the implicit algorithm is strengthened, which is helpful to make computation stable and speed up the convergence. But this improvement is achieved at the price of consuming more CPU time since more Block-Jacobi steps are involved. This fact can be observed in Figure 4.42(a). So finally for Block-Jacobi Implicit algorithm, if CPU time is concerned, $l_{max} = 4$ is the best choice. Similar results were reported by Frederic Sicot [24]. Compared to the Block-Jacobi Implicit algorithm, the convergence using the ST-LU-SGS scheme is not achieved at the minimum number of multigrid cycles. But when CPU time is considered, the ST-LU-SGS scheme reaches convergence the fastest. If the real time derivative approximated by Fourier time spectral operator is completely not considered in implicit operator, convergence of the computation becomes much slower. Besides the maximum allowable CFL number can only be used up to 3 and only three-level V-type multigrid can be used to make the computation converge. Any higher CFL number and more complex multigrid will make it unstable. This indicates the importance of including the time derivative term in implicit operator.

Next the Block-Jacobi Implicit algorithm is tested in viscous flow problems. The second test case in Section 4.5, which is the laminar periodic flow over an pitching NACA0012 airfoil, is employed again.

The convergence history versus multigrid cycles and CPU time is shown in Figure 4.43. In this test case, Newton iteration cannot be used for both algorithms. The maximum allowable CFL number decreases below 20. All computations including the one using ST-LU-SGS scheme converge at nearly the same number of multigrid cycles. When CPU time is considered, the ST-LU-SGS scheme is the least time consuming algorithm. However, the Block-Jacobi Implicit algorithm with $l_{max} = 2$ becomes almost equally efficient as the ST-



(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

Figure 4.43: Pitching NACA0012 airfoil, periodic laminar flow, convergence history.

LU-SGS scheme. This is because both of them use the same CFL number and converge at the same number of multigrid cycles in this test case. Besides, the CPU time consumed during each multigrid cycle is about the same. It seems in viscous flow problems similar to this test case, compared to the Block-Jacobi Implicit algorithm, the present ST-LU-SGS scheme has no advantages. However, computational experiment show that the present ST-LU-SGS scheme is more stable than the Block Jacobi Implicit algorithm for high frequency flow problems. For instance, if the airfoil pitching reduced frequency increases to 0.1, the Block-Jacobi Implicit algorithm cannot converge even at a low CFL number of 6, whereas the ST-LU-SGS scheme can still reach convergence at CFL number of 16 even without the high frequency modification. This fact can be observed in Figure 4.44.

A possible reason can be found from the difference of the two algorithms. In the Block-Jacobi Implicit algorithm, during each Block Jacobi step, the just updated solution change of current real time level is not used immediately on the next levels. But in the ST-LU-SGS scheme, since Gauss-Seidel sweeping is also carried out in time domain, the coupling of δw is in the strongest manner among different real time level. So the ST-LU-SGS scheme is more

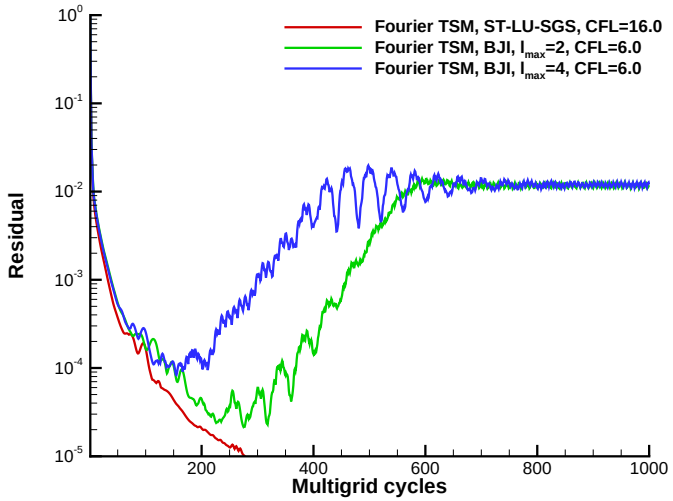


Figure 4.44: Pitching NACA0012 airfoil, periodic laminar flow, convergence history.

stable than the Block-Jacobi implicit algorithm.

4.7 The Space-Time LU-SGS Scheme for the Chebyshev Time Spectral Method and its Formulation

For fast periodic unsteady calculations, the newly proposed ST-LU-SGS scheme works well with the Fourier time spectral method. When the solution of a non-periodic problem is desired, Chebyshev time spectral method proposed in last chapter can be employed to save computational efforts. As mentioned before, the ST-LU-SGS scheme is not based on any feature of a specific time spectral operator, such as the Fourier time spectral operator. In principle, it can be applied to any time spectral method, including the Chebyshev time spectral method. The ST-LU-SGS scheme for the Chebyshev time spectral method can be derived in the same way as presented in section 4.4.

