UNIVERSITY OF MIAMI

# PARALLEL COMPUTATION OF FLUID-STRUCTURAL INTERACTIONS USING HIGH RESOLUTION UPWIND SCHEMES

By

Zongjun Hu

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

May 2005

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the reuirements for the degree of
Doctor of Philosophy

# PARALLEL COMPUTATION OF FLUID-STRUCTURAL INTERACTIONS USING HIGH RESOLUTION UPWIND SCHEMES

Zongjun Hu

Approved:

Dr. Gecheng Zha
Associate Professor of
Mechanical & Aerospace Engineering

Dr. Steven G. Ullmann
Dean of the Graduate School

Dr. Hongtan Liu
Associate Professor of
Mechanical & Aerospace Engineering

Dr. Xudong(Don) Yang
Assistant Professor of Civil, Arch.
and Environmental Engineering

Dr. Singiresu S. Rao
Professor and Chairman of
Mechanical & Aerospace Engineering

HU, ZONGJUN (Ph.D., Mechanical Engineering)

Parallel computation of fluid-structural interactions using high resolution upwind schemes (May 2005)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Gecheng Zha.
No. of pages in text. (220)

An efficient and accurate solver is developed to simulate the non-linear fluid-structural interactions in turbomachinery flutter flows. A new low diffusion E-CUSP scheme, Zha CUSP scheme, is developed to improve the efficiency and accuracy of the inviscid flux computation. The 3D unsteady Navier-Stokes equations with the Baldwin-Lomax turbulence model are solved using the finite volume method with the dual-time stepping scheme. The linearized equations are solved with Gauss-Seidel line iterations. The parallel computation is implemented using MPI protocol.

The solver is validated with 2D cases for its turbulence modeling, parallel computation and unsteady calculation. The Zha CUSP scheme is validated with 2D cases, including a supersonic flat plate boundary layer, a transonic converging-diverging nozzle and a transonic inlet diffuser. The Zha CUSP2 scheme is tested with 3D cases, including a circular-to-rectangular nozzle, a subsonic compressor cascade and a transonic channel. The Zha CUSP schemes are proved to be accurate, robust and efficient in these tests.

The steady and unsteady separation flows in a 3D stationary cascade under high incidence and three inlet Mach numbers are calculated to study the steady state separation flow patterns and their unsteady oscillation characteristics. The leading edge vortex shedding is the mechanism behind the unsteady characteristics of the high incidence separated flows. The separation flow characteristics is affected by the inlet Mach number.

The blade aeroelasticity of a linear cascade with forced oscillating blades is studied using parallel computation. A simplified two-passage cascade with periodic boundary condition is first calculated under a medium frequency and a low incidence. The full scale cascade with 9 blades and two end walls is then studied more extensively under three oscillation frequencies and two incidence angles. The end wall influence and the blade stability are studied and compared under different frequencies and incidence angles.

The Zha CUSP schemes are the first time to be applied in moving grid systems and 2D and 3D calculations. The implicit Gauss-Seidel iteration with dual time stepping is the first time to be used for moving grid systems. The NASA flutter cascade is the first time to be calculated in full scale.

HU, ZONGJUN (Ph.D., Mechanical Engineering)
Parallel computation of fluid-structural interactions (May 2005)
using high resolution upwind schemes

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Gecheng Zha.
No. of pages in text. (220)

The objective of this dissertation is to develop an efficient and accurate solver to simulate the non-linear fluid-structural interaction in turbomachinery flutter flow. The fully implicit three-dimensional time accurate solver is developed to solve the non-linear Favre-averaged Navier-Stokes equations with Baldwin-Lomax turbulence model. The governing equations are discretized with the finite volume method. The inviscid flux on the control volume inter surface is calculated with several upwind schemes. A new low diffusion E-CUSP scheme is developed to improve the efficiency and accuracy of inviscid flux computation. The inviscid flux differencing achieves the third order accuracy by the MUSCL differencing approach. The viscous flux achieves the second order with central differencing. A second order of time marching is achieved by using dual-time stepping scheme. The Gauss-Seidel line iteration is applied on the inner pseudo time steps to implicitly solve the linearized equations. The convergence is accelerated using the technique of local time stepping. The parallel computation is implemented using MPI protocol.

The solver is first validated with several 2D cases. The Baldwin-Lomax turbulence model is validated with a subsonic flat plate boundary layer flow and a transonic inlet diffuser flow with the interaction between shock wave and boundary layer. The computed velocity profile in the flat plate boundary layer agrees very well with the law of the wall.

The shock wave in the inlet diffuser is clearly captured and the pressure distribution is compared well with experiment measurement. The flat plate boundary layer flow is further used to test the parallel computation capability. A good wall clock time speedup scalability is achieved when the job is computed with up to 10 processors. The solver's unsteady calculation capability is tested with the self-excited shock wave oscillation in the transonic inlet-diffuser and a forced oscillating NACA 64A010 airfoil. The shock wave frequency in the inlet diffuser and the unsteady coefficients of lift and moment of the airfoil are predicted well compared with experiment measurement.

The new low diffusion E-CUSP scheme, Zha CUSP scheme, uses scalar dissipation and is consistent with the characteristic direction of the disturbance propagation. It makes the computation more CPU efficient and robust compared with other popular upwind schemes. The Zha CUSP scheme is modified to Zha CUSP2 scheme to remove the temperature oscillation near wall boundary when grid is skewed. The Zha CUSP scheme is tested in 2D cases. It accurately resolve the boundary layer velocity and temperature profile using the first order differencing in a supersonic boundary layer flow. In a transonic converging-diverging nozzle, the oblique shock waves and the reflections are crisply captured even though the shock waves do not align with the mesh lines. The predicted wall surface isentropic Mach number is compared well with the experiment. In the transonic inlet diffuser case, the Zha CUSP scheme predicts the wall pressure as well as the Roe scheme and the AUSM+ scheme. The Zha CUSP2 scheme is tested in 3D cases, including a circular-to-rectangular nozzle, a subsonic compressor cascade and a transonic channel. The Zha CUSP scheme predicts the wall surface pressure distribution well in the circular-to-rectangular nozzle and the subsonic compressor cascade. The shock wave structure in the transonic channel is better predicted using the Zha CUSP2 scheme than using the Roe scheme. The Zha CUSP schemes are proved to be accurate, robust and efficient in these tests.

The steady and unsteady separation flow in a 3D stationary cascade under high incidence and different inlet Mach numbers is calculated by the solver. The steady state results capture the large separation when the incidence angle is large. The separation size increases with inlet Mach number in subsonic region. In supersonic flow, the shock wave-boundary layer interaction in the leading edge region pushes the separation region more downstream and decreases the size of the separation. These results are in good agreement with experiment. In the two subsonic unsteady simulation cases, the separation flow shows a sinusoidal pattern on the oscillation of the surface pressure and the separation bubble size. The leading edge vortex shedding is the mechanism behind the unsteady characteristics of the subsonic high incidence separated flow. The characteristics of the separation flow is determined by the inlet Mach number. The size and the oscillation intensity of the separation bubble increase with the increased Mach number in subsonic region. The flow is attached to the leading edge and a small separation bubble is located right after the shock wave when the flow goes further into supersonic.

The parallel computation solver is used to study the blade aeroelasticity with the fluid-structural interaction of a linear cascade with forced oscillating blades. The calculation is carried out with 2 incidences and 3 oscillating frequencies. The cascade has a constant IBPA (Inter Blade Phase Angle) of 180 degree. The calculation is first conducted in a simplified 2-passage cascade with periodical boundary condition applied under a medium frequency and a low incidence. The full scale cascade with 9 blades and 2 end walls are then calculated under more extensive conditions. The comparison between the simplified 2-passage result and the corresponding full scale cascade shows the strong influence of the end walls. Good spacial periodicity is achieved on the central passages in the full scale cascade results. The 9 blades in the cascade are stable in all full scale results. The spatial and temporal periodicity are decreased when the incidence increases. The suction surface separation brings high unsteadiness to the flow field. The damping coefficients increase with the increased oscillation frequency at low incidence, but decrease at high incidence. The unsteady pressure coefficient and the local stability analysis in all the full scale results compare well with experiment measurement in trend. The magnitude prediction does not agree very well with the experiment.

In this dissertation, the Zha CUSP schemes are verified to be accurate and efficient and are the first time to be applied in 2D and 3D calculations. The Zha CUSP schemes are the first time to be applied in moving grid systems. The implicit Gauss-Seidel iteration with dual time stepping is the first time to be used for moving grid systems. The NASA flutter cascade is the first time to be calculated in full scale.

To my dear wife, Peilin

for her endless love, understanding, support and encouragement

# Acknowledgment

There are so many people who I would like to give thanks to, because without them, the success of this research would not have been possible.

First and foremost, I would like to express my sincerest gratitude to Professor Gecheng Zha, my academic advisor, for his invaluable advice and guidance in my study and research. His continuous encouragement and inspiration always helped me break through the difficulties I encountered in the research work.

I would like to thank very much my wife, Peilin, for her understanding, support and encouragement during my three years' study and research in the University of Miami. I would not be able to concentrate on the research work without the time and energy she spent selflessly on my daily living.

I would like to thank Dr. Xiangying Chen for his enlightening discussions and suggestions on the applications of moving grid system and aeroelasticity. I also appreciate his precious help during my stay in Miami.

I would like to express my thanks to my dissertation committee members, Professor Hongtan Liu, Professor Xudong Yang and Professor Singiresu S. Rao, for their insightful comments and suggestions from their careful review of my research work.

I am also very grateful to all the people and friends who gave me help and convenience during my PhD study.

# Contents

# List of Figures

# List of Tables

# List of Symbols

| | |
|---|---|
| $\Delta V$ | control volume in Cartesian coordinates |
| $\mathbf{D}$ | source term |
| $\mathbf{E}, \mathbf{F}, \mathbf{G}$ | inviscid fluxes at $\xi$, $\eta$, $\zeta$ directions |
| $\mathbf{f}$ | body force vector |
| $\mathbf{l}, \mathbf{m}, \mathbf{n}$ | surface area vectors |
| $\mathbf{q}$ | conductive heat transfer vector |
| $\mathbf{R}, \mathbf{S}, \mathbf{T}$ | viscous fluxes at $\xi$, $\eta$, $\zeta$ directions |
| $\mathbf{RHS}$ | summation of the right hand side |
| $\mathbf{s}$ | surface area vector |
| $\mathbf{U}$ | conservative variable vector |
| $\mathbf{V}$ | flow velocity vector |
| $\mathrm{Pr}$ | Prantdl number |
| $\mathrm{Pr}_t$ | turbulent Prantdl number |
| $\mathrm{Re}$ | Reynolds number, real part of a complex value |
| $A$ | inviscid flux Jacobian matrix |
| $C$ | contravariant speed of sound, chord length |
| $c$ | speed of sound |
| $C_M$ | unsteady complex aerodynamic moment coefficient |
| $C_p$ | steady state static pressure coefficient |
| $c_p$ | constant pressure specific heat coefficient |
| $c_v$ | constant volume specific heat coefficient |
| $C_w$ | work coefficient |
| $CFL$ | Courant-Friedrichs-Levy condition number |
| $e$ | total energy per unit mass |
| $H$ | total enthalpy per unit mass |
| $h$ | static enthalpy per unit mass |
| $Im$ | imaginary part of a complex value |
| $J$ | coordinates transformation Jacobian |
| $k$ | Fourier coefficient of heat conductivity, turbulent kinetic energy |
| $k_c$ | reduced frequency |
| $k_t$ | turbulent conductivity |
| $L$ | reference length |
| $l$ | mixing length |
| $L, R$ | left and right eigenvector matrices |
| $M$ | Mach number, aerodynamic moment |

| | |
|---|---|
| $p$ | static pressure |
| $R$ | gas constant |
| $T$ | static temperature |
| $t$ | time |
| $t_c$ | characteristic time |
| $U, V, W$ | contravariant velocities in $\xi, \eta, \zeta$ directions |
| $u, v, w$ | velocity components in $x$, $y$, $z$ direction |
| $u^+$ | dimensionless velocity |
| $u_\tau$ | friction velocity |
| $U_\infty$ | free stream reference velocity |
| $x$ | dimensionless chordwise distance |
| $x, y, z$ | Cartesian coordinates |
| $y$,$y^+$ | dimensional and dimensionless distances to the wall |
| IBPA | inter-blade phase angle |

## Greek Symbols

| | |
|---|---|
| $\alpha$ | angle of attack, blade angular deflection |
| $\alpha, \beta$ | flow angles |
| $\beta$ | inter blade phase angle (IBPA) |
| $\Delta\Omega$ | control volume in generalized coordinates |
| $\delta_{ij}$ | Kronecker delta function |
| $\gamma$ | specific heat ratio |
| $\Lambda$ | eigenvalue matrix |
| $\lambda$ | eigenvalue |
| $\mu$ | molecular viscosity |
| $\mu_t$ | turbulent viscosity |
| $\nu$ | dynamic viscosity |
| $\omega$ | vorticity, turbulent dissipation rate |
| $\phi$ | phase angle of a complex value |
| $\Pi$ | stress tensor |
| $\rho$ | density |
| $\tau$ | time, pseudo time |
| $\tau_{ij}$ | viscous stress tensor |
| $\tau_w$ | wall shear stress |
| $\varepsilon$ | convergence residual criteria |
| $\Xi$ | damping coefficient |
| $\xi, \eta, \zeta$ | generalized coordinates |

## Subscripts

| | |
|---|---|
| $0$ | mean |
| $\infty$ | free stream reference variable |
| $A$ | amplitude |
| $b$ | value at boundary |
| $i$ | inner layer, value at the first inner cell |

| | |
|---|---|
| $i,j,k$ | cell indexes in $\xi$, $\eta$, $\zeta$ directions |
| $in$ | inlet |
| $L,R$ | values evaluated to the left and right sides of the surface |
| $o$ | outer layer |
| $out$ | outlet |
| $t$ | total (stagnation) value |
| $w$ | wall |
| 1 | first harmonic variable |
| R,I | real, imaginary parts of a complex value |

## Superscripts

| | |
|---|---|
| $'$ | fluctuation relative to the Reynolds average |
| $''$ | fluctuation relative to the Favre average |
| $*$ | dimensionless variable |
| $-$ | Reynolds average |
| $\tilde{}$ | Favre average, Roe average |
| $c$ | convective vector |
| $L,R$ | values evaluated to the left and right sides of the surface |
| $m,m+1$ | pseudo time step indexes |
| $n$ | blade index in cascade |
| $n,n+1$ | physical time step indexes |
| $p$ | pressure wave vector |

# Chapter 1

# Introduction

## 1.1 Objective

One of the most important requirements in developing modern turbomachines, either jet engines or stationary gas turbines, is the use of less primary energy by optimizing efficiency [1]. This results in higher compressor pressure ratio, which leads to highly loaded compressor stage with transonic or high-subsonic flow. Thinner blades are employed in the modern turbofan engines to achieve high pressure ratio under high Mach numbers. Their tip sections are usually thin and flat with low camber angles. However, these types of blades are more vulnerable to flutter.

Flutter is one of the most complicated and challenging areas in turbomachinery design and development. It is a highly undesirable and dangerous self-excited blade oscillation mode. The blade flutter has a strong negative influence on the turbomachinery performance. The high cycle fatigue problems resulting from the flutter are very detrimental to engine reliability and must be avoided. However, the origins of the flutter are not fully understood yet. Therefore, a great deal of flutter research has been carried out in recent years using experimental methods [2] or computational methods [3].