Replacing the Fourier time spectral operator with the Chebyshev time spectral operator,

equation (4.44) and (4.45) are modified as

the forward sweep

$$\begin{aligned}\delta w^*_{i,j,l} = \frac{\Delta t}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \Omega_{i,j}\sum_{n=0}^{l-1} C_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.59}$$

the backward sweep

$$\begin{aligned}\delta w^{(1)}_{i,j,l} = \frac{\Delta t}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \hat{\mathbf{A}}^-_{i+1,j,l}\Delta S_{i+\frac{1}{2},j,l}\delta w^{(1)}_{i+1,j,l} - \hat{\mathbf{A}}^-_{i,j+1,l}\Delta S_{i,j+\frac{1}{2},l}\delta w^{(1)}_{i,j+1,l} \\ &- \Omega_{i,j}\sum_{n=l+1}^{N} C_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.60}$$

where

$$\begin{aligned}D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l}n^I_{x,i,j,l} + \bar{v}_{i,j,l}n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l}n^J_{x,i,j,l} + \bar{v}_{i,j,l}n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Delta t\Omega C_{l,l} + \Delta t\frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^I_{i,j,l})^2 + (\Delta S^J_{i,j,l})^2}{\Omega_{i,j}}\end{aligned} \tag{4.61}$$

where $C_{l,n}$ denotes element of the Chebyshev time spectral operator. Modification due to viscous effect has been added if Navier-Stokes equations are solved.

when Δt goes to infinity, the above scheme reduces to the Newton iteration as

the forward sweep

$$\begin{aligned}\delta w^*_{i,j,l} = \frac{1}{D}[&\tilde{\mathbf{A}}^+_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^*_{i-1,j,l} + \tilde{\mathbf{A}}^+_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^*_{i,j-1,l} \\ &- \Omega_{i,j}\sum_{n=0}^{l-1} C_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.62}$$

the backward sweep

$$\begin{aligned}\delta w^{(1)}_{i,j,l} = \frac{1}{D}[&\tilde{\mathbf{A}}^{+}_{i-1,j,l}\Delta S_{i-\frac{1}{2},j,l}\delta w^{*}_{i-1,j,l} + \tilde{\mathbf{A}}^{+}_{i,j-1,l}\Delta S_{i,j-\frac{1}{2},l}\delta w^{*}_{i,j-1,l} \\ &- \hat{\mathbf{A}}^{-}_{i+1,j,l}\Delta S_{i+\frac{1}{2},j,l}\delta w^{(1)}_{i+1,j,l} - \hat{\mathbf{A}}^{-}_{i,j+1,l}\Delta S_{i,j+\frac{1}{2},l}\delta w^{(1)}_{i,j+1,l} \\ &- \Omega_{i,j}\sum_{n=l+1}^{N} C_{l,n}\delta w_{i,j,n} - \tilde{R}_{i,j,l}]\end{aligned} \tag{4.63}$$

where

$$\begin{aligned}D =&(\left|\bar{u}_{i,j,l}n^{I}_{x,i,j,l} + \bar{v}_{i,j,l}n^{I}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{I}_{i,j,l} \\ &+ (\left|\bar{u}_{i,j,l}n^{J}_{x,i,j,l} + \bar{v}_{i,j,l}n^{J}_{y,i,j,l}\right| + c_{i,j,l})\Delta S^{J}_{i,j,l} \\ &+ \Omega C_{l,l} + \frac{\gamma\mu_{i,j,l}}{\rho_{i,j,l}Pr}\frac{(\Delta S^{I}_{i,j,l})^2 + (\Delta S^{J}_{i,j,l})^2}{\Omega_{i,j}}\end{aligned} \tag{4.64}$$

Numerical experiments show that if the time interval of interest (a, b) is sufficiently short, the implicit operator of the ST-LU-SGS scheme for the Chebyshev time spectral method may have the convergence problem. Inspired from the analysis of the same problem for Fourier time spectral method, it is helpful to look into the features of the Chebyshev time spectral operator. If (a, b) is divided into 4 intervals, the Chebyshev time spectral operator can be calculated and written in matrix form as

$$C = \frac{2}{b-a}\begin{pmatrix} -5.5 & 6.828 & -2 & 1.172 & -0.5 \\ -1.707 & 0.707 & 1.414 & -0.707 & 0.293 \\ 0.5 & -1.414 & 0 & 1.414 & -0.5 \\ -0.293 & 0.707 & -1.414 & -0.707 & 1.707 \\ 0.5 & -1.172 & 2 & -6.828 & 5.5 \end{pmatrix} \tag{4.65}$$

Unlike the Fourier time spectral operator, the diagonal elements are not all zero in the Chebyshev time spectral operator. However, the matrix is not diagonally dominant. This feature is similar to that of the Fourier time spectral operator. Besides, it can be observed

that the elements of the above matrix are all inversely proportional to the time length $b-a$. In the ST-LU-SGS scheme for Chebyshev time spectral method, at any time level l, the diagonal element of the time spectral operator $C_{l,l}$ only appears in the diagonal elements of the implicit operator. The rest elements in the row l of the Chebyshev time spectral operator $C_{l,n}(n \neq l)$ can be found only in off-diagonal elements of the implicit operator. So the contribution from the time spectral operator is not desirable to make the implicit operator diagonally dominant, while the treatments in space make favourable contribution. Whether or not the implicit operator is diagonally dominant depends on the dominant contribution. If the time length $b-a$ is not too small, the spectral radius of the convective Jacobian matrices are big enough to maintain the diagonal dominance of the implicit operator. It can be imagine that when the time length $b-a$ is smaller than a certain value, the implicit operator loses diagonal dominance. Then the convergence of the ST-LU-SGS for Chebyshev time spectral method is not guaranteed.