In high subsonic or transonic regime, when engine speed is reduced along operating line, the incidence increases with the reduced speed. The flow is very likely to separate on the suction surface at a part speed operating point. The large separation is considered as one of the reasons to trigger stall flutter. The high subsonic and transonic torsional stall flutter usually occurs near the fan stall limit line at speeds up to about 80% of the design speed [4]. Another factor to trigger the flutter phenomenon is the shock wave motion under transonic conditions. There are two distinct stall flutter regions in turbomachinery, subsonic/transonic stall flutter and supersonic stall flutter [5]. The subsonic/transonic stall flutter occurs at part speed operating conditions where the incidence angles are high and the inlet flow is high subsonic or transonic. The supersonic stall flutter occurs at high speed operating conditions in which the inlet flow is supersonic and detached leading edge shocks are present.

Flutter is a complicated aeroelastic problem with highly coupled fluid-structural interactions. To predict the flutter boundaries, a widely used method is the uncoupled energy method [6]. The unsteady aerodynamic forcing for given blade natural vibration modes are calculated and then the system stability is determined based on the net energy transfer. The difficulty in the flutter research is to accurately predict or measure the unsteady pressure distribution over the blade surfaces. Because the cascade test of unsteady transonic flow is extremely difficult and costly, the computational methods gain more and more interest in the area of flutter study. In recent years, tremendous advances have been achieved in computer technology and the numerical methods. Computational fluid dynamics (CFD) has become a valuable and widely used tool in turbomachinery design and research. Because of the unsteady separation and shock wave oscillation in the flutter phenomenon, the numerical simulation is difficult and time consuming. This requires advanced numerical algorithms with high accuracy and efficiency.

The objective of this dissertation is to develop an efficient and accurate Navier-Stokes solver to simulate the three-dimensional compressible unsteady flow field with non-linear fluid-structural interaction. The governing equations are discretized and solved in generalized coordinates with the dual time stepping method. A new efficient and accurate upwind scheme is developed to compute the inviscid flux. The new scheme will be compared with other popular upwind schemes. The three dimensional flow separation in a linear cascade will be studied using the solver. The solver will be implemented with parallel computation to compute the multi-passage flow in the linear cascade, in which the blades oscillates with a constant inter-blade phase angle(IBPA). The unsteady aeroelasticity of the blades under different flow angles and oscillating frequencies will be studied.

## 1.2 Fluid-Structural Interaction Simulation in Turbomachinery

The governing equations for unsteady flow modeling is the full Navier-Stokes equations. However solving the unsteady flow field is a very complicated and time consuming process. It becomes even more difficult when strong viscous effect is involved, for example when shock wave exists and it interacts with boundary layer. Unfortunately, this is the common flow condition in most of the turbomachinery flows. For simplification, the early researches are focused on solving the linearized Euler equations or the potential equations. With the development of the computer technology and the advances in the numerical algorithms, the numerical simulations in turbomachinery are able to solve the full 3D unsteady non-linear Navier-Stokes equations. Remarkable improvements have been made in recent years.

### 1.2.1 Potential equation models

The early unsteady flow studies date back to the 1970s. Because of the lack of powerful computers and efficient numerical methods, the governing equations used for the unsteady flow were the unsteady small disturbance potential equations. Compared with the Euler equations, or the Navier-Stokes equations, the potential equations are simpler and easier to solve. Some examples are the unsteady flows around airfoils or in cascade reported in [7,8]. This method was improved to solve the linearized unsteady full potential equations in the later work [9–12]. The potential equation solutions are more efficient in computing transonic flows with shock waves than the Euler equation solutions [13]. However, because the assumption of zero entropy change and vorticity production across shock wave it utilizes is not strictly correct [13], the potential equation method is not appropriate in applications with strong shock waves.

Different from the potential equation method, the linear and non-linear models based on the Euler or Navier-Stokes governing equations with primitive variables are used in the later researches.

### 1.2.2 Linear models

The commonly used unsteady flow models for aeroelastic investigations are limited to two-dimensional linearized methods in late 1980s. The time-dependent flow is linearized with respect to the steady flow. The unsteadiness in the flow is assumed to be a small disturbance of a known average flow and the full Euler or Navier-Stokes equations are approximated by a set of linear equations for the unsteady flow values [14]. The problems are further simplified by assuming harmonic time-dependency of all unsteady equations. After these simplification, the computations achieve good efficiency and are good for routine aeroelastic design studies.

The Early methods for unsteady flow over oscillating blades apply the time linearized models in a global [15] or local [16] sense. More recent study using the linearized Euler equation methods are the works of Hall et al. [17], Kahl et al. [18] followed by Montgomery et al. [19]. These studies are based on the linearized Euler equations. The works of Holmes et al. [20] and Clark et al. [21] are based on the linearized Navier-Stokes equations. Shabata and Kaji [22] analyzed the bending flutter for the tip section of a transonic fan using a linearized unsteady two-dimensional Euler analysis and found that the shock-induced unsteady aerodynamic loading on the airfoil suction surface produced the largest destabilizing force for unstarted supersonic inlet flows.

The linearized solver is very fast and robust even for transonic flow with shocks. However, because it imposes limits on the oscillation amplitudes and assumes no nonlinear flow phenomena produced by these oscillations, the solutions fail to predict the unsteady behaviour of the flow accurately when the amplitude of the oscillating blade is high or the pressure fluctuations due to shocks are larger than approximate 10% of the steady state solution [1].

### 1.2.3 Non-linear models

In reality, the unsteady flow field in turbomachinery may not respond to oscillating blades in a linear way owing to the strong nonlinear effects in unsteady transonic or supersonic flows. With the rapid advances in CFD and computer technology, the time-linearized models are gradually replaced by nonlinear flow models in recent years. The unsteady solutions of the full Euler and Navier-Stokes equations by time-marching methods become more and more popular.

Solutions of the Euler equations by time marching methods for unsteady flows in oscillating blades were developed by different researchers, including Jourbert [23], Koya et al. [24], Gerolymos [25], He [26], Peitsch et al. [27], Huff et al. [28], and Carstens [29]. Gerolymos [30] developed one of the early full 3D Euler methods followed by He et al. [31], Carstens [32], Peitsch et al. [33] and Peitsch et al. [34]. A sizable body of literature exists on the unsteady calculations of inviscid rotor-stator interaction phenomena [35–40].

The viscous effects in unsteady flow fields in turbomachines are taken into account by solving the Euler equations coupled with a boundary layer model or by solving the Reynolds averaged Navier Stokes equations directly. He et al. [41] calculated the flow around vibrating blades by coupling the Euler equations to an integral boundary layer solution. Ekaterinaris et al. [42] studied a vibrating rotor blade of a helicopter with stalled flow using a thin-layer Navier Stokes method. Abhari et al. [43] published their work on integrating the 2D Navier Stokes equations with a thin shear layer model.

The inviscid models are more efficient in computation than the viscous models. However, when strong shocks and separation flows are involved, the inviscid models do not perform as accurately as the viscous models. The inviscid solutions often over-predict the strength of the shock as reported by Brenneis [44]. Grüber et al. [14] studied the influence of viscous effects on the damping behavior of a transonic compressor cascade with tuned pitching oscillation. The results showed that the viscous effects may cause a significant change in the aerodynamic damping. Ji et al. [45] solved both Euler and Navier-Stokes equations for the flow in a transonic oscillating cascade.

As mentioned above, the blades in modern turbomachinery compressor usually work in transonic regime. The flow is characterized by the occurrence of shock, shock-boundary layer interaction and by flow separation on suction surfaces. These phenomena mix with blade oscillation and the flows become more complicated. To simulate these flows accurately, the viscosity and turbulence effects are necessary to be incorporated. The inviscid models should be replaced by the Navier-Stokes models in predictions of the aeroelastic performance of turbomachinery blades. The Navier-Stokes method gains more and more popularity in the simulations of complex, unsteady, turbulent flows [41].

Several time marching solutions for the unsteady Reynolds-averaged Navier-Stokes equations have been developed for blade row interactive flows and for oscillating cascade flows [26, 46–50]. More unsteady Navier-Stokes calculation are found in the works of Nakahashi et al. [51], Rai et al. [52] and Scott et al. [53] . Hwang et al. [54] solved 2D steady and unsteady turbine cascade flows by both Euler and Navier-Stokes equations with the Baldwin-Lomax turbulence model. Abhari et al. [43] computed aerodynamic damping using a hybrid structured and unstructured grid. He et al. [55] obtained 3D unsteady inviscid and viscous flow solutions over vibrating blades. Sanders [5] experimentally and numerically studied the stall flutter in a transonic low-aspect ratio fan blisk. The numerical simulations are performed using an unsteady three-dimensional nonlinear viscous solver with the effects of tip clearance and vibration amplitude.

Recent steady and unsteady stall computations for single airfoils have yielded remarkably improved agreement with measurement by utilizing appropriate turbulence models and laminar-turbulent transition models. The widely used turbulence models include the Baldwin-Lomax algebraic turbulence model [56], the one-equation Baldwin-Barth model

[57], the Spalart-Allmaras model [58] and the Wilcox's $\kappa - \omega$ model [59]. Some good results are reported in the works of Ekaterinaris et al. [60], Sanz et al. [61], Weber et al. [62, 63], Grüber et al. [14] and Ji et al. [45]. In the work of Grüber et al. [14], a self-induced shock oscillation due to shock-boundary layer interaction is simulated with the Baldwin-Lomax model. The prediction of the shock amplitude is significantly improved by a slight modification of the turbulence model. A 2D solver for the Reynolds averaged Navier Stokes equations using an extended turbulence model by Granville or a low Reynolds number $\kappa - \varepsilon$ model by Chien for the numerical analysis of unstalled flutter has been developed by Eguchi et al. [64]. Grüber et al. [65] have presented a 2D Navier Stokes solver with the Baldwin-Lomax turbulence model .

However the simulation for strong viscous flows are still fraught when compared with the experiments. Some aeroelasticity phenomena, including stall flutter, are still not predicted well. The uncertainties are caused by the laminar-to-turbulent transition model, the turbulent flow with strong flow oscillation and the three dimensional separation in flutter and the shock wave oscillation in high subsonic/transonic stall flutter [63].

In turbomachinery designs, even though the empirical relations are still widely used, efforts have been made to apply the fully unsteady nonlinear three-dimensional Navier-Stokes analyses to predict the stall flutter characteristics for advanced fans and compressors. Weber et al. [1] presented a time accurate Navier Stokes code for S1-stream surfaces to improve the numerical analysis and prediction of unsteady flow through turbomachine cascades with vibrating blades. It computes the sub and transonic viscous flow in a single passage on S1 stream surfaces of revolution following the theory of Wu [66] with a harmonically oscillating blade using a periodic boundary condition by Erdos et al. [67]. Silkowski et al. [68] described how the CFD based techniques can be applied to solve a wide variety of aeroelastic problems in the design phase. Their flutter predictions for a transonic fan agreed fairly well with the test data. Srivastava et al. [69] utilized an unsteady 3D Navier-Stokes analysis to investigate the flutter characteristics of an advanced forward swept fan design that encountered stall-side flutter.

### 1.2.4 Time marching methods

The computing cost is a particular concern when the full non-linear time dependent Navier-Stokes equations are solved for an unsteady flow. The computation cost becomes more severe for transonic flow cases in complicated 3D geometry with shock-boundary layer interaction, large separation and fluid-structural interaction.

The linear algebraic equations resulting from the discretization of the governing equations are solved with implicit or explicit methods. The implicit schemes allow the use of large time steps but usually require large computation time for each time step. The explicit schemes are efficient in time marching on each time step, but the magnitude of the time step is restricted by the Courant-Friedrichs-Levy (CFL) condition number to ensure the numerical stability.

The issue of time step becomes more important in the strong viscous flows where flow field characteristics are highly influenced by the viscous layers in the vicinity of solid surfaces or shock waves. The mesh should be refined in these high viscous regions to capture the flow details. The mesh size will vary in the magnitude of several orders in the computation domain. Because the explicit time accurate methods must advance all cells in the computational domain with a uniform time step to maintain consistency, the global time step limited by the most restrictive of allowable time steps in the domain will be quite small. Usually this time step will be much smaller than the time step required by the solution accuracy of the problems [70]. This make the explicit method very inefficient in unsteady simulation.

For steady flow calculation, where the time accuracy is not required, the CFL limitation can be relieved. The numerical convergence can be improved by several pseudo-time-wise acceleration techniques, such as multi-grid [71, 72], non-uniform local time-stepping [73]. The multi-grid method accelerates the convergence by applying multiple layers of grids in numerical iterations. The low frequency error on the fine grid is effectively removed by the solution on the overlayed coarser grid. In the local time stepping method, the convergence is accelerated by advancing the solution by the local maximum possible time step on each grid point, instead of the uniform time step for the whole domain.

These techniques are not directly applicable in unsteady simulations where time accuracy is required. The time-wise implicit schemes can then be used to overcome the time step limitation. However the overall efficiency of the implicit schemes is still low because of the large amount of matrix inversion. The 3D linearized algebraic equations are usually solved using the Approximate Factorization (AF) schemes [74, 75]. However the factorization error is time step magnitude dependent. The time accuracy and the convergence are affected when larger time step is used.

Jameson formulated a so called dual stepping scheme [76] to combine the advantages of both the explicit and implicit schemes to achieve an efficient and accurate unsteady time marching. At the first step, the explicit time step limitation is bypassed by formulating the unsteady problem in an implicit form. Then at each time step, the implicit set of the equations are treated as a modified steady state problem on an inner pseudo time step. In this way, all the acceleration techniques for explicit schemes are able to be applied in the iterations over the pseudo time step. Great improvement in the computation performance is achieved without losing time accuracy [70]. This method is originally developed for Euler equations [76] and is further applied to Navier-Stokes equations [77]. Currently this scheme is adopted by many researchers in their unsteady elastic studies. Aloso et al. [70] calculated the unsteady shedding flow behind a circular cylinder and the unsteady aerodynamics of a pitching NACA 64A010 airfoil. McBean et al. [78] simulated a 2D unsteady oscillating turbine cascade. Ji et al. [45] calculated the flow in an oscillating compressor cascade. Cinnella et al. [79] adopted this method in their turbomachinery aeroelasticity numerical method.

### 1.2.5 3D calculations

The turbomachinery blade channel flows in reality exhibits high three-dimensional characteristics, for example, 3D flow separation. It is desirable to simulate the aerodynamic and aeroelastic problem with 3D codes. However, solving the fully 3D problems, which is very time consuming, requires high computation cost. A good simplification is to use the so-called quasi-3D Navier-Stokes method. The governing equations are expressed on the S1-stream surfaces of revolution. The three-dimensional flow effects are represented by a stream surface thickness varying with the streamwise location [63].

Joubert performed an unsteady quasi-3D Euler calculation and showed that the shock oscillation is indeed a source of blade excitation [23]. Weber et al. presented a quasi-3D solver and validated the solver by simulating the Standard Configuration No. 4 [1]. Then the solver is further used to calculated the flow over an oscillating cascade under different frequencies and incidences [63]. The results obtained by the quasi-3D solver yielded good agreement with the steady and unsteady measurement. Isomura [80] studied the bending mode flutter of a modern transonic fan using a quasi-3D viscous unsteady CFD code. More application of the quasi-3D method can be found in the works of He [31], Abhari and Giles [43], Grüber et al. [65], Tuncer et al. [81], Carstens et al. [82], Lin et al. [83] and Ji [45].

Three-dimensional flutter computations are presently being developed by a limited number of researchers. Gerolymos et al. [84] and Chuang et al. [85] studied the turbomachinery aeroelasticity with 3D Euler simulations. McBean et al. [86] developed a 3D Navier-Stokes code for the aeroelasticity in turbomachinery. The code is used to calculate the Srandard Configuration 4 in 3D annular turbine cascade [87]. However, the 3D flutter computation is still premature for the analysis of stall flutter.