To solve this problem, a similar modification made to the ST-LU-SGS for the Fourier time spectral method can still be employed. The $C_{l,l}$ in the diagonal element of the implicit operator can be replaced by the following term

$$C_{l,max} = max(C(l,n)), 0 \leq n \leq N \tag{4.66}$$

It can be shown the modified implicit operator is still diagonally dominant even for very small time length $b-a$. So the convergence of the ST-LU-SGS scheme for Chebyshev time spectral method for very small time length $b-a$ could be improved. And at the same time the accuracy of the flow solution is not influenced since the it completely depends on the explicit residual other than the implicit operator. This treatment is called small time interval modification in this dissertation. With this treatment, only equation (4.61) need to

be revised as

$$\begin{aligned} D = \Omega_{i,j} &+ \Delta t(\left|\bar{u}_{i,j,l} n^I_{x,i,j,l} + \bar{v}_{i,j,l} n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ \Delta t(\left|\bar{u}_{i,j,l} n^J_{x,i,j,l} + \bar{v}_{i,j,l} n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Delta t \Omega C_{l,max} + \Delta t \frac{\gamma \mu_{i,j,l}}{\rho_{i,j,l} Pr} \frac{(\Delta S^I_{i,j,l})^2 + (\Delta S^J_{i,j,l})^2}{\Omega_{i,j}} \end{aligned} \tag{4.67}$$

in the original ST-LU-SGS scheme of the general form and only equation (4.64) need to be revised as

$$\begin{aligned} D = &(\left|\bar{u}_{i,j,l} n^I_{x,i,j,l} + \bar{v}_{i,j,l} n^I_{y,i,j,l}\right| + c_{i,j,l})\Delta S^I_{i,j,l} \\ &+ (\left|\bar{u}_{i,j,l} n^J_{x,i,j,l} + \bar{v}_{i,j,l} n^J_{y,i,j,l}\right| + c_{i,j,l})\Delta S^J_{i,j,l} \\ &+ \Omega C_{l,max} + \frac{\gamma \mu_{i,j,l}}{\rho_{i,j,l} Pr} \frac{(\Delta S^I_{i,j,l})^2 + (\Delta S^J_{i,j,l})^2}{\Omega_{i,j}} \end{aligned} \tag{4.68}$$

for the Newton iteration form if infinitely large pseudo-time step is allowed. This treatment will be tested with the flow test cases in next section.

4.8 The Non-periodic Flow Test Cases Using the ST-LU-SGS Scheme for the Chebyshev Time Spectral Method

As can be seen in section 3.4.2, non-symmetric solution could be produced using Chebyshev time spectral method for a symmetric periodic problem. For this reason, only non-periodic flow test cases are used in this section to validate the ST-LU-SGS scheme for the Chebyshev time spectral method. For the sake of convenience, the non-periodic test cases discussed in this section are still constructed for the inviscid and laminar flow over a pitching NACA0012 airfoil. In either test case, the exact initial flow field is obtained from the periodic solution using Fourier time spectral method.

4.8.1 Inviscid Flow over a Pitching NACA0012 Airfoil

The same non-periodic inviscid flow problem presented in section 3.4.3 is still used as the inviscid test case here. In this test case, a NACA0012 airfoil is forced to pitch around its quarter chord at $M_\infty = 0.755$. The mean angle of attack, pitching range and reduced frequency are given by $\alpha_0 = 0.0°$, $\alpha_m = 2.51°$ and $\kappa = 0.0814$, respectively.

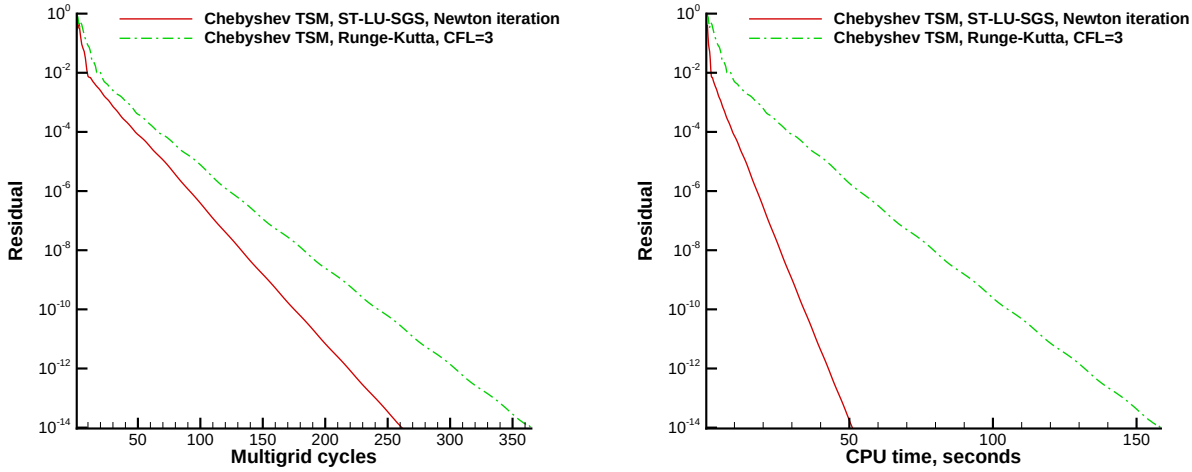
To construct the non-periodic inviscid test case, the pitching motion of the airfoil is considered only in the first quarter period. The computed flow field at the zero time level in the periodic solution using Fourier time spectral method with 32 time levels is assigned to be the exact initial flow field.