### 1.2.6 Phase-shift boundary condition

In the unsteady turbomachinery studies, the flutter calculation usually follows Lane's traveling wave model [88]. The blades are assumed to vibrate with the same frequency and a constant IBPA. When the IBPA is not zero, two neighboring blades oscillate with a constant phase shift, which makes the conventional periodic boundary not applicable in the blade oscillation unsteady simulation. In order to calculate such flows within a single passage, a phase-shift boundary condition is applied instead. The phase-shift boundary condition implies periodicity spatially and temporally.

The most widely used implementation of the phase-shift boundary condition is suggested by Erdos et al. [67] and is called the “direct store” method. The primitive flow variables on the periodic boundary over a whole period of time are saved and the boundary variables for current solutions are updated based on the saved variables by considering the current time and the specific IBPA. The implementation of this method is straightforward. However it requires a large amount of computer storage, especially for explicit time marching schemes. This method is widely adopted by different researchers. Ji et al. [45] applied this method in their oscillating transonic cascade Navier-Stokes calculation and compared its performance with a parallel computing method. Weber et al. [63] used this method to calculated a stall flutter cascade. More applications of the “direct store” method can be found in the works of He et al. [26] and Abhari et al. [43].

The storage requirement can be relieved very much by using the dual-time stepping method, where the physical time step is much larger than the explicit method. However, the computation time to reach the final periodic solution may be prolonged by the direct store method [45]. The phase-shift periodic boundary conditions is applied in an iterative manner. The stored flow field history used to set the boundary condition for the computation of the next period is form a less converged solution.

In order to save storage, Giles [89] developed a space-time transformation method. The computation time plane is inclined along the blade pitch direction according to a given IBPA. The phase-shifted boundary condition is applied as the ordinary zero phase-shift periodic condition. However, the time-inclination angle and therefore the IBPA is restricted by the characteristics of the governing equations. He [26] suggested a shape correction method. The Fourier components of the approximate Fourier series of the flow variables instead of the time-wise flow parameters are stored. The computer storage is greatly reduced, but extra computation time is added.

With the advance in computer technology, the parallel computers or the clustered computer networks with distributed memory become popular. A cost efficient parallel computation with the message passing interface (MPI) protocol [90] provides a powerful way to calculate the multiple passage flows. By assigning each passage a processor, the multi-passage flow calculation can be carried out simultaneously for all passages involved. After each iteration, the neighboring passages exchange their boundary information simultaneously. The phase-shift boundary assumption is eliminated. The computation is easier to implement and the results are more close to reality. In parallel computation, the computing jobs should be evenly distributed on all involved processors to achieve better scalability. With the availability of parallel computers consisting hundreds or even thousands of processors, it become feasible to simulated a complete blade row or even multiple stages in turbomachines. In such cases, the parallel computation is very flexible. It can be based on a parallel computer with multiple CPUs and shared memory installed or on a cluster of workstations with distributed memories. The parallel computation is adopted in the works of Ji et al. [45], McBean et al. [78, 86, 87] and Cinnella et al. [79].

The parallel computation becomes even more useful when the mistuned oscillation is considered [3]. In reality, the spatial and temporal periodic assumption is not valid in most cases. The flow may be non-periodic in time or space or it is periodic but with multiple non-harmonic frequencies. The phase-shift boundary condition may not be valid any more. The flow will have to be calculated individually in multiple passages. This can be easily and efficiently handled by parallel computation.

## 1.3 Upwind Schemes

The fundamental problem in solving Euler or Navier-Stokes equations with control volume method is to compute the surface flux correctly and efficiently. Development of an accurate and efficient numerical scheme for compressible flow governing equations is essential due to the increasing engineering demand for aircraft and spacecraft design [91]. Such a scheme is particularly important when aircraft engine aeroelasticity problems are simulated using a fully coupled fluid-structural interaction model, which is usually very CPU intensive.

The early schemes used in 1960-1970's are mainly based on central differencing. They are inherently unstable and artificial dissipation is required to suppress the oscillation near a shock. The artificial dissipation is usually difficult to devise appropriately and it may smear out the captured shock wave [92]. The upwind schemes are designed to make the flux computation follow the characteristic directions of the governing equations and are employed in many of the current CFD codes.

The upwind schemes are categorized into flux-difference splitting (FDS) schemes and flux-vector splitting (FVS) schemes. In the FDS schemes, the changes in the flux quantities at the control volume inter surface are interpreted as being caused by a series of waves. The FVS schemes are interpreted as schemes that transport particles according to the characteristics information [93, 94]. The upwind schemes have inherent numerical dissipation, which makes the artificial dissipation unnecessary. However, problems will be encountered when the numerical dissipations are too large and the real dissipations are distorted [92].

Roe developed one of the most well known FDS schemes, the Roe scheme, in 1981 [95]. The numerical flux expression incorporates upwind influence through the addition of contributions across positive and negative waves. The matrix of the dissipation vector is calculated using the Roe-average variables. The Roe scheme was considered as the most accurate scheme among the available differencing schemes in 1987 [96]. The disadvantage of the Roe scheme is its time consuming matrix operation.

The FVS schemes split the flux contributions into positive and negative components based on the eigenvalue structure of the system or some other appropriately assumed behavior [94]. Because they use scalar dissipation instead of matrix dissipation, the FVS schemes are simpler in formulation and more efficient than the accurate Roe scheme.

Steger et al. [74] developed the Steger-Warming scheme in 1979. They split the flux vector into two sub-vectors according to the information travel direction. The two sub-vectors are based on the positive and negative parts of the eigenvalue matrix respectively. The fatal disadvantage of the Steger-Warming scheme is its large numerical dissipation, which does not vanish in any discontinuities. It smears out the discontinuities and distorts the viscous solutions. Besides, the Steger-Warming scheme is not continuously differentiable at sonic and stagnation points, therefore a glitch will be yielded at these points. The van Leer scheme suggested by van Leer in 1982 [97] removes the discontinuities and make one of the eigenvalues vanish in subsonic region, which results in sharper shock compared with the Steger-Warming scheme. However, the van Leer scheme is also dissipative. It works very well in inviscid flow fields, but broadens the viscous boundary layer. A low diffusion FVS scheme was developed by Zha and Bilgen in 1993 [98]. The Zha-Bilgen scheme has the individual mass flux vanishing when the velocity approaches zero. It hence achieves very low numerical dissipation. However, the Zha-Bilgen scheme may generate non-smoothness in multi-dimensional flows.

Efforts have been made to improve the original van Leer scheme by using some techniques borrowed from FDS schemes. The original van Leer scheme is further improved by other researchers. For example, the van Leer-Hänel scheme [99], which uses the net mass flux and one side velocity and total enthalpy for the transverse momentum and energy equations, achieves an accurate temperature profile for the supersonic conical viscous flow. More recent FDS schemes include the AUSM family schemes of Liou represented by their latest scheme of AUSM$^+$ [100–104], Edwards's LDFSS schemes [105], Jameson's CUSP schemes and limiters [106, 107], and the schemes developed by Zha, et al. [108, 109], etc.

Pioneered by Liou and Steffen [100, 102–104], the researchers seeking the scalar dissipation primarily follow the guideline that the velocity and pressure should be separated to consider their characteristics representing the physics of the convection and waves. Liou and his colleagues termed their schemes as advection upstream splitting method(AUSM) schemes, and Jameson gave the name of convective upwind and split pressure (CUSP) schemes [106,107]. The name of CUSP seems more precisely reflecting the physical meaning.

As pointed out by Jameson [106, 107], the CUSP schemes can be basically categorized to two types, the H-CUSP schemes and E-CUSP schemes. The H-CUSP schemes has the total enthalpy from the energy equation in their convective vector, while the E-CUSP schemes use the total energy in the convective vector. The Liou's AUSM family schemes, van Leer-Hänel scheme [99], and Edwards's LDFSS schemes [105] belong to the H-CUSP group. The schemes developed by Zha [98, 108] belong to the E-CUSP group. Jameson suggested schemes for both groups [106, 107].

The H-CUSP schemes such as AUSM family schemes may have the advantages to better conserve the total enthalpy for steady state flows and have achieved great success. However, from the characteristic theory point of view, these schemes are not fully consistent with the disturbance propagation directions, which may affect the stability and robustness of the schemes [110]. A H-CUSP scheme may have more inconsistency when it is extended to moving grid system. It will leave a pressure term multiplied by the grid velocity in the energy flux that is not contained in the total enthalpy and the term will be treated as a part of the pressure term. From characteristics point of view, it is not obvious how to treat this term in a consistent manner [111]. By splitting the eigenvalues of the Jacobians to convection (velocity) and waves (speed of sound), one will find that the convection terms only contain the total energy [98], which will lead to the E-CUSP schemes. However, the early E-CUSP schemes could not handle contact discontinuities [98, 106, 107]. Borrowing from AUSMDV scheme [101], Zha used the interface speed of sound and made his E-CUSP scheme able to capture exact contact discontinuities [108, 109]. However, due to lacking proper numerical dissipation, the scheme will generate odd-even pressure oscillations when applied to multi-dimensional flows.

Recently, Zha and Hu [110] suggested an efficient E-CUSP scheme (named as Zha CUSP) which is consistent with the characteristic directions. The scheme has low diffusion and is able to capture crisp shock profiles and exact contact discontinuities. The scheme is shown to be accurate, robust and efficient. In addition, the Zha CUSP scheme is straightforward to be extended to moving grid system [111]. The original Zha CUSP scheme is further modified to remove the temperature oscillations occurring occasionally near walls, in particular when the mesh is skewed. Zha modified the Zha CUSP scheme to Zha CUSP2 scheme by replacing the pressure in the dissipation of energy equation with the total enthalpy [112]. The modified scheme also yields more precise wall surface temperature even at coarse grid.

## 1.4 Summary of Present Work

In the current work, a 3D unsteady Navier-Stokes solver is developed as the first step towards the aim to study the full non-linear turbulent fluid-structural interaction in turbomachinery. The contents of the current work are summarized below.

The unsteady 3D Navier-Stokes governing equations are derived from the fundamental conservation laws of mass, momentum and energy. The equations are then taken the Favre-average to consider turbulence effects. The Baldwin-Lomax turbulence model is applied for the turbulence modeling. Finally the governing equations are expressed in the generalized coordinates to handle complex geometry.

The inviscid flux on the control volume surface is calculated using upwind schemes. A high resolution low diffusion Zha CUSP upwind scheme is developed and updated to achieve high computation efficiency and accuracy. The new scheme strictly follows the characteristic directions of the disturbance propagation, which is believed to improve the computation stability. The numerical diffusion of the new scheme is lower than the Roe scheme and it is more efficient in computation than the Roe scheme.

The governing equations are discretized with the finite volume method. The resulting linear algebraic equations are implicitly solved using the Gauss-Seidel line iteration method. The convergence is accelerated using the technique of local time stepping. The third order of accuracy is achieved in the space differencing for inviscid flux using the MUSCL approach. The second order of central differencing is adopted for the viscous flux. The unsteady time marching is carried out with the dual time stepping method to achieve the second order of temporal accuracy. A pseudo time step is added within each physical time step. The numerical iterations are conducted over the pseudo time step until a converged solution is obtained to finish one physical step time marching. The solver is made parallel using the MPI protocol. The moving grid system is implemented in the solver.

The code is validated with a series of test cases for its turbulence modeling, parallel computation scalability and unsteady simulation. The turbulence modeling is validated using a supersonic flat plate boundary layer flow and an inlet-diffuser transonic flow. The flat plate boundary layer flow is further used to test the parallel speedup scalability of the solver. The self-excited shock wave oscillation in the stationary inlet-diffuser and the unsteady flow over a forced oscillating airfoil are calculated to validate the unsteady calculation capability.

The newly developed Zha CUSP schemes are compared with other popular upwind schemes. The 2D cases include a flat plate boundary layer flow, a transonic converging-diverging nozzle and the transonic inlet-diffuser. The 3D cases include a transonic channel, a circular-to-rectangular duct and a 3D compressor cascade.

The 3D Navier Stokes solver is then used to study the separation flow in the 3D transonic cascade under high incidence. Both steady and unsteady simulations are conducted for a central passage in the cascade with different inlet Mach numbers. The computation results are compared with experiment measurement. The unsteady periodic vortex shedding mechanism behind the unsteady separation phenomenon is studied in details.

The aeroelasticity of the blades vibrating within the linear NASA flutter cascade is calculated in parallel. The simulation is carried out in 2D condition. The simplified 2-passage cascade is first calculated with a medium frequency and a low incidence angle. The mesh dependence of the computation results are studied with the simplified cascade. More extensive studies are carried out for the influence of the oscillation frequencies and the incidence angles on the blade aeroelasticity. The numerical results are compared with experiment measurement.

In this dissertation, the Zha CUSP schemes are the first time to be applied in 2D and 3D calculations. The Zha CUSP schemes are the first time to be applied in moving grid systems. The implicit Gauss-Seidel iteration with dual time stepping is the first time to be used for moving grid systems. The NASA flutter cascade is the first time to be calculated in full scale.

# Chapter 2

# Governing Equations

In the current work, both two dimensional and three dimensional problems are numerically studied. The code is developed in 3D form for generality. The two dimensional computation is made possible by assuming unit length in the third dimension. In this chapter, the governing equations are derived from the fundamental physical conservation laws of mass, momentum and energy. They are expressed in conservative forms to be able to capture discontinuities in compressible flows. The governing equations are normalized with free stream variables. The generalized coordinates are employed to transform an arbitrary physical domain to a well-constructed computation domain. The governing equations are transformed to and are discretized in the computation domain with the generalized coordinates. The Baldwin-Lomax turbulence model is applied to simulate the flow turbulence. The moving grid system is employed in the solver to study the cases with moving solid boundaries.

## 2.1 3D Navier-Stokes Equations

The fundamental governing equations are the compressible Navier-Stokes equations derived from the physical laws of the continuity of mass, conservation of momentum and the conservation of energy. They are expressed as the following [94],

1. Conservation of mass (continuity of mass):

$$\frac{D\rho}{Dt} + \rho\left(\nabla \cdot \mathbf{V}\right) = 0 \tag{2.1}$$

2. Conservation of momentum (Newton's second law of motion)

$$\rho \frac{\mathrm{D}\mathbf{V}}{\mathrm{D}t} = \rho \mathbf{f} + \nabla \cdot \Pi \tag{2.2}$$

3. Conservation of energy (the first law of thermodynamics)

$$\frac{\mathrm{D}\left(\rho e\right)}{\mathrm{Dt}} = -\nabla \cdot \mathbf{q} + \rho \mathbf{f} \cdot \mathbf{V} + \nabla \cdot \left(\Pi \cdot \mathbf{V}\right) \tag{2.3}$$

In the above equations, $t$ is time, $\rho$ is the density, $\mathbf{V}$ is the flow velocity vector,

$$\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k} \tag{2.4}$$

$u$, $v$, $w$ are the velocity components in $x$, $y$, $z$ direction respectively. $\mathbf{q}$ is the conductive heat flux vector,

$$\mathbf{q} = q_x\mathbf{i} + q_y\mathbf{j} + q_z\mathbf{k} \tag{2.5}$$

$\mathbf{f}$ is the body force. $e$ is the total energy per unit mass,

$$e = c_v T + \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{2.6}$$

where $c_v$ is the constant volume specific heat coefficient, $T$ is the static temperature.

The stress tensor $\Pi$ is frequently separated as the following,

$$\Pi_{ij} = -p\delta_{ij} + \tau_{ij} \tag{2.7}$$

where $p$ is the static pressure, Indies $i = 1,2,3$ and $j = 1,2,3$ indicate $x$, $y$ and $z$ directions,$\delta_{ij}$ is the Kronecker delta function,

$$\delta_{ij} = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{when } i \neq j \end{cases} \tag{2.8}$$

$\tau_{ij}$ is the viscous stress tensor and is written as,

$$\tau_{ij} = \mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \quad i,j,k = 1,2,3 \tag{2.9}$$

$\mu$ is the molecular viscosity and is computed by the Sutherland's law [113],

$$\mu = C_1 \frac{T^{3/2}}{T + C_2} \tag{2.10}$$

where $C_1$ =1.458×$10^{-6}$kg/(m · s · $\sqrt{\text{K}}$), $C_2$ =110.4 K.

The substantial derivative $\frac{\mathrm{D}}{\mathrm{D}t}$ is defined as,

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + u_k \frac{\partial}{\partial x_k} \tag{2.11}$$

The repeated index $k$ stands for the summation over $x$, $y$, $z$ direction ($k = 1,2,3$) following the Einstein convention.

The heat flux $\mathbf{q}$ is given by the Fourier's law,

$$\mathbf{q} = -k \nabla T \tag{2.12}$$

where $k$ is the Fourier coefficient of heat conductivity.

Substitute Eqs. (2.11) through (2.12) into the governing equations (2.1) through (2.3) and neglect the body force $\mathbf{f}$, the governing equations are re-written into the following conservative forms,

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_k}{\partial x_k} = 0 \tag{2.13}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_k}{\partial x_k} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ik}}{\partial x_k} \tag{2.14}$$

$$\frac{\partial \rho e}{\partial t} + \frac{\partial \left(\rho e + p\right) u_k}{\partial x_k} = \frac{\partial \left(q_k + \tau_{jk} u_j\right)}{\partial x_k} \tag{2.15}$$

The governing equations are further closed by the equations of state with the assumption of perfect gas,

$$p = \rho R T \tag{2.16}$$

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \left(u^2 + v^2 + w^2\right) \tag{2.17}$$

where $R$ is the gas constant $\gamma$ is the specific heat ratio taken the value of 1.4 for air.

## 2.2 The Favre-Averaged Navier-Stokes Equations

When the Reynolds number is high enough, the flow becomes turbulent. A large range of temporal and spatial scales exist in the flow domain. It is not feasible to solve the 3D Navier-Stokes equations directly for engineering turbulent flows, because the mesh size required will be too large and the computation will be too expensive. In engineering point of view, the macro average flow properties are of more interest. Therefore, the governing equations are reformulated using appropriate averaging process. The turbulent effects are considered with the help of turbulence models.

The Reynolds average is a long time average for any flow variable. The Favre average based on the long time mass average is adopted for compressible flows [114].

The Favre Average of an fluctuating variable $f$ is defined as,

$$\tilde{f} = \frac{1}{\bar{\rho}} \int_{t_0}^{t_0+\Delta t} \rho f \,\mathrm{d}t = \frac{\overline{\rho f}}{\bar{\rho}} \tag{2.18}$$

where the tilde stands for the Favre average and the overbar stands for the Reynolds average. The average is taken over a time interval $\Delta t$ starting from an arbitrary time $t_0$. $\Delta t$ is significantly larger than the period of the fluctuating quantities but smaller than the time interval associated the mean flow.

The Reynolds averaging is defined as,

$$\bar{f} = \frac{1}{\Delta t} \int_{t_0}^{t_0+\Delta t} f \,\mathrm{d}t \tag{2.19}$$

The instantaneous variables in the governing equations (2.13) through (2.15) are then decomposed into an averaged mean and a local fluctuation term. The averaged means of the density $\rho$, static pressure $p$, stress tensor $\tau_{ij}$ and heat flux $q_i$ are based on the Reynolds averaging,

$$\rho = \bar{\rho} + \rho', \; p = \bar{p} + p', \; \tau_{ij} = \bar{\tau}_{ij} + \tau'_{ij}, \; q_i = \bar{q}_i + q'_i \tag{2.20}$$

For the velocity components $u_i$, the internal energy $e$, the molecular viscosity coefficient $\mu$, the static temperature $T$ and the heat conductivity $k$, the averaged means are based on the Favre averaging,

$$u_i = \tilde{u}_i + u''_i, \; e = \tilde{e} + e'', \; \mu = \tilde{\mu} + \mu'', \; T = \tilde{T} + T'', \; k = \tilde{k} + k'' \tag{2.21}$$

Substitute the variable decompositions of Eq. (2.20) and Eq. (2.21) into the governing equations (2.13) through (2.15). After rearrangement and simplification, the Favre averaged Navier-Stokes Equations are expressed as the following,

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_j}{\partial x_j} = 0 \tag{2.22}$$

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i\tilde{u}_k}{\partial x_k} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial\left(\bar{\tau}_{ik} - \overline{\rho u_i'' u_k''}\right)}{\partial x_k} \tag{2.23}$$

$$\frac{\partial \bar{\rho}\tilde{e}}{\partial t} + \frac{\partial\left(\bar{\rho}\tilde{e} + \bar{p}\right)\tilde{u}_k}{\partial x_k} = \frac{\partial\left[\tilde{u}_j(\bar{\tau}_{jk} - \overline{\rho u_j'' u_k''}) + \left(\bar{q}_k - c_p\overline{\rho u_k'' T''}\right)\right]}{\partial x_k} \tag{2.24}$$

where $c_p$ is the constant pressure specific heat coefficient.

Compared with Eqs. (2.13) through (2.15), the two extra terms in the Favre averaged Navier-Stokes equations are the turbulent shear stress $-\overline{\rho u_i'' u_k''}$ (Reynolds stress) and turbulent heat flux $-c_p\overline{\rho u_k'' T''}$ , which represent the turbulent transport of momentum and energy. In the eddy viscosity based turbulence models, these two terms are related to the gradient of the mean flow variables and are expressed in a form similar to the mean flow shear stress and mean flow heat flux based on the Boussinesq assumption [113].

$$-\overline{\rho u_i'' u_k''} = \mu_t\left(\frac{\partial \tilde{u}_i}{\partial x_k} + \frac{\partial \tilde{u}_i}{\partial x_i} - \frac{2}{3}\frac{\partial \tilde{u}_j}{\partial x_j}\delta_{ik}\right) \tag{2.25}$$

$$-c_p\overline{\rho u_k'' T''} = k_t\frac{\partial \tilde{T}}{\partial x_k} \tag{2.26}$$

where $\mu_t$ is called the turbulent viscosity, $k_t$ is the turbulent conductivity,

$$k_t = \frac{\mu_t c_p}{\mathrm{Pr}_t} \tag{2.27}$$

$\mathrm{Pr}_t$ is the turbulent Prandtl number taking the value of 0.9.

Substitute Eqs. (2.25) and (2.26) into the governing equations (2.23) and (2.24). The Favre-Averaged Navier-Stokes equations can be transformed to a similar form like that in laminar flow by replacing the molecular viscosity $\mu$ in the stress terms by $\mu + \mu_t$ and the term $\mu/\mathrm{Pr}$ in the heat conduction term by $\mu/\mathrm{Pr} + \mu_t/\mathrm{Pr}_t$. Pr is the Prantdl number.

Eqs. (2.23) and (2.24) are re-written as the following,

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i\tilde{u}_k}{\partial x_k} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \hat{\tau}_{ik}}{\partial x_k} \tag{2.28}$$

$$\frac{\partial \bar{\rho}\tilde{e}}{\partial t}+\frac{\partial\left(\bar{\rho}\tilde{e}+\bar{p}\right)\tilde{u}_k}{\partial x_k}=\frac{\partial\left(\tilde{u}_j\hat{\tau}_{jk}+\hat{q}_k\right)}{\partial x_k} \tag{2.29}$$

where $\hat{\tau}_{ik}$ and $\hat{q}_k$ are the total shear stress and total heat flux in turbulent flows,

$$\hat{\tau}_{ik}=\left(\mu+\mu_t\right)\left[\left(\frac{\partial u_i}{\partial x_k}+\frac{\partial u_k}{\partial x_i}\right)-\frac{2}{3}\delta_{ik}\frac{\partial u_j}{\partial x_j}\right] \tag{2.30}$$

$$\hat{q}_k=\left(\frac{\mu}{\mathrm{Pr}}+\frac{\mu_t}{\mathrm{Pr}_t}\right)\frac{\partial T}{\partial x_k} \tag{2.31}$$

The value of $\mu_t$ is determined by an appropriate turbulence model. The zero-equation Baldwin-Lomax model [56] is applied in the current work and will be covered later in this chapter.

## 2.3 Normalization of the Governing Equations

The Favre-averaged governing equations (2.22), (2.28) and (2.29) are normalized before discretization. The hat, overbar and tilde are all dropped for clarity in the rest of the current work.

The dimensionless flow variables in the governing equations are defined as the following,

$$\begin{array}{lll} x^*=\dfrac{x}{L} & y^*=\dfrac{y}{L} & z^*=\dfrac{z}{L} \\ u^*=\dfrac{u}{U_\infty} & v^*=\dfrac{v}{U_\infty} & w^*=\dfrac{w}{U_\infty} \\ \rho^*=\dfrac{\rho}{\rho_\infty} & \mu^*=\dfrac{\mu}{\mu_\infty} & t^*=\dfrac{t}{L/U_\infty} \\ T^*=\dfrac{T}{T_\infty} & p^*=\dfrac{p}{\rho_\infty U_\infty^2} & e^*=\dfrac{e}{U_\infty^2} \end{array}$$

where the superscript asterisk denotes dimensionless variable, the free stream conditions are denoted by $\infty$, $L$ is the reference length used in the Reynolds number $\mathrm{Re}_L$,

$$\mathrm{Re}_L = \frac{\rho_\infty U_\infty L}{\mu_\infty} \tag{2.32}$$

Based on Eq. (2.25), the dimensionless turbulent viscosity $\mu_t^*$ is defined as,

$$\mu_t^* = \frac{\mu_t}{\rho_\infty U_\infty L} \tag{2.33}$$

Substitute the above normal variables into Eqs. (2.22), (2.28), (2.29), (2.30) and (2.26). After simplification, the normalized governing equations are written as the following,

$$\frac{\partial \rho^*}{\partial t^*} + \frac{\partial \rho * u_j^*}{\partial x_j^*} = 0 \tag{2.34}$$

$$\frac{\partial \rho^* u_i^*}{\partial t^*} + \frac{\partial \rho^* u_i^* u_k^*}{\partial x_k^*} = -\frac{\partial p^*}{\partial x_i^*} + \frac{1}{\mathrm{Re}_L}\frac{\partial \tau_{ik}^*}{\partial x_k^*} \tag{2.35}$$

$$\frac{\partial \rho^* e^*}{\partial t^*} + \frac{\partial \left(\rho^* e^* + p^*\right) u_k^*}{\partial x_k^*} = \frac{1}{\mathrm{Re}_L}\frac{\partial \left(u_j^* \tau_{jk}^* + q_k^*\right)}{\partial x_k^*} \tag{2.36}$$

$$\tau_{ik}^* = \left(\mu^* + \mu_t^* \mathrm{Re}_L\right)\left[\left(\frac{\partial u_i^*}{\partial x_k^*} + \frac{\partial u_k^*}{\partial x_i^*}\right) - \frac{2}{3}\delta_{ik}\frac{\partial u_j^*}{\partial x_j^*}\right] \tag{2.37}$$

$$q_j^* = \frac{1}{(\gamma - 1) M_\infty^2}\left(\frac{\mu^*}{\mathrm{Pr}} + \mathrm{Re}_L \frac{\mu_t^*}{\mathrm{Pr}_t}\right)\frac{\partial T^*}{\partial x_j^*} \tag{2.38}$$

where $M_\infty$ is the reference Mach number,

$$M_\infty = \frac{U_\infty}{\sqrt{\gamma R T_\infty}} \tag{2.39}$$

which does not need to be the precise free stream Mach number.

The perfect gas equation of state is re-written in the dimensionless form as,

$$T^* = \frac{\gamma M_\infty^2 p^*}{\rho^*} \tag{2.40}$$

The dimensionless total energy per unit mass $e^*$ is related with the dimensionless temperature $T^*$ and velocity $u^*$, $v^*$, $w^*$ as,

$$e^* = \frac{1}{\gamma(\gamma-1)M_\infty^2}T^* + \frac{1}{2}\left(u^{*2}+v^{*2}+w^{*2}\right) \tag{2.41}$$

The dimensionless pressure $p^*$ is related with the dimensionless density $\rho^*$, energy $e^*$ and velocity $u^*$, $v^*$, $w^*$ as,

$$p^* = (\gamma-1)\left[\rho^* e^* - \frac{1}{2}\rho^*\left(u^{*2}+v^{*2}+w^{*2}\right)\right] \tag{2.42}$$

The dimensionless speed of sound $c^*$ is calculated as the following,

$$c^{*2} = \frac{T}{(\gamma-1)M_\infty^*} \tag{2.43}$$

Drop the superscript asterisk, the vector form of the normalized governing equations are,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y} + \frac{\partial \mathbf{T}}{\partial z} + \mathbf{D} \tag{2.44}$$

where **U** is the conservative variable vector, **D** is an additional source term used later for the moving grid system or the $k-\omega$ turbulence model. **E**, **F**, **G** and **R**, **S**, **T** are the inviscid and viscous flux vectors in $x$, $y$ and $z$ direction respectively.

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{bmatrix} \tag{2.45}$$

$$\mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho e + p)\, u \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho e + p)\, v \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho e + p)\, w \end{bmatrix} \tag{2.46}$$

$$\mathbf{R} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u_k \tau_{xk} - q_x \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u_k \tau_{yk} - q_y \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u_k \tau_{zk} - q_z \end{bmatrix} \tag{2.47}$$

The repeated index $k$ stands for the Einstein summation over $x$, $y$ and $z$. The Reynolds number $\mathrm{Re}_L$ is absorbed into the stress $\tau$ and heat flux $q$ as,

$$\tau_{ik} = \left( \frac{\mu}{\mathrm{Re}_L} + \mu_t \right) \left[ \left( \frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right) - \frac{2}{3} \delta_{ik} \frac{\partial u_j}{\partial x_j} \right] \tag{2.48}$$

$$q_j = \frac{1}{(\gamma - 1) M_\infty^2} \left( \frac{\mu}{\mathrm{Re}_L \mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t} \right) \frac{\partial T}{\partial x_j} \tag{2.49}$$

## 2.4 Governing Equations in Generalized Coordinates

To apply Eqs. (2.44) in computation domain with more complicated geometry, they are further transformed from the Cartesian coordinates $(x, y, z)$ to the generalized coordinates $(\xi, \eta, \zeta)$ as shown in Fig. 2.1 . The mesh is assumed stationary in the computation domain. The grid is only moving in the physical domain if moving grid is activated. The relation between the two coordinate systems is as the following,

$$\tau = t,\ \xi = \xi\,(t, x, y, z),\ \eta = \eta\,(t, x, y, z),\ \zeta = \zeta\,(t, x, y, z) \tag{2.50}$$

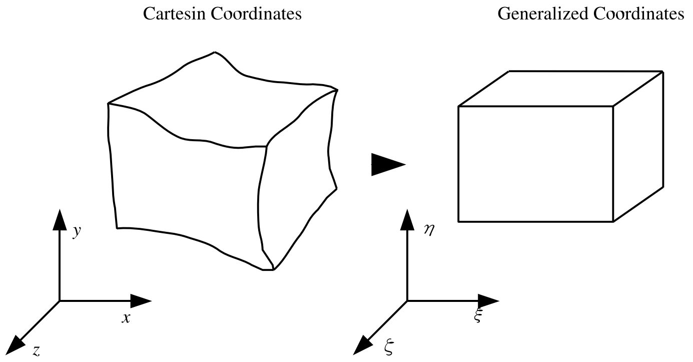


Figure 2.1: Coordinates transformation

The temporal and spatial derivatives in the physical domain are connected with those in the computational domains as,

$$\frac{\partial}{\partial t}=\frac{\partial}{\partial \tau}+\xi_t\frac{\partial}{\partial \xi}+\eta_t\frac{\partial}{\partial \eta}+\zeta_t\frac{\partial}{\zeta} \tag{2.51}$$

$$\frac{\partial}{\partial x_i}=\xi_{x_i}\frac{\partial}{\partial \xi}+\eta_{x_i}\frac{\partial}{\partial \eta}+\zeta_{x_i}\frac{\partial}{\partial \zeta} \tag{2.52}$$

where the index $i = 1, 2, 3$ stands for the direction of $x$, $y$ and $z$ respectively.