The same 128×32 C-type grid as in the previous steady inviscid flow problem is used in this test case again. To implement the Chebyshev time spectral method, the time interval of interest(the first quarter period) is divided into eight segments according to Gauss-Lobatto quadrature rule. Computational results using the ST-LU-SGS scheme are compared with those of the explicit Runge-Kutta solver. The convergence criteria is set to be 1×10^{-14}.

Numerical experiments show that the length of the time interval of interest in this test case, which is a quarter period, is small enough to make the original ST-LU-SGS scheme for the Chebyshev time spectral method unstable. However, computations can converge once the small time interval modification proposed in last section is added.

Computational results show that the ST-LU-SGS scheme can be carried out in the way of Newton iteration method. Which is equivalent to use infinitely large CFL number(hence infinitely large pseudo-time step) in the scheme for this inviscid case. This indicates the stability of the ST-LU-SGS scheme for the Chebyshev time spectral method is well maintained when the time intervals of interest is small.

The residual history with respect to multigrid cycles and CPU time is shown in Figure 4.45(a).



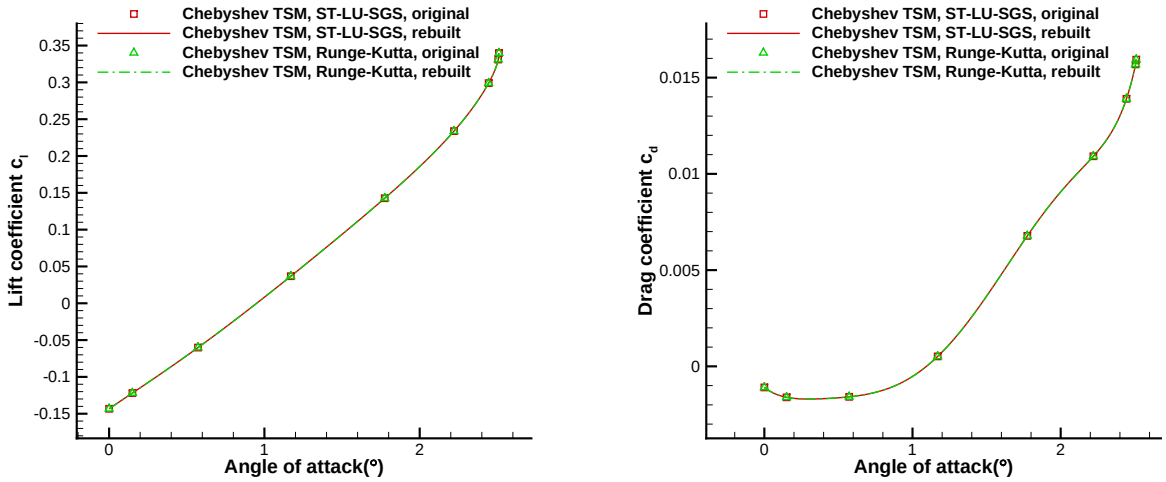
(a) Residual versus Multigrid Cycles

(b) Residual versus CPU time

Figure 4.45: Pitching NACA0012 airfoil, non-periodic inviscid flow, convergence history.

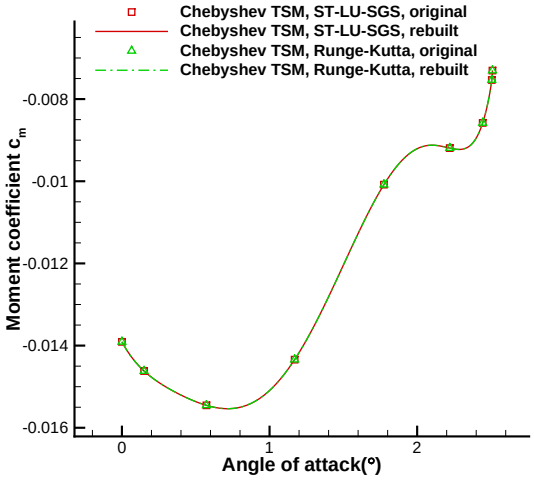
The number of multigrid cycles for the explicit Runge-Kutta solver to reach convergence is about 1.38 times of that for the ST-LU-SGS scheme. When CPU time is compared, it can be observed in Figure 4.45(b) this ratio goes up to 3.2. So more than a half of the CPU time spent in a multigrid cycle can be saved if the explicit Runge-Kutta solver is replaced by the ST-LU-SGS scheme. This result is similar to that of the ST-LU-SGS scheme for Fourier time spectral method.

The force coefficient variation with respect to angle of attack is shown in Figure 4.46. The symbols denote the original results on the 9 real time levels, while the lines represent the rebuilt results by the resolved Chebyshev polynomials. The rebuilt results are plotted on 129 equally spaced real time levels. It can be seen computations using the ST-LU-SGS scheme and explicit Runge-Kutta both converge to the same solution. So the application of ST-LU-SGS scheme does not change solution. It only improves computational efficiency. Results of this test case verifies the above facts as expected.



(a) Lift coefficient

(b) Drag coefficient



(c) Moment coefficient

Figure 4.46: Pitching NACA0012 airfoil, non-periodic inviscid flow, force and moment coefficients versus angle of attack.

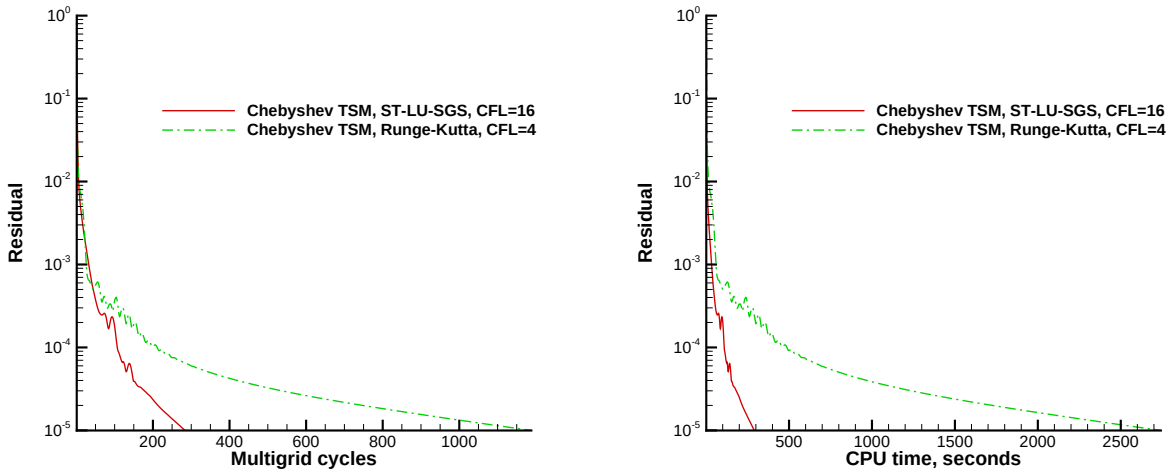
4.8.2 Laminar Flow over a Pitching NACA0012 Airfoil

Next the ST-LU-SGS scheme for the Chebyshev time spectral method is further tested with a laminar flow over a pitching NACA0012 airfoil. The same constructed non-periodic test case in section 3.4.4 is used again. In this test case, the airfoil still pitches around its quarter chord. Flow conditions are still given as $M_\infty = 0.5$ and $Re = 5000$. For the pitching motion, $\alpha_0 = 0.0°$, $\alpha_m = 1.51°$ and $\kappa = 0.05$.

To construct the non-periodic test case, the airfoil pitching motion is considered only in time intervals $(0, 0.53T)$, where T is the period of the pitching motion. The exact initial flow field is produced in the same way as that in the inviscid test case. However, only 16 intervals are used in the Fourier time spectral method to obtain the periodic solution. This is because this test case is shock-free and 16 intervals in a period is sufficient to achieve accurate solution.

The same 257×65 C-type grid in the previous steady laminar flow example is used. When implementing the time spectral method, the time intervals of interest is split into 8 smaller intervals according to Gauss-Lobatto quadrature rule. Computational results are still compared between the ST-LU-SGS solver and the Runge-Kutta solver. It is found only finite large CFL numbers, such as 16, are allowed in the ST-LU-SGS scheme for the Chebyshev time spectral method when laminar flows are calculated. This fact can also be observed when the ST-LU-SGS scheme for the Fourier time spectral method is used. The convergence criteria is set to be 1×10^{-5}.

The convergence history vesus multigrid cycles is shown in Figure 4.47(a). It can be seen to reach convergence, the explicit Runge-Kutta solver uses almost 4.3 times the multigrid cycles needed by the ST-LU-SGS scheme. And Figure 4.47(b) shows this ratio is enlarged to over 9 when CPU time is compared. Just like the applying the ST-LU-SGS scheme for the Fourier time spectral method to periodic laminar flows, the acceleration effect of considering Chebyshev time spectral operator in the implicit operator can be accumulated on that of