Using these relations, the governing equations are transformed from the physical domain to the computational domain as the following [115],

$$\frac{\partial \mathbf{U}'}{\partial t}+\frac{\partial \mathbf{E}'}{\partial \xi}+\frac{\partial \mathbf{F}'}{\partial \eta}+\frac{\partial \mathbf{G}'}{\partial \zeta}=\frac{\partial \mathbf{R}'}{\partial \xi}+\frac{\partial \mathbf{S}'}{\partial \eta}+\frac{\partial \mathbf{T}'}{\partial \zeta}+\mathbf{D}' \tag{2.53}$$

where

$$\mathbf{U}'=\frac{1}{J}\mathbf{U} \tag{2.54}$$

$$\mathbf{E}'=\frac{1}{J}\left(\xi_t\mathbf{U}+\mathbf{E}\xi_x+\mathbf{F}\xi_y+\mathbf{G}\xi_z\right) \tag{2.55}$$

$$\mathbf{F}'=\frac{1}{J}\left(\eta_t\mathbf{U}+\mathbf{E}\eta_x+\mathbf{F}\eta_y+\mathbf{G}\eta_z\right) \tag{2.56}$$

$$\mathbf{G}' = \frac{1}{J}\left(\zeta_t \mathbf{U} + \mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z\right) \tag{2.57}$$

$$\mathbf{R}' = \frac{1}{J}\left(\mathbf{R}\xi_x + \mathbf{S}\xi_y + \mathbf{T}\xi_z\right) \tag{2.58}$$

$$\mathbf{S}' = \frac{1}{J}\left(\mathbf{R}\eta_x + \mathbf{S}\eta_y + \mathbf{T}\eta_z\right) \tag{2.59}$$

$$\mathbf{T}' = \frac{1}{J}\left(\mathbf{R}\zeta_x + \mathbf{S}\zeta_y + \mathbf{T}\zeta_z\right) \tag{2.60}$$

$$\mathbf{D}' = \frac{1}{J}\mathbf{D} \tag{2.61}$$

In the above equations,

$$\xi_x = J\left(y_\eta z_\zeta - y_\zeta z_\eta\right),\ \xi_y = J\left(x_\zeta z_\eta - x_\eta z_\zeta\right),\ \xi_z = J\left(x_\eta y_\zeta - x_\zeta y_\eta\right) \tag{2.62}$$

$$\eta_x = J\left(y_\zeta z_\xi - y_\xi z_\zeta\right),\ \eta_y = J\left(x_\xi z_\zeta - x_\zeta z_\xi\right),\ \eta_z = J\left(x_\zeta y_\xi - x_\xi y_\zeta\right) \tag{2.63}$$

$$\zeta_x = J\left(y_\xi z_\eta - y_\eta z_\xi\right),\ \zeta_y = J\left(x_\eta z_\xi - x_\xi z_\eta\right),\ \zeta_z = J\left(x_\xi y_\eta - x_\eta y_\xi\right) \tag{2.64}$$

$$\begin{aligned}
\xi_t &= -\left(x_\tau \xi_x + y_\tau \xi_y + z_\tau \xi_z\right) \\
\eta_t &= -\left(x_\tau \eta_x + y_\tau \eta_y + z_\tau \eta_z\right) \\
\zeta_t &= -\left(x_\tau \zeta_x + y_\tau \zeta_y + z_\tau \zeta_z\right)
\end{aligned} \tag{2.65}$$

$J$ is the coordinates transformation Jacobian,

$$\begin{aligned}
J = \frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} &= \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix} \\
&= \frac{1}{x_\zeta\left(y_\eta z_\zeta - y_\zeta z_\eta\right) - x_\eta\left(y_\xi z_\zeta - y_\zeta z_\xi\right) + x_\zeta\left(y_\xi z_\eta - y_\eta z_\xi\right)}
\end{aligned} \tag{2.66}$$

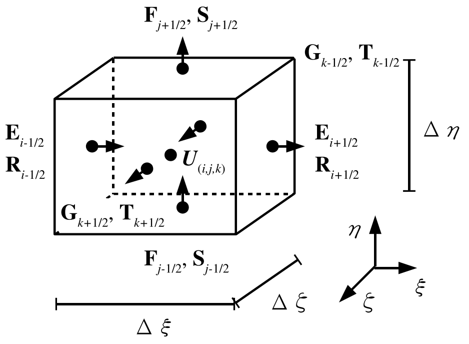

$\mathbf{F}_{j+1/2}$, $\mathbf{S}_{j+1/2}$
$\mathbf{G}_{k-1/2}$, $\mathbf{T}_{k-1/2}$
$\mathbf{E}_{i-1/2}$
$\mathbf{R}_{i-1/2}$
$U_{(i,j,k)}$
$\mathbf{E}_{i+1/2}$
$\mathbf{R}_{i+1/2}$
$\Delta\,\eta$
$\mathbf{G}_{k+1/2}$, $\mathbf{T}_{k+1/2}$
$\mathbf{F}_{j-1/2}$, $\mathbf{S}_{j-1/2}$
$\eta$
$\Delta\,\zeta$
$\zeta$
$\xi$
$\Delta\,\xi$

Figure 2.2: Control volume

## 2.5 Governing Equations in Integral Forms

In the current work, a finite volume method is used to discretize the governing equations. The governing equations are further transformed from the differential forms to their integral form.

As shown in Fig. 2.2 , integrate the Eqs. (2.53) in the generalized coordinates over a control volume $\Delta\Omega$ , which has a volume of $\Delta\xi\Delta\eta\Delta\zeta$.

$$\int_{\Delta\Omega}\left[\frac{\partial \mathbf{U}'}{\partial t}+\frac{\partial \mathbf{E}'}{\partial \xi}+\frac{\partial \mathbf{F}'}{\partial \eta}+\frac{\partial \mathbf{G}'}{\partial \zeta}\right]\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta=\int_{\Delta\Omega}\left[\frac{\partial \mathbf{R}'}{\partial \xi}+\frac{\partial \mathbf{S}'}{\partial \eta}+\frac{\partial \mathbf{T}'}{\partial \zeta}+\mathbf{D}'\right]\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \tag{2.67}$$

The control volume $\Delta\Omega$ defined in the computational domain has a corresponding volume of $\Delta V=\Delta x\Delta y\Delta z$ in the physical domain.

Let $\Delta\xi=\Delta\eta=\Delta\zeta=1$,

$$\Delta V=J\left(\frac{x,y,z}{\xi,\eta,\zeta}\right)\Delta\xi\Delta\eta\Delta\zeta=\Delta\xi\Delta\eta\Delta\zeta/J\left(\frac{\xi,\eta,\zeta}{x,y,z}\right)=\frac{\Delta\Omega}{J}=\frac{1}{J} \tag{2.68}$$

Finally, the governing equations are discretized into an integral form as,

$$\frac{\Delta V}{\Delta t}\left(\mathbf{U}^{n+1}-\mathbf{U}^{n}\right)+\left(\mathbf{E}_{i+\frac{1}{2}}-\mathbf{E}_{i-\frac{1}{2}}\right)+\left(\mathbf{F}_{j+\frac{1}{2}}-\mathbf{F}_{j-\frac{1}{2}}\right)+\left(\mathbf{G}_{k+\frac{1}{2}}-\mathbf{G}_{k-\frac{1}{2}}\right)$$
$$=\left(\mathbf{R}_{i+\frac{1}{2}}-\mathbf{R}_{i-\frac{1}{2}}\right)+\left(\mathbf{S}_{j+\frac{1}{2}}-\mathbf{S}_{j-\frac{1}{2}}\right)+\left(\mathbf{T}_{k+\frac{1}{2}}-\mathbf{T}_{k-\frac{1}{2}}\right)+\mathbf{D}\cdot\Delta V \qquad (2.69)$$

where $i\pm\frac{1}{2}$, $j\pm\frac{1}{2}$, $k\pm\frac{1}{2}$ refer to the right and left inter faces of the control volume at $(i,j,k)$ in $\xi$, $\eta$ and $\zeta$ directions. $n$ and $n+1$ are two sequential time steps. **E**, **F**, **G** and **R**, **S**, **T** are the inviscid and viscid fluxes on the control volume inter surfaces in each direction. They have the same forms as $\mathbf{E}'$, $\mathbf{F}'$, $\mathbf{G}'$, $\mathbf{R}'$, $\mathbf{S}'$, $\mathbf{T}'$ when $\Delta\xi=\Delta\eta=\Delta\zeta=1$.

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho e\end{bmatrix} \qquad (2.70)$$

$$\mathbf{E}=\begin{bmatrix}\rho U\\ \rho uU+l_x p\\ \rho vU+l_y p\\ \rho wU+l_z p\\ (\rho e+p)U\end{bmatrix},\ \mathbf{F}=\begin{bmatrix}\rho V\\ \rho uV+m_x p\\ \rho vV+m_y p\\ \rho wV+m_z p\\ (\rho e+p)V\end{bmatrix},\ \mathbf{G}=\begin{bmatrix}\rho W\\ \rho uW+n_x p\\ \rho vW+n_y p\\ \rho wW+n_z p\\ (\rho e+p)W\end{bmatrix} \qquad (2.71)$$

$$\mathbf{R}=\begin{bmatrix}0\\ l_k\tau_{xk}\\ l_k\tau_{yk}\\ l_k\tau_{zk}\\ l_k\beta_k\end{bmatrix},\ \mathbf{S}=\begin{bmatrix}0\\ m_k\tau_{xk}\\ m_k\tau_{yk}\\ m_k\tau_{zk}\\ m_k\beta_k\end{bmatrix},\ \mathbf{T}=\begin{bmatrix}0\\ n_k\tau_{xk}\\ n_k\tau_{yk}\\ n_k\tau_{zk}\\ n_k\beta_k\end{bmatrix} \qquad (2.72)$$

where

$$\beta_k = u_i \tau_{ki} - q_k \tag{2.73}$$

In equations above, $U$, $V$ and $W$ are the contravariant velocities in $\xi$, $\eta$ and $\zeta$ directions,

$$U = l_t + l_x u + l_y v + l_z w \tag{2.74}$$

$$V = m_t + m_x u + m_y v + m_z w \tag{2.75}$$

$$W = n_t + n_x u + n_y v + n_z w \tag{2.76}$$

where **l**, **m**, **n** are the normal vectors on $\xi, \eta, \zeta$ surfaces with their magnitudes equal to the elemental surface area and pointing to the directions of increasing $\xi, \eta, \zeta$.

$$\mathbf{l} = \frac{\nabla \xi}{J} \mathrm{d}\eta \mathrm{d}\zeta, \; \mathbf{m} = \frac{\nabla \eta}{J} \mathrm{d}\xi \mathrm{d}\zeta, \; \mathbf{n} = \frac{\nabla \zeta}{J} \mathrm{d}\xi \mathrm{d}\eta \tag{2.77}$$

$l_t$, $m_t$, $n_t$ stand for the grid moving velocities and are defined as

$$l_t = \frac{\xi_t}{J} \mathrm{d}\eta \mathrm{d}\zeta, \; m_t = \frac{\eta_t}{J} \mathrm{d}\xi \mathrm{d}\zeta, \; n_t = \frac{\zeta_t}{J} \mathrm{d}\xi \mathrm{d}\eta \tag{2.78}$$

When the grid is stationary, $l_t = m_t = z_t = 0$.

Since $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ in the current discretization, Eqs.(2.77) and (2.78) are written as the following in the solver,

$$\mathbf{l} = \frac{\nabla \xi}{J}, \; \mathbf{m} = \frac{\nabla \eta}{J}, \; \mathbf{n} = \frac{\nabla \zeta}{J} \tag{2.79}$$

$$l_t = \frac{\xi_t}{J}, \; m_t = \frac{\eta_t}{J}, \; n_t = \frac{\zeta_t}{J} \tag{2.80}$$

It is shown from Eq. (2.69) that the governing equations for stationary grid system and moving grid system have the same mathematical formulation. It is straightforward to transfer the code from a stationary grid system to a moving grid system by simply including the grid moving velocity in the contravariant velocity components.

## 2.6 Geometric Conservation Law

It was pointed out by Thomas et al. [116] that due to the mixed temporal and spatial derivatives after discretization, an additional term appears, which theoretically equals to zero but numerically still remains. Consequently numerical error could be introduced in the discretized form of the equations of the flow motion if this term is neglected. In order to reduce or avoid this error, the geometric conservation law needs to be enforced. In other words, the following additional term should be added to the right-hand side of the governing equations as a source term as shown in Eqs. (2.44) and (2.69),

$$\mathbf{D} = \mathbf{U}\left[\frac{\partial J^{-1}}{\partial t} + \left(\frac{\xi_t}{J}\right)_\xi + \left(\frac{\eta_t}{J}\right)_\eta + \left(\frac{\zeta_t}{J}\right)_\zeta\right] \tag{2.81}$$

As has been observed, the overall performance of this numerical supplement is beneficial with very little CPU time cost.

## 2.7 Baldwin-Lomax Turbulence Model

To achieve high CPU efficiency, the Baldwin-Lomax turbulence model is employed in the current work for its simplicity and robustness. The Baldwin-Lomax model has been successfully applied to 2D and 3D subsonic, transonic steady and unsteady flows [14, 63, 117, 118].

The classic algebraic Baldwin-Lomax model [56] is a two-layer model.

At inner layer,

$$\mu_{t,i} = \rho l^2 |\omega| \tag{2.82}$$

where the mixing length is determined by,

$$l = ky\left[1-\exp\left(-\frac{y^+}{A^+}\right)\right] \tag{2.83}$$

and $\omega$ is the local vorticity, $y$ and $y^+$ are the dimensional and dimensionless distance to the wall.

At the outer layer,

$$\mu_{t,o} = KC_{\text{cp}}\rho F_{\text{wake}}F_{\text{kleb}} \tag{2.84}$$

$$F_{\text{wake}} = \min\left(y_{\text{max}}F_{\text{max}},\, C_{\text{wake}}y_{\text{max}}u_{\text{diff}}^2/F_{\text{max}}\right) \tag{2.85}$$

$$F_{\text{kleb}} = \left[1+5.5\left(\frac{C_{\text{kleb}}y}{y_{\text{max}}}\right)^6\right]^{-1} \tag{2.86}$$

In the above formulations, constants take the following values, $k$=0.4, $A^+$=26, $C_{\text{wake}}$=0.25, $C_{\text{kleb}}$=0.3, $C_{\text{cp}}$=1.6 and $K$=0.0168.