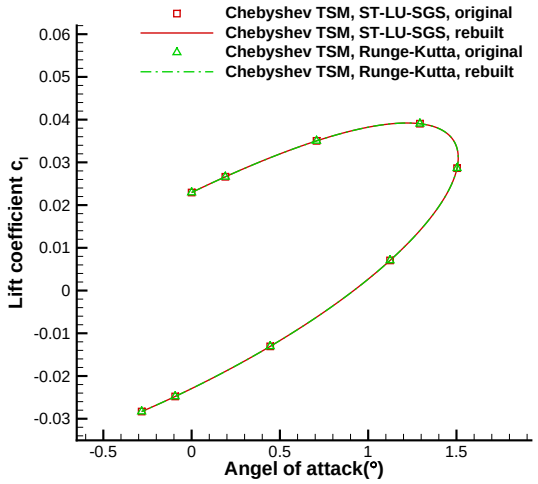
(a) Residual versus Multigrid cycles

(b) Residual versus CPU time

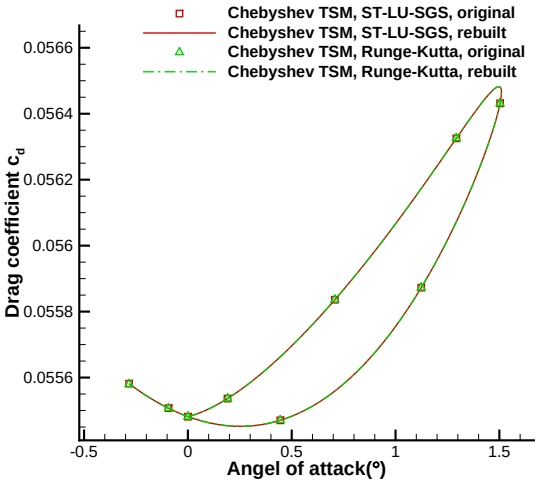
Figure 4.47: Pitching NACA0012 airfoil, non-periodic laminar flow, convergence history.

adding modification due to viscous fluxes in the implicit operator. This result confirms that the performance of the ST-LU-SGS scheme in laminar flow cases is better than that in inviscid flow cases. More computational effort can be saved in laminar flow cases.

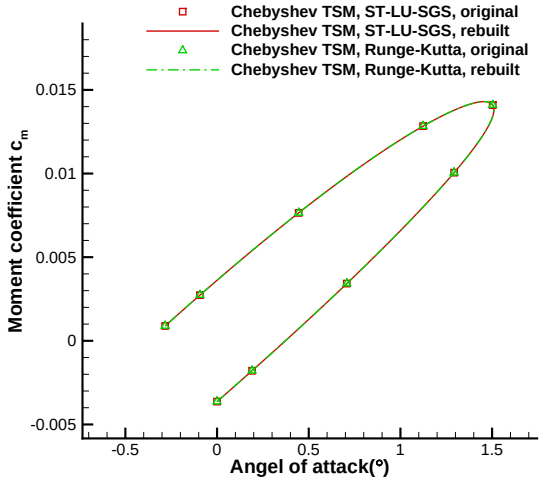
The variation of the force coefficients with respect to angle of attack is shown in Figure 4.48. As before, symbols denote original results on the 9 real time levels, while the lines represent the rebuilt results by the resolved Chebyshev polynomial. The rebuilt results are plotted on 129 equally spaced real time levels. It can be observed that computational results using the ST-LU-SGS scheme and those of explicit Runge-Kutta solver completely overlap each other as expected.



(a) Lift coefficient



(b) Drag coefficient



(c) Moment coefficient

Figure 4.48: Pitching NACA0012 airfoil, non-periodic laminar flow, force and moment coefficients versus angle of attack.

Chapter 5

Conclusions and Future Work

In this dissertation, firstly, the Fourier time spectral method is studied and validated with test cases. The non-symmetric solutions caused by odd number of intervals in a period for symmetric periodic flow problems are discussed in detail. The requirement of ensuring symmetric solution is proposed. Then the Fourier time spectral method is applied to periodic flow problems where the period is not known a priori. To search the frequency during unsteady computation, a new approach based on Fourier analysis of the lift coefficient is proposed to work with the Fourier time spectral method. A combined method in which the new approach is followed by the gradient based method is validated with a vortex shedding flow problem. Computational results show that the correct frequency can be found out, even when the initial guesses of the frequency are relatively far away from the exact value. Since Fourier time spectral method is theoretically only capable of solving periodic unsteady problems, a Chebyshev time spectral method is proposed to solve non-periodic unsteady problems. The method has been validated by solving one dimensional viscous Burgers' equation and two dimensional periodic and non-periodic flow problems. Computational results show that this method provides higher accuracy in approximating the time derivative term than any algebraic order finite-difference method. So it is very efficient in simulating

unsteady flows, especially non-periodic problems. A drawback of using Chebyshev time spectral method to solve periodic problems is pointed out and studied. The method produces non-symmetric solutions to symmetric problems even though it uses even number of intervals. The newly proposed requirement of ensuring symmetric solution is found still true for the present Chebyshev time spectral method.