The quantities $u_{\text{diff}}$, $F_{\text{max}}$ and $y_{\text{max}}$ are determined by the velocity profile following a line normal to the wall. $F_{\text{max}}$ and $y_{\text{max}}$ are the maximum value and the corresponding distance of function $F_y$,

$$F_y = y\,|\omega|\left[1-\exp\left(-\frac{y^+}{A^+}\right)\right] \tag{2.87}$$

$$u_{\text{diff}} = \left(\sqrt{u^2+v^2+w^2}\right)_{\text{max}} - \left(\sqrt{u^2+v^2+w^2}\right)_{\text{min}} \tag{2.88}$$

Finally the turbulent viscosity distribution across the boundary layer is determined by

$$\mu_t = \min\left(\mu_{t,i},\, \mu_{t,o}\right) \tag{2.89}$$

that is, $\mu_t$ is set to $\mu_{t,i}$ from the wall to a location where $\mu_{t,i}$ exceeds $\mu_{t,o}$ and $\mu_t$ is set to $\mu_{t,o}$ beyond this location.

In the present work, the Baldwin-Lomax turbulent model is implemented for 3D flows and wake flows as the following. In 3D computation, there could be more than one surface treated as walls in the computation domain. At each grid point, the turbulent viscosity $\mu_t$ is first computed relative to each wall respectively using the above method. Then the distances to each wall are compared and the turbulent viscosity value based on the closest wall is chosen for the turbulent viscosity on the grid point. This method is applied in both the interior region and the corner regions of the computation domain. In the wake region, the exponential part is set to zero in Eq. (2.83) and Eq. (2.87). The second part of Eq. (2.88) is set to be zero outside of the wake region.

# Chapter 3

# The Upwind Schemes

The new efficient and accurate upwind schemes, Zha CUSP schemes, are developed to compute the inviscid inter surface fluxes on the control volume. Two other popular upwind schemes, the Roe scheme is also presented in this chapter for comparison. These schemes are applied in the present work for numerical simulation. The Zha CUSP schemes belong to E-CUSP (Convective upwind and split pressure) schemes. They use scalar dissipation, which is more efficient in CPU usage. The Roe scheme uses matrix dissipation. The Zha CUSP schemes strictly follow the characteristic directions of the disturbance propagation, which may improve the stability and robustness of computation.

## 3.1 Characteristics Basis

The Zha CUSP schemes and the Roe scheme covered in this chapter are both based on the characteristics of the hyperbolic Euler equations.

Take the $\xi$ direction inviscid flux **E** in moving grid system for example. The Jacobian matrix of **E** in Eq. (2.69) is defined as,

$$A = \frac{\partial \mathbf{E}}{\partial \mathbf{U}} =$$
$$\begin{pmatrix} 0 & l_x & l_y & l_z & 0 \\ -uU + (\gamma - 1)\, l_x q & U + (2-\gamma)\, l_x u & l_y u - (\gamma - 1)\, l_x v & l_z u - (\gamma - 1)\, l_x w & (\gamma - 1) l_x \\ -vU + (\gamma - 1)\, l_y q & l_x v - (\gamma - 1)\, l_y u & U + (2-\gamma)\, l_y v & l_z v - (\gamma - 1)\, l_y w & (\gamma - 1) l_y \\ -wU + (\gamma - 1)\, l_z q & l_x w - (\gamma - 1)\, l_z u & l_y w - (\gamma - 1)\, l_z v & U + (2-\gamma)\, l_z w & (\gamma - 1) l_z \\ a_{51} & a_{52} & a_{53} & a_{54} & \gamma U \end{pmatrix} \tag{3.1}$$

where

$$a_{51} = -\gamma U e + 2qU\,(\gamma - 1) \tag{3.2}$$

$$a_{52} = -\,(\gamma - 1)\, uU + l_x \left[\gamma e - (\gamma - 1)\, q\right] \tag{3.3}$$

$$a_{53} = -\,(\gamma - 1)\, vU + l_y \left[\gamma e - (\gamma - 1)\, q\right] \tag{3.4}$$

$$a_{54} = -\,(\gamma - 1)\, wU + l_z \left[\gamma e - (\gamma - 1)\, q\right] \tag{3.5}$$

$$q = \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{3.6}$$

The eigenvalues of matrix $A$ are

$$\lambda_1 = U + C \tag{3.7}$$

$$\lambda_2 = U - C \tag{3.8}$$

$$\lambda_{3,4,5} = U \tag{3.9}$$

where $C$ is the contravariant speed of sound and is determined by the speed of sound $c$ by ,

$$C = c\sqrt{l_x^2 + l_y^2 + l_z^2} \tag{3.10}$$

Then $A$ can be split into

$$A = R\Lambda L \tag{3.11}$$

where $\Lambda$ is the eigenvalue matrix,

$$\Lambda = \begin{bmatrix} U+C & & & & \\ & U-C & & 0 & \\ & & U & & \\ & 0 & & U & \\ & & & & U \end{bmatrix} \tag{3.12}$$

$R$ and $L$ are the right and left eigenvector matrices,

$$L = \begin{pmatrix} q - \dfrac{c\hat{U}}{\gamma-1} & -u + \dfrac{c\hat{l}_x}{\gamma-1} & -v + \dfrac{c\hat{l}_y}{\gamma-1} & -w + \dfrac{c\hat{l}_z}{\gamma-1} & 1 \\ q + \dfrac{c\hat{U}}{\gamma-1} & -u - \dfrac{c\hat{l}_x}{\gamma-1} & -v - \dfrac{c\hat{l}_y}{\gamma-1} & -w - \dfrac{c\hat{l}_z}{\gamma-1} & 1 \\ -\hat{V} & \hat{m}_x & \hat{m}_y & \hat{m}_z & 0 \\ -\hat{W} & \hat{n}_x & \hat{n}_y & \hat{n}_z & 0 \\ q-h & -u & -v & -w & 1 \end{pmatrix} \tag{3.13}$$

$$R = \begin{pmatrix} \dfrac{1}{2h} & \dfrac{1}{2h} & 0 & 0 & -\dfrac{1}{h} \\ \dfrac{u+c\hat{l}_x}{2h} & \dfrac{u-c\hat{l}_x}{2h} & \hat{m}_x & \hat{n}_x & -\dfrac{u}{h} \\ \dfrac{v+c\hat{l}_y}{2h} & \dfrac{v-c\hat{l}_y}{2h} & \hat{m}_y & \hat{n}_y & -\dfrac{v}{h} \\ \dfrac{w+c\hat{l}_z}{2h} & \dfrac{w-c\hat{l}_z}{2h} & \hat{m}_z & \hat{n}_z & -\dfrac{w}{h} \\ \dfrac{c\hat{U}+\gamma e-(\gamma-1)q}{2h} & \dfrac{-c\hat{U}+\gamma e-(\gamma-1)q}{2h} & \hat{V} & \hat{W} & -\dfrac{q}{h} \end{pmatrix} \tag{3.14}$$

where the static enthalpy $h$ is calculated as,

$$h = \frac{c^2}{\gamma - 1} = \frac{T}{(\gamma - 1) M_\infty^2} \tag{3.15}$$

$\hat{\mathbf{l}}$ is the unit vector normal to $\xi$ surface pointing to the direction where $\xi$ increases,

$$\hat{\mathbf{l}} = \frac{\mathbf{l}}{|\mathbf{l}|} \tag{3.16}$$

$\hat{\mathbf{m}}, \hat{\mathbf{n}}$ and $\hat{\mathbf{l}}$ are mutually orthogonal unit vectors.

$\hat{U}$, $\hat{V}$ and $\hat{W}$ are determined by,

$$\hat{U} = \mathbf{V} \cdot \hat{\mathbf{l}} \tag{3.17}$$

$$\hat{V} = \mathbf{V} \cdot \hat{\mathbf{m}} \tag{3.18}$$

$$\hat{W} = \mathbf{V} \cdot \hat{\mathbf{n}} \tag{3.19}$$

## 3.2 CUSP Schemes

The CUSP schemes split the inviscid flux into a convective part and a pressure part. According to the splitting method, there are two types of CUSP schemes. The H-CUSP schemes have the total enthalpy in their convective part and the E-CUSP schemes have the total energy in their convective part. The H-CUSP schemes may have the advantages to better conserve the total enthalpy for steady state flows. However, from the characteristic theory point of view, the E-CUSP schemes are more consistent with the disturbance propagation directions and their stability and robustness may be improved.

Take $\xi$ direction as example, the E-CUSP scheme splits the inviscid flux $\mathbf{E}$ on interface $i \pm \frac{1}{2}$ into convective vector $\mathbf{E}^c$ and wave (pressure) vector $\mathbf{E}^p$ to represent the velocity and pressure wave characteristics [110].

Based on Eqs. (3.1), (3.11) and (3.12),

$$
\begin{aligned}
\mathbf{E} &= R\begin{bmatrix} U+C & & & & \\ & U-C & & 0 & \\ & & U & & \\ & 0 & & U & \\ & & & & U \end{bmatrix} L\cdot\mathbf{U} \\
&= R\begin{bmatrix} U & & & & \\ & U & & 0 & \\ & & U & & \\ & 0 & & U & \\ & & & & U \end{bmatrix} L\cdot\mathbf{U} + R\begin{bmatrix} C & & & & \\ & -C & & 0 & \\ & & 0 & & \\ & 0 & & 0 & \\ & & & & 0 \end{bmatrix} L\cdot\mathbf{U} \\
&= \mathbf{E}^c + \mathbf{E}^p
\end{aligned}
\tag{3.20}
$$

where $U$ is the contravariant velocity in $\xi$ direction.

The convective and pressure vectors of the E-CUSP schemes are then split as the following,

$$
\mathbf{E}^c = \begin{bmatrix} \rho U \\ \rho u U \\ \rho v U \\ \rho w U \\ \rho e U \end{bmatrix}, \ \mathbf{E}^p = \begin{bmatrix} 0 \\ l_x p \\ l_y p \\ l_z p \\ p\bar{U} \end{bmatrix} \tag{3.21}
$$

It should be noted that $U$ in the convective vector is the contravariant velocity computed using the main flow velocity relative to the grid moving velocity as Eq. (2.74). In the pressure vector, $\bar{U}$ is the contravariant computed using the absolute main flow velocity only,

$$\begin{aligned}\bar{U} &= U - l_t \\ &= ul_x + vl_y + wl_x \end{aligned} \tag{3.22}$$

For H-CUSP schemes, the inviscid flux is split as the following,

$$\mathbf{E}^c = \begin{bmatrix} \rho U \\ \rho uU \\ \rho vU \\ \rho wU \\ \rho HU \end{bmatrix}, \mathbf{E}^p = \begin{bmatrix} 0 \\ l_x p \\ l_y p \\ l_z p \\ 0 \end{bmatrix} \tag{3.23}$$

where the total enthalpy $H$ instead of total energy $e$ is included in its convective vector $\mathbf{E}^c$. Liou's AUSM$^+$ scheme [104] belongs to H-CUSP schemes.

The total enthalpy $H$ is defined as,

$$H = e + \frac{p}{\rho} \tag{3.24}$$

The H-CUSP schemes are not fully consistent with characteristics direction analysis [110]. The Zha CUSP schemes belong to the E-CUSP schemes. They split the inviscid surface flux following the characteristics directions.

## 3.3 Zha CUSP Schemes

In this section, two Zha CUSP schemes are presented. Modification is made in the Zha CUSP2 scheme to remove the temperature oscillation near the wall in the original Zha CUSP scheme when the grid is skewed.

### 3.3.1 Zha CUSP scheme

The Zha CUSP scheme is an E-CUSP scheme and is based on the convective and pressure vectors given in Eqs. (3.20) and (3.21). In subsonic regime, the convective vector $\mathbf{E}^c$ is hence treated in an upwind manner. The eigenvalues of $\mathbf{E}^p$ are $(C, -C, 0, 0, 0)$. So the pressure vector $\mathbf{E}^p$ is averaged with the weight of the eigenvalues $U \pm C$ from both the upwind and the downwind directions.

The interface flux $\mathbf{E}_{\frac{1}{2}}$ is evaluated as the following.

In subsonic regime, $|U|_{\frac{1}{2}} < C_{\frac{1}{2}}$,

$$\mathbf{E}_{\frac{1}{2}} = \frac{1}{2}\left[(\rho U)_{\frac{1}{2}}(\mathbf{q}_L^c + \mathbf{q}_R^c) - |\rho U|_{\frac{1}{2}}(\mathbf{q}_R^c - \mathbf{q}_L^c)\right] + \begin{pmatrix} 0 \\ \mathscr{P}^+ p l_x \\ \mathscr{P}^+ p l_y \\ \mathscr{P}^+ p l_z \\ \frac{1}{2}p\left[\bar{U} + \bar{C}_{\frac{1}{2}}\right] \end{pmatrix}_L + \begin{pmatrix} 0 \\ \mathscr{P}^- p l_x \\ \mathscr{P}^- p l_y \\ \mathscr{P}^- p l_z \\ \frac{1}{2}p\left[\bar{U} - \bar{C}_{\frac{1}{2}}\right] \end{pmatrix}_R \tag{3.25}$$

where,

$$\mathbf{q}^c = \begin{pmatrix} 1 \\ u \\ v \\ w \\ e \end{pmatrix} \tag{3.26}$$

The subscripts $L$ and $R$ represent the left and right hand sides of the interface. The interface speed of sound $C_{\frac{1}{2}}$ is computed as,

$$C_{\frac{1}{2}} = \frac{1}{2}(C_L + C_R) \tag{3.27}$$

where $C_L$ and $C_R$ are the speed of sound determined from the left and the right sides of the interface.

The contravariant speed of sound $\bar{C}$ in the pressure vector is consistent with $\bar{U}$. It is computed based on $C$ as the following,

$$\bar{C} = C - l_t \tag{3.28}$$

The use of $\bar{U}$ and $\bar{C}$ instead of $U$ and $C$ in the pressure vector to take into account of the grid speed so that the flux will transit from subsonic to supersonic smoothly. When the grid is stationary, $l_t = 0, \bar{C} = C, \bar{U} = U$.

The mass flux on the interface is introduced as the following,

$$(\rho U)_{\frac{1}{2}} = \left(\rho_L U_L^+ + \rho_R U_R^-\right) \tag{3.29}$$

where,

$$U_L^+ = C_{\frac{1}{2}} \left\{ \frac{M_L + |M_L|}{2} + \alpha_L \left[ \frac{1}{4}(M_L+1)^2 - \left( \frac{M_L + |M_L|}{2} \right) \right] \right\} \tag{3.30}$$

$$U_R^- = C_{\frac{1}{2}} \left\{ \frac{M_R - |M_R|}{2} + \alpha_R \left[ -\frac{1}{4}(M_R-1)^2 - \left( \frac{M_R - |M_R|}{2} \right) \right] \right\} \tag{3.31}$$

$$\alpha_L = \frac{2(p/\rho)_L}{(p/\rho)_L + (p/\rho)_R} \tag{3.32}$$

$$\alpha_R = \frac{2(p/\rho)_R}{(p/\rho)_L + (p/\rho)_R} \tag{3.33}$$

$$M_L = \frac{U_L}{C_{\frac{1}{2}}} \tag{3.34}$$

$$M_R = \frac{U_R}{C_{\frac{1}{2}}} \tag{3.35}$$

The coefficient $\mathscr{P}$ is defined as,

$$\mathscr{P}_L^+ = \frac{1}{4}(M_L+1)^2 (2 - M_L) + \alpha M_L \left(M_L^2 - 1\right)^2 \tag{3.36}$$

$$\mathscr{P}_R^- = \frac{1}{4}\left(M_R - 1\right)^2\left(2 + M_R\right) - \alpha M_R\left(M_R^2 - 1\right)^2 \tag{3.37}$$

where

$$\alpha = \frac{3}{16}$$

In supersonic regime, $|U|_{\frac{1}{2}} \geqslant C_{\frac{1}{2}}$, $\mathbf{E}_{\frac{1}{2}}$ is simply computed using upwind variables.