The solver that has been used extensively in the past is the explicit Runge-Kutta solver. In a lot of occasions only small pseudo-time steps are allowed due to stability or accuracy requirement for such solvers. When it is applied to time spectral method, theoretical analysis points out the problem becomes even worse. The high computational efficiency expected to be achieved by time spectral method is usually jeopardized by the explicit solver. To overcome this problem, a new Space-Time LU-SGS (ST-LU-SGS) implicit scheme is proposed for both Fourier and Chebyshev time spectral methods. In that scheme, time domain is regarded as one additional dimension in space. Computational experiments show that using this new scheme for time spectral methods can make periodic and non-periodic computations converge faster and save CPU time in each multigrid cycle compared to the explicit Runge-Kutta solver. Hence the total consumed CPU time for an unsteady computation is greatly saved. For the present laminar flow test cases, computations using the ST-LU-SGS implicit scheme can converge over ten times faster than those employing the explicit Runge-Kutta solver. The ST-LU-SGS implicit scheme also works very well with the proposed frequency search approach. The ST-LU-SGS scheme for the Fourier time spectral method is compared with the Block-Jacobi implicit algorithm by[24]. Computational results show that convergence of using the ST-LU-SGS scheme is not slower than that using Block-Jacobi implicit algorithm. Besides, unlike the Block-Jacobi implicit algorithm, the ST-LU-SGS scheme uses updated information as soon as it becomes available. So the ST-LU-SGS implicit scheme is found more stable than the Block-Jacobi implicit algorithm, especially for high frequency flow problems. Modifications to the ST-LU-SGS scheme is proposed to ensure fast convergence for periodic problems of high frequency and non-periodic problems with small time

intervals of interest. These modifications have been verified in the test cases.

In the future, to further improve the efficiency of unsteady computations, a Space-Time multigrid (ST-multigrid) method for time spectral method would be proposed and studied. It regards time domain as one additional dimension in space during each multigrid cycle. It has the potential to further save computational cost of using time spectral method. The next step is to apply ST-LU-SGS scheme and hopefully the ST-multigrid method along with Fourier and Chebyshev time spectral method to practical three-dimensional unsteady turbulent flows.

Bibliography

[1] Magnus, R. and Yoshihara, H., "Inviscid Transonic Flow over Airfoils", *AIAA Journal*,Vol. 8, Dec. 1970, pp. 2157-2162.

[2] MacCormack, R.W., "The Effect of Viscosity in Hypervelocity Impact Cratering", *AIAA Paper* 69-0354, 1969.

[3] Jameson, A., Schmidt, W. and Turkel,E., "Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes", *AIAA Paper* 81-1259, 1981.

[4] Jameson, A. and Baker, T.J.,"Mutltigrid Solution of the Euler Equations for Aircraft Configurations", *AIAA Paper* 84-0093, 1984.

[5] Beam, R.M. and Warming, R.F., "Inviscid Transonic Flow over Airfoils", *AIAA Journal*,Vol. 16, No. 4 Apr. 1978, pp. 393-402.

[6] Jameson, A. and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes", *AIAA Journal*,Vol. 24, No. 11 Nov. 1986, pp. 1737-1743.

[7] Jameson, A. and Turkel, E., "Implicit Schemes and LU Decompositions", *Mathematics of Computation*,Vol. 37, No. 156 Nov. 1981, pp. 385-397.

[8] Yoon, S. and Jameson, A., "A Multigrid LU-SSOR Scheme for Approximate Newton Iteration Applied to the Euler Equations", *NASA CR 179524*, Sep. 1986.

[9] Yoon, S. and Jameson, A., "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations", *AIAA Journal*,Vol. 26, No. 9 Nov. 1988, pp. 1025-1026.

[10] Jameson, A. and Yoon, S., "LU Implicit Schemes with Multiple Grids for the Euler Equations", *AIAA Paper* 86-0105, 1986.

[11] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings", *AIAA Paper* 91-1596, 1991.

[12] Hall, K. C. and Crawley, E. F., "Calculation of Unsteady Flows in Turbomachinery Using the Linearized Euler Equations", *AIAA Journal*, Vol. 27, No. 6 1989, pp. 777-787.

[13] Ning, W. and He, L., "Computation of Unsteady Flows around Oscillating Blades Using Linear and Nonlinear Harmonic Euler Methods", *ASME Journal of Turbomachinery*, Vol. 120 July 1998. pp. 508-514.

[14] Hall, K. C., Thomas, J. P. and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique", *AIAA Journal*, Vol. 40, No. 5 May 2002.

[15] McMullen, M., Jameson, A. and Alonso, J.J., "Application of a Non-Linear Frequency Domain Solver to the Euler and Navier-Stokes Equations", *AIAA Paper* 2002-0120, 2002.

[16] McMullen, M. and Jameson, A., "The Computational Efficiency of Non-Linear Frequency Domain Methods", *Journal of Computational Physics*, Vol. 212 2006. pp. 637-661.

[17] Gopinath, A. K., Weide, E., Alonso, J. J., Jameson, A., Ekici, K. and Hall, K. C., "Turbomachinery Simulations Using the Harmonic Balance Technique", *AIAA Paper*, Jan. 2007.

[18] Ekici, K., Hall, K. C. and Dowell, E. H., "Computationally Fast Harmonic Balance Methods for Unsteady Aerodynamic Predictions of Helicopter Rotors", *AIAA Paper*, Jan. 2008.

[19] Gopinath, A. K. and Jameson, A., "Application of the Time Spectral Method to Periodic Unsteady Vortex Shedding", *AIAA Paper*, Jan. 2006.

[20] Ekici, K. and Hall, K. C., "Harmonic Balance Analysis of Limit Cycle Oscillations in Turbomachinery", *AIAA Journal*, Vol. 49, No. 7 July 2011.