When $U_{\frac{1}{2}} > C_{\frac{1}{2}}$,

$$\mathbf{E}_{\frac{1}{2}} = \mathbf{E}_L \tag{3.38}$$

When $U_{\frac{1}{2}} < -C_{\frac{1}{2}}$,

$$\mathbf{E}_{\frac{1}{2}} = \mathbf{E}_R \tag{3.39}$$

### 3.3.2 Zha CUSP2 scheme

The above formulations are for the original Zha CUSP scheme, which can capture sharp shock profile and exact contact surface with low diffusion [110]. However, the scheme is found to have temperature oscillations near the wall when the grid is skewed. The Zha CUSP scheme is hence modified to the following Zha CUSP2 scheme [112].

In Zha CUSP2 scheme, the total enthalpy instead of the static pressure is used to compute the numerical dissipation coefficients $\alpha_L$ and $\alpha_R$ for the energy equation,

$$\alpha_L = \frac{2(H/\rho)_L}{(H/\rho)_L + (H/\rho)_R} \tag{3.40}$$

$$\alpha_R = \frac{2(H/\rho)_R}{(H/\rho)_L + (H/\rho)_R} \tag{3.41}$$

It needs to be emphasized that, when computing the fluxes of continuity and momentum equations, the formulations of $\alpha_L$ and $\alpha_R$ given in Eq. (3.32) and Eq. (3.33) must be used. Eq.(3.40) and (3.41) are only for the energy equation.

Compared with the results of the original Zha CUSP scheme, the temperature oscillations are removed by using Eq.(3.40) and (3.41) and the wall temperature is more precisely predicted by this modified scheme even if a coarse grid is used [112].

## 3.4 Roe Scheme

Roe scheme suggested his approximate Riemann solver in 1D form in 1981 [95]. The scheme is extended to 3D as the following for moving grid systems [119].

Take $\xi$ direction for example, ,

$$\mathbf{E}_{i+\frac{1}{2}} = \frac{1}{2}\left[\mathbf{E}_L + \mathbf{E}_R + \mathbf{U}_L l_{t,L} + \mathbf{U}_R l_{t,R} + \tilde{R}|\tilde{\Lambda}|\tilde{L}(\mathbf{U}_L - \mathbf{U}_R)\right] \tag{3.42}$$

where $i+\frac{1}{2}$ stands for the right surface of cell $i$, subscripts $L$ and $R$ indicate variables computed from the left and right hand sides of this surface. The tilde stands for the Roe average. $l_{t,L}$ and $l_{t,R}$ are calculated based on the reconstructed grid velocity $\xi_{t,L}$ and $\xi_{t,R}$,

$$l_{t,L} = \frac{\xi_{t,L}}{J}\mathrm{d}\eta\mathrm{d}\zeta,\ l_R = \frac{\xi_{t,R}}{J}\mathrm{d}\eta\mathrm{d}\zeta \tag{3.43}$$

Compared with Eq. (3.11), $\tilde{R}$, $\tilde{\Lambda}$ and $\tilde{L}$ have the same formulation with $R$, $\Lambda$ and $L$, but the variables are replaced by their corresponding Roe-averaged counterparts. For example, $\left|\tilde{\Lambda}\right|$ is defined as,

$$|\tilde{\Lambda}\models \begin{pmatrix} |\tilde{U}+\tilde{C}| & & & & \\ & |\tilde{U}-\tilde{C}| & & 0 & \\ & & |\tilde{U}| & & \\ & 0 & & |\tilde{U}| & \\ & & & & |\tilde{U}| \end{pmatrix} \tag{3.44}$$

The Roe-averaged variables are evaluated at interface $i+\frac{1}{2}$ as,

$$\tilde{\rho} = \sqrt{\rho_i \rho_{i+1}} \tag{3.45}$$

$$\tilde{u} = \frac{u_i\sqrt{\rho_i} + u_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{3.46}$$

$$\tilde{v} = \frac{v_i\sqrt{\rho_i} + v_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{3.47}$$

$$\tilde{w} = \frac{w_i\sqrt{\rho_i} + w_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{3.48}$$

$$\tilde{H} = \frac{H_i\sqrt{\rho_i} + H_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{3.49}$$

The Roe average of the grid moving velocity is similar to the flow variable average,

$$\tilde{\xi_t} = \frac{\xi_{t,i}\sqrt{\rho_i} + \xi_{t,i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{3.50}$$

It can be proved that the eigenvector matrix $L$ and $R$ are exactly the same as the one without moving grid. The only difference between the moving grid system and the stationary grid system is that, for the moving grid system, the contravariant velocity in the eigenvalues contains the grid velocity as given in Eq. (2.74). It is hence straightforward to extend the code from a stationary grid system to the moving grid system using Roe scheme.

## 3.5 van Leer Scheme

The van Leer scheme is proposed in 1982 [97]. When written in Cartesian coordinates, the $x$ direction inviscid flux $\mathbf{E}$ at the inter surface is split as,

$$\mathbf{E} = \mathbf{E}^+ + \mathbf{E}^- \tag{3.51}$$

In supersonic region, $|M_x| > 1$,

$$\mathbf{E} = \mathbf{E}^+,\ \mathbf{E}^- = 0 \text{ when } M_x > 1 \tag{3.52}$$

$$\mathbf{E} = \mathbf{E}^-,\ \mathbf{E}^+ = 0 \text{ when } M_x < 1 \tag{3.53}$$

In subsonic region, $|M_x| < 1$,

$$\mathbf{E}^{\pm} = \begin{bmatrix} f_m^{\pm} \\ f_m^{\pm}\dfrac{(\gamma-1)\,u \pm 2a}{\gamma} \\ f_m^{\pm} v \\ f_m^{\pm} w \\ f_m^{\pm}\left\{\dfrac{\left[(\gamma-1)\,u \pm 2a\right]^2}{2\,(\gamma^2-1)} + \frac{1}{2}\left(v^2+w^2\right)\right\} \end{bmatrix} \tag{3.54}$$

where

$$f_m^{\pm} = \pm\rho a \left[\frac{1}{2}\,(M_x \pm 1)\right]^2 \tag{3.55}$$

$M_x$ is the local Mach number based on $u$.

The van Leer scheme is transformed to generalized coordinates in reference [92].

## 3.6 Numerical Dissipation

The low numerical dissipation at stagnation is important to accurately resolve wall boundary layers. As shown in Eq. (3.23), the H-CUSP scheme has all the numerical dissipation terms vanishing when the velocity approaches zero, which hence yields low numerical dissipation for wall boundary layers. For the Zha CUSP schemes, as shown in Eq. (3.25), almost all of the numerical dissipation terms vanish with the velocity approaching zero, except one term in the energy equation due to the pressure splitting.

Assuming $U = 0$, the numerical dissipation vector of the Zha CUSP schemes at stagnation is:

$$\mathbf{D}_{\text{Zha-CUSP}} = -\frac{C_{\frac{1}{2}}}{2}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \delta p \end{pmatrix} \tag{3.56}$$

where

$$\delta p = p_R - p_L \tag{3.57}$$

The numerical dissipation of the Roe scheme at stagnation is:

$$\mathbf{D}_{\text{Roe}} = -\frac{\tilde{C}_{\frac{1}{2}}}{2(\gamma-1)}\begin{pmatrix} (\gamma-1)/\tilde{C}^2_{\frac{1}{2}}\delta p \\ 0 \\ 0 \\ 0 \\ \delta p \end{pmatrix} \tag{3.58}$$

When Eqs. (3.56) and (3.58) are compared, it can be seen that the numerical dissipation of the Zha CUSP schemes for the continuity equation vanishes at $U = 0$ while the Roe scheme has the non-vanishing dissipation. For the energy equation, the two schemes have equivalent dissipation. For perfect gas with the $\gamma = 1.4$, the coefficient of the Roe scheme energy dissipation term is 2.5 times larger than that of the Zha CUSP schemes.

In conclusion, even though there is one non-vanishing numerical dissipation term in the energy equation for the Zha CUSP schemes, the overall numerical dissipation of the Zha CUSP schemes is not greater than that of the Roe scheme. The Roe scheme is proved to be accurate to resolve wall boundary layers [96]. It is hence expected that the Zha CUSP schemes should also have sufficiently low dissipation to accurately resolve wall boundary layers. This is indeed the case shown by the numerical experiment for a flat plate boundary layer in later chapters.

# Chapter 4

# Numerical Methods

In this chapter, the numerical methods used to solve the three dimensional compressible time-dependent Navier-Stokes equations in the current work are described in details. The governing equations are discretized using a control volume method. The discretized linear equations are solved by the implicit line Gauss-Seidel iteration method. Third order accuracy is achieved on the spatial differencing for inviscid fluxes by the MUSCL differencing approach. The second order central differencing is used for viscous fluxes. The final spatial accuracy is second order. The time marching is made the second order of accuracy with the dual time stepping scheme. The local time stepping technique is used to speed up the convergence. The solver is made parallel by implementing the message passing interface (MPI) protocol. Finally the solver is able to study the fluid-structural interaction problems by applying the moving grid system.

## 4.1 Implicit Discretization

In the current work, the finite volume discretization is obtained with a cell-center method. The meshes for the computation domain are established using algebraic method or partial differential equation method. Then the grid points are placed in the geometric centers of these mesh cells. The flow variables are defined and computed at these cell center points.

The current work only makes use of structured quadrilateral control volumes or cells. The conservation laws are applied on each cell and the governing equations are discretized in the integral forms as Eq. (2.69) on page 32. The inviscid and viscous fluxes are evaluated on the cell surfaces. The temporal term and the source term are evaluated at the cell center for each cell.

To implement implicit time marching on the integral equations, Eq. (2.69) is first rewritten into the following form.

$$\begin{aligned}\frac{\Delta V}{\Delta t}\left(\mathbf{U}^{n+1}-\mathbf{U}^{n}\right)+\left(\mathbf{E}_{i+\frac{1}{2}}-\mathbf{E}_{i-\frac{1}{2}}\right)^{n+1}+\left(\mathbf{F}_{j+\frac{1}{2}}-\mathbf{F}_{j-\frac{1}{2}}\right)^{n+1}+\left(\mathbf{G}_{k+\frac{1}{2}}-\mathbf{G}_{k-\frac{1}{2}}\right)^{n+1}\\ =\left(\mathbf{R}_{i+\frac{1}{2}}-\mathbf{R}_{i-\frac{1}{2}}\right)^{n+1}+\left(\mathbf{S}_{j+\frac{1}{2}}-\mathbf{S}_{j-\frac{1}{2}}\right)^{n+1}+\left(\mathbf{T}_{k+\frac{1}{2}}-\mathbf{T}_{k-\frac{1}{2}}\right)^{n+1}+\mathbf{D}^{n+1}\cdot\Delta V \quad (4.1)\end{aligned}$$

where $n$ and $n+1$ are two sequential time levels, which have a time interval of $\Delta t$.

The surface fluxes **E**, **F**, **G** and **R**, **S**, **T** at the control volume left $(-\frac{1}{2})$ and right $(+\frac{1}{2})$ inter surface are computed based on the flow variables at the current cell and its neighboring cells. The inviscid fluxes are evaluated following the characteristic directions by using appropriate upwind schemes. In the present work, the employed upwind schemes include the popular Roe scheme, van Leer scheme and the new low diffusion upwind scheme, Zha CUSP scheme, which are given Chapter 3.

Take $\xi$ direction as the example, the inviscid flux on control volume inter surface $i+\frac{1}{2}$ is computed with its left ($L$) and right ($R$) side fluxes. The formulation of the inter surface flux differs between the Roe scheme and the van Leer scheme.

In the Roe scheme,

$$\mathbf{E}_{i+\frac{1}{2}}=\frac{1}{2}\left[\mathbf{E}_L+\mathbf{E}_R+\tilde{A}\left(\mathbf{U}_L-\mathbf{U}_R\right)\right]_{i+\frac{1}{2}} \quad (4.2)$$

Since the flux **E** is a first degree homogeneous function of conservative variable vector **U**, the surface inviscid fluxes $\mathbf{E}_L$ and $\mathbf{E}_R$ on time level $n$ and $n+1$ can be expressed as,

$$\begin{aligned}\mathbf{E}_L^{n+1} &= \mathbf{E}_L^n + \left(\frac{\partial \mathbf{E}}{\partial \mathbf{U}}\right)_L^n \Delta\mathbf{U}_L^{n+1} \\ &= \mathbf{E}_L^n + A_{i+\frac{1}{2}}^L \Delta\mathbf{U}_L^{n+1} \qquad (4.3)\end{aligned}$$

$$\begin{aligned}\mathbf{E}_R^{n+1} &= \mathbf{E}_R^n + \left(\frac{\partial \mathbf{E}}{\partial \mathbf{U}}\right)_R^n \Delta\mathbf{U}_R^{n+1} \\ &= \mathbf{E}_R^n + A_{i+\frac{1}{2}}^R \Delta\mathbf{U}_R^{n+1} \qquad (4.4)\end{aligned}$$

The Jacobian matrices are evaluated at time level $n$.

$$\tilde{A}\left(\mathbf{U}_L - \mathbf{U}_R\right)^{n+1} = \tilde{A}\left(\mathbf{U}_L - \mathbf{U}_R\right)^n + \widetilde{A}^n\left(\Delta\mathbf{U}_L^{n+1} - \Delta\mathbf{U}_R^{n+1}\right) \qquad (4.5)$$

Use first order accuracy,

$$\Delta\mathbf{U}_L^{n+1} = \Delta\mathbf{U}_i^{n+1},\ \Delta\mathbf{U}_R^{n+1} = \Delta\mathbf{U}_{i+1}^{n+1} \qquad (4.6)$$

Therefore,

$$\begin{aligned}\mathbf{E}_{i+\frac{1}{2}}^{n+1} - \mathbf{E}_{i-\frac{1}{2}}^{n+1} &= \left(\mathbf{E}_{i+\frac{1}{2}}^n - \mathbf{E}_{i-\frac{1}{2}}^n\right) + \frac{1}{2}\left(A_{i+\frac{1}{2}}^R - \tilde{A}_{i+\frac{1}{2}}\right)\Delta\mathbf{U}_{i+1}^{n+1} + \frac{1}{2}\left(A_{i+\frac{1}{2}}^L + \tilde{A}_{i+\frac{1}{2}}\right)\Delta\mathbf{U}_i^{n+1} \\ &\quad - \frac{1}{2}\left(A_{i-\frac{1}{2}}^R - \tilde{A}_{i-\frac{1}{2}}\right)\Delta\mathbf{U}_i^{n+1} - \frac{1}{2}\left(A_{i-\frac{1}{2}}^R + \tilde{A}_{i-\frac{1}{2}}\right)\Delta\mathbf{U}_{i-1}^{n+1} \\ &= \left(\mathbf{E}_{i+\frac{1}{2}}^n - \mathbf{E}_{i-\frac{1}{2}}^n\right) + \hat{A}_{i+\frac{1}{2}}^R\Delta\mathbf{U}_{i+1}^{n+1} + \hat{A}_{i+\frac{1}{2}}^L\Delta\mathbf{U}_i^{n+1} - \hat{A}_{i-\frac{1}{2}}^R\Delta\mathbf{U}_i^{n+1} - \hat{A}_{i-\frac{1}{2}}^L\Delta\mathbf{U}_{i-1}^{n+1} \qquad (4.7)\end{aligned}$$

where

$$\hat{A}_{i\pm\frac{1}{2}}^R = \frac{1}{2}\left(A_{i\pm\frac{1}{2}}^R - \tilde{A}_{i\pm\frac{1}{2}}\right) \qquad (4.8)$$

$$\hat{A}_{i\pm\frac{1}{2}}^L = \frac{1}{2}\left(A_{i\pm\frac{1}{2}}^L + \tilde{A}_{i\pm\frac{1}{2}}\right) \qquad (4.9)$$

However, for the van Leer scheme,

$$\mathbf{E}_{i+\frac{1}{2}} = \mathbf{E}_L + \mathbf{E}_R = \mathbf{E}^+_{i+\frac{1}{2}} + \mathbf{E}^-_{i+\frac{1}{2}} \tag{4.10}$$

$$\begin{aligned}\mathbf{E}_L^{n+1} &= \mathbf{E}_L^n + \left(\frac{\partial \mathbf{E}^+}{\partial \mathbf{U}}\right)_L^n \Delta\mathbf{U}_L^{n+1} \\ &= \mathbf{E}_L^n + J^L_{i+\frac{1}{2}}\Delta\mathbf{U}_i^{n+1}\end{aligned} \tag{4.11}$$

$$\begin{aligned}\mathbf{E}_R^{n+1} &= \mathbf{E}_R^n + \left(\frac{\partial \mathbf{E}^-}{\partial \mathbf{U}}\right)_R^n \Delta\mathbf{U}_R^{n+1} \\ &= \mathbf{E}_R^n + J^R_{i+\frac{1}{2}}\Delta\mathbf{U}_{i+1}^{n+1}\end{aligned} \tag{4.12}$$

Therefore,

$$\begin{aligned}&\mathbf{E}^{n+1}_{i+\frac{1}{2}} - \mathbf{E}^{n+1}_{i-\frac{1}{2}} \\ &\quad = \left(\mathbf{E}^n_{i+\frac{1}{2}} - \mathbf{E}^n_{i-\frac{1}{2}}\right) + J^R_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i+1} + J^L_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i} - J^R_{i-\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i} - J^L_{i-\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i-1}\end{aligned} \tag{4.13}$$

If let

$$\hat{A}^L_{i\pm\frac{1}{2}} = J^L_{i\pm\frac{1}{2}},\ \hat{A}^R_{i\pm\frac{1}{2}} = J^R_{i\pm\frac{1}{2}} \tag{4.14}$$

Eq. (4.13) will have the same form as Eq. (4.7).