[21] Spiker, M. A., "Development of an Efficient Desigh Method for Non-Synchronous Vibrations", PhD thesis, Duke University, 2008.

[22] Ekici, K. and Hall, K. C., "Nonlinear Frequency-Domain Analysis of Unsteady Flows in Turbomachinery with Multiple Excitation Frequencies", *AIAA Paper*, Jun. 2006.

[23] Kopriva, D.A. "Implementing Spectral Methods for Partial Differential Equations", Springer.

[24] Sicot, F., Puigt, G. and Montagnac, M., "Block-Jacobi Implicit Algorithms for the Time Spectral Method", *AIAA Journal*, Vol. 46, No. 12 Dec. 2008, pp. 3080-3089.

[25] Woodgate, M.A., Badcock, K.J., "Implicit Harmonic Balance Solver for Transonic Flow with Forced Motions", *AIAA Journal*, Vol. 47, No. 4 Apr. 2009, pp. 893-901.

[26] Su, X. and Yuan, X., "Implicit Solution of Time Spectral Method for Periodic Unsteady Flows", *International Journal for Numerical Methods in Fluids*,Vol. 63, 2010, pp. 860-876.

[27] Thomas, J.P., Custer, C.H., Dowell, E.H. and Hall, K.C. "Compact Implementation Strategy for a Harmonic Balance Method within Implicit Flow Solvers", *AIAA Journal*, Vol. 51, No. 6 Jun. 2013, pp. 1374-1381.

[28] MacCormack, R. and Baldwin, B.,"A Numerical Method for Solving the Navier-Stokes Equations With Application to Shock-Boundary Layer Interactions", *AIAA Paper* 1975-0001, 1975.

[29] Beam, R.M. and Warming, R.F. "An Implicit Finite-Defference Algorithm for Hyperbolic Systems in Consevation-Law Form-Application to Eulerian Gasdynamic Equations", *Journal of Computational Physics*, Vol. 22, 1976, pp. 87-110.

[30] Landon, R. H., "Naca0012 Oscillation and Transient Pitching, Compendium of Unsteady Aerodynamic Measurements, Data Set 3", *AGARD TR R-702*, Jan. 1982.

[31] Antheaume, S., Corre, C., "Implicit Time Spectral Method for Periodic Incompressible Flows", *AIAA Journal*, Vol. 49, No. 4 Apr. 2011, pp. 791-805.

[32] Im, D.K. and Kwon, J.H.,"Periodic Unsteady Flow Analysis using a Diagonally Harmonic Balance Method", *AIAA Paper* 2011-3703, 2011.

[33] Gao, C., Yang, S., Luo, S. and Liu, F., "Calculation of Airfoil Flutter by and Euler Method with Approximate Boundary Condition", *AIAA Journal*, Vol. 43, No. 2 Feb. 2005, pp. 295-305.

[34] White, F.M. "Viscous Fluid Flow" Second Edition, McGraw Hill.

[35] Williamson, C. H. K., "Defining a Universal and Contiguous Strouhal-Reynolds Number Relationship of the Laminar Vortex Shedding of a Circular Cylinder", *Physics of Fluids*, Technical Report, Vol. 31 Oct 1988, pp. 2747-2747.

[36] Williamson, C. H. K., "Measurements of Base Pressure in the Wake of a Cylinder at Low Renolds Numbers", Technical report 14, 1990, pp. 38-46.

[37] Roshko, A., "On the Development of Turbulent Wakes from Vortex Streets", *NACA TR 1191*, Jan. 1954.

[38] Henderson, R.D., "Details of the Drag Curve near the Onset of Vortex Shedding", *Physics of Fluids TR*, 1955.

[39] Blazed, J. "Computational Fluid Dynamics: Principles and Applications",Second Edition Elsevier.

[40] Tysinger T.L. and Caughey, D.A.,"Implicit Multigrid Algorithm for the Navier-Stokes Equations", *AIAA Paper* 91-0242, 1991.

[41] Tysinger T.L. and Caughey, D.A., "Alternating Direction Implicit Methods for the Navier-Stokes Equations", *AIAA Journal*,Vol. 30, No. 8, Technical Notes, 1992, pp. 2158-2161.

[42] Tritton, D. J., "Experiments on the Flow past a Circular Cylinder at Low Reynolds Numbers", Journal of Fluid Mechanics, Vol. 6, 1959, pp. 547-567.

[43] Coutanceau, M., and Bouard, R., "Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 1. Steady Flow", Journal of Fluid Mechanics, Vol. 79, 1977, pp. 231-256.

[44] Dennis, S., and Chang, G., "Numerical Simulations for Steady Flow past a Circular Cylinder at Reynolds Number Up to 100", Journal of Fluid Mechanics, Vol. 42, 1970, pp. 471-489.

[45] Fornberg, B., "A Numerical Study of Steady Viscous Flow past a Circular Cylinder", Journal of Fluid Mechanics, Vol. 98, 1980, pp. 819-855.

[46] Gautier, R., Biau, D., and Lamballais, E., "A Reference Solution of the Flow over a Circular Cylinder at Re = 40", Computers & Fluids, Vol. 75, 2013, pp. 103-111.