The viscous flux $\mathbf{R}$ at time level $n+1$ on the right hand side is implicitly discretized by “freezing” the viscosity and evaluating the cross-derivative terms at time level $n$,

$$\begin{aligned}\mathbf{R}^{n+1}_{i+\frac{1}{2}} &= \mathbf{R}^n_{i+\frac{1}{2}} + \frac{\partial \mathbf{R}_{i+\frac{1}{2}}}{\partial \mathbf{U}_{i+1}}\Delta\mathbf{U}^{n+1}_{i+1} + \frac{\partial \mathbf{R}_{i+\frac{1}{2}}}{\partial \mathbf{U}_{i}}\Delta\mathbf{U}^{n+1}_{i} \\ &= \mathbf{R}^n_{i+\frac{1}{2}} + L^R_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i+1} + L^L_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i}\end{aligned} \tag{4.15}$$

Therefore,

$$\mathbf{R}^{n+1}_{i+\frac{1}{2}} - \mathbf{R}^{n+1}_{i-\frac{1}{2}} = \mathbf{R}^{n}_{i+\frac{1}{2}} - \mathbf{R}^{n}_{i-\frac{1}{2}} + L^{R}_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i+1} + \left(L^{L}_{i+\frac{1}{2}} - L^{R}_{i-\frac{1}{2}}\right)\Delta\mathbf{U}^{n+1}_{i} - L^{L}_{i+\frac{1}{2}}\Delta\mathbf{U}^{n+1}_{i-1} \quad (4.16)$$

The other two directions are treated similarly. In $\xi$ direction, the formulations for $\mathbf{F}$ and $\mathbf{S}$ have coefficients $\hat{B}^{R,L}_{j\pm\frac{1}{2}}$, $M^{R,L}_{j\pm\frac{1}{2}}$. In $\zeta$ direction, the formulations for $\mathbf{G}$ and $\mathbf{T}$ have coefficients $\hat{C}^{R,L}_{k\pm\frac{1}{2}}$ and $N^{R,L}_{k\pm\frac{1}{2}}$. The exact formulation of $\hat{A}^{R,L}_{i\pm\frac{1}{2}}$, $\hat{B}^{R,L}_{j\pm\frac{1}{2}}$, $\hat{C}^{R,L}_{k\pm\frac{1}{2}}$ depends on the inviscid flux discretization scheme.

The source term is implicitly discretized as,

$$\mathbf{D}^{n+1}_{i,j,k} = \mathbf{D}^{n}_{i,j,k} + \frac{\partial \mathbf{D}_{i,j,k}}{\partial \mathbf{U}_{i,j,k}}\Delta\mathbf{U}^{n+1}_{i,j,k} \quad (4.17)$$

Substitute Eqs. (4.7), (4.16), (4.17) and other similar terms on $\eta$ and $\zeta$ directions into the implicit integral governing equation (4.1). Move all the terms with $\Delta\mathbf{U}^{n+1}$ to the left hand side and the rest to the right hand side, the integrated governing equations are written in implicit form as,

$$\begin{aligned}(I-\Delta D)\,\Delta\mathbf{U}^{n+1}_{i,j,k} + A^{+}\Delta\mathbf{U}^{n+1}_{i+1,j,k} + A\Delta\mathbf{U}^{n+1}_{i,j,k} + A^{-}\Delta\mathbf{U}^{n+1}_{i-1,j,k} + B^{+}\Delta\mathbf{U}^{n+1}_{i,j+1,k} + B\Delta\mathbf{U}^{n+1}_{i,j,k} \\ + B^{-}\Delta\mathbf{U}^{n+1}_{i,j-1,k} + C^{+}\Delta\mathbf{U}^{n+1}_{i,j,k+1} + C\Delta\mathbf{U}^{n+1}_{i,j,k} + C^{-}\Delta\mathbf{U}^{n+1}_{i,j,k-1} = \mathbf{RHS}^{n} \quad (4.18)\end{aligned}$$

where the coefficients $A$, $A^{\pm}$, $B$, $B^{\pm}$ and $C$, $C^{\pm}$ are defined as,

$$A^{+} = \frac{\Delta t}{\Delta V}\left(\hat{A}^{R}_{i+\frac{1}{2}} - L^{R}_{i+\frac{1}{2}}\right) \quad (4.19)$$

$$A = \frac{\Delta t}{\Delta V}\left[\left(\hat{A}^{L}_{i+\frac{1}{2}} - L^{L}_{i+\frac{1}{2}}\right) - \left(\hat{A}^{R}_{i-\frac{1}{2}} - L^{R}_{i-\frac{1}{2}}\right)\right] \quad (4.20)$$

$$A^{-} = -\frac{\Delta t}{\Delta V}\left(\hat{A}^{L}_{i-\frac{1}{2}} - L^{L}_{i-\frac{1}{2}}\right) \quad (4.21)$$

$$B^{+} = \frac{\Delta t}{\Delta V}\left(\hat{B}^{R}_{j+\frac{1}{2}} - M^{R}_{j+\frac{1}{2}}\right) \quad (4.22)$$

$$B = \frac{\Delta t}{\Delta V}\left[\left(\hat{B}^L_{j+\frac{1}{2}} - M^L_{j+\frac{1}{2}}\right) - \left(\hat{B}^R_{j-\frac{1}{2}} - M^R_{j-\frac{1}{2}}\right)\right] \tag{4.23}$$

$$B^- = -\frac{\Delta t}{\Delta V}\left(\hat{B}^L_{j-\frac{1}{2}} - M^L_{j-\frac{1}{2}}\right) \tag{4.24}$$

$$C^+ = \frac{\Delta t}{\Delta V}\left(\hat{C}^R_{k+\frac{1}{2}} - N^R_{k+\frac{1}{2}}\right) \tag{4.25}$$

$$C = \frac{\Delta t}{\Delta V}\left[\left(\hat{C}^L_{k+\frac{1}{2}} - N^L_{k+\frac{1}{2}}\right) - \left(\hat{C}^R_{k-\frac{1}{2}} - N^R_{k-\frac{1}{2}}\right)\right] \tag{4.26}$$

$$C^- = -\frac{\Delta t}{\Delta V}\left(\hat{C}^L_{k-\frac{1}{2}} - N^L_{k-\frac{1}{2}}\right) \tag{4.27}$$

$\mathbf{RHS}^n$ is the summation of all the terms on the right hand side (RHS) of the equation.

$$\begin{aligned}\mathbf{RHS}^n = \frac{\Delta t}{\Delta V}\Big\{&\left[\left(\mathbf{R}^n_{i+\frac{1}{2}} - \mathbf{R}^n_{i-\frac{1}{2}}\right) + \left(\mathbf{S}^n_{j+\frac{1}{2}} - \mathbf{S}^n_{j-\frac{1}{2}}\right) + \left(\mathbf{T}^n_{k+\frac{1}{2}} - \mathbf{T}^n_{k-\frac{1}{2}}\right)\right] \\ &- \left[\left(\mathbf{E}^n_{i+\frac{1}{2}} - \mathbf{E}^n_{i-\frac{1}{2}}\right) + \left(\mathbf{F}^n_{j+\frac{1}{2}} - \mathbf{F}^n_{j-\frac{1}{2}}\right) + \left(\mathbf{G}^n_{k+\frac{1}{2}} - \mathbf{G}^n_{k-\frac{1}{2}}\right)\right]\Big\} + \mathbf{D}^n \cdot \Delta t\end{aligned} \tag{4.28}$$

$\Delta D$ is a term resulting from the source term,

$$\Delta D = \Delta t \cdot \left(\frac{\partial \mathbf{D}}{\partial \mathbf{U}}\right)^n_{i,j,k} \tag{4.29}$$

Eq. (4.18) also means that the first order accuracy is always used for the left hand side (LHS), which has the advantage of diagonal dominance [92]. Since the accuracy of the converged solution is controlled by the RHS, the first order LHS will not affect the final solution. The diagonal dominance of the LHS matrix is the necessary and sufficient conditions for Gauss-Seidel iteration to converge.

In the current work, when the implicit method is used for time marching, the LHS van Leer coefficient matrices are used for all upwind schemes other than the Roe scheme.

## 4.2 Gauss-Seidel Line Iteration

Eq. (4.18) is the final implicitly discretized linear equation system for the Navier-Stokes governing equations (2.44). This linear equation system is then solved by the Gauss-Seidel line iteration method. Eq. (4.18) is rearranged as the following,

$$[(I-\Delta D)+A+B+C]\Delta\mathbf{U}_{i,j,k}^{n+1}+A^{+}\Delta\mathbf{U}_{i+1,j,k}^{n+1}+A^{-}\Delta\mathbf{U}_{i-1,j,k}^{n+1}+B^{+}\Delta\mathbf{U}_{i,j+1,k}^{n+1} \\ +B^{-}\Delta\mathbf{U}_{i,j-1,k}^{n+1}+C^{+}\Delta\mathbf{U}_{i,j,k+1}^{n+1}+C^{-}\Delta\mathbf{U}_{i,j,k-1}^{n+1}=\mathbf{RHS}^{n} \quad (4.30)$$

The Gauss-Seidel line iteration is applied on each direction ($\xi$, $\eta$, $\zeta$) respectively and is swept forward and backward on each direction. Take $\eta$ direction for example, the equation form for Gauss-Seidel iteration following lines along $\eta$ axis ($\xi$ and $\zeta$ are constant) is written as,

$$B^{-}\Delta\mathbf{U}_{i,j-1,k}^{n+1}+\overline{B}\Delta\mathbf{U}_{i,j,k}^{n+1}+B^{+}\Delta\mathbf{U}_{i,j+1,k}^{n+1}=\mathbf{RHS}' \quad (4.31)$$

where

$$\overline{B}=(I-\Delta D)+A+B+C \quad (4.32)$$

The terms in the neighboring cells in $\xi$ and $\zeta$ directions are absorbed into $\mathbf{RHS}^n$ to make $\mathbf{RHS}'$ as the following,

$$\mathbf{RHS}'=\mathbf{RHS}^{n}-A^{+}\Delta\mathbf{U}_{i+1,j,k}^{n+1}-A^{-}\Delta\mathbf{U}_{i-1,j,k}^{n+1}-C^{+}\Delta\mathbf{U}_{i,j,k+1}^{n+1}-C^{-}\Delta\mathbf{U}_{i,j,k-1}^{n+1} \quad (4.33)$$

When iterating through all $\eta$ direction lines, the coefficients $B^{\pm}$, $\overline{B}$, $A^{\pm}$, $\overline{A}$, $C^{\pm}$, $\overline{C}$ and the right hand side $\mathbf{RHS}^n$ are all evaluated based on the flow variables on the previous time level $n$. However, the $\Delta\mathbf{U}^{n+1}$ vectors all use the most recent values whenever they are calculated as required by Gauss-Seidel iteration. Same rules apply for the line iterations in the other two directions.

In the steady state calculation or the iteration within each physical time step in the unsteady calculation, the above Gauss-Seidel line iteration are repeated until the prescribed convergence criteria is reached.

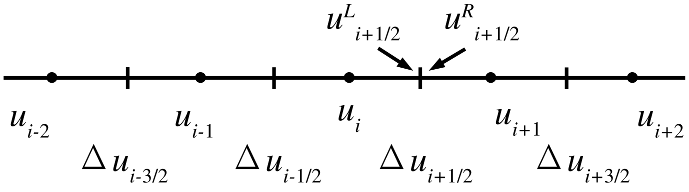


Figure 4.1: MUSCL approach

## 4.3 MUSCL Differencing Approach

To achieve higher order schemes for the inter surface inviscid flux computation, the primitive variables, the density $\rho$, the velocity $\mathbf{V}$ and the energy $e$ are first extrapolated to the cell interface as left and right values using the values on neighboring cells. The inviscid fluxes are then reconstructed based on these extrapolated left and right primitive variables. The MUSCL (monotone upstream-centered schemes for conservation laws) differencing approach suggested by van Leer [120] is applied to extrapolate the primitive variables [121].

As shown in Fig. 4.1 , for a cell with index of $i$, the left and right hand side reconstructed variable $u$ on its inter surface $i+\frac{1}{2}$ is computed as the following,

$$u^L_{i+\frac{1}{2}} = u_i + \frac{\phi}{4}\left[(1-\kappa)\,\Delta u_{i-\frac{1}{2}} + (1+\kappa)\,\Delta u_{i+\frac{1}{2}}\right] \tag{4.34}$$

$$u^R_{i+\frac{1}{2}} = u_{i+1} - \frac{\phi}{4}\left[(1+\kappa)\,\Delta u_{i+\frac{1}{2}} + (1-\kappa)\,\Delta u_{i+\frac{3}{2}}\right] \tag{4.35}$$

where

$$\Delta u_{i+\frac{1}{2}} = u_{i+1} - u_i,\ \Delta u_{i-\frac{1}{2}} = u_i - u_{i-1} \tag{4.36}$$

When $\phi = 0$, the first order interpolation is used for $u$. When $\phi = 1$, the constant $\kappa$

determines the order of accuracy of the MUSCL approach. For example,

$$\kappa = \begin{cases} -1 & \text{second order fully upwind scheme} \\ 0 & \text{second order Fromm's method} \\ \frac{1}{3} & \text{third order upwind biased} \\ 1 & \text{central difference} \end{cases} \tag{4.37}$$

Both the upwind scheme ($\kappa$=-1) and central difference ($\kappa$=1) are linear approximations to the cell boundary values of $u$ and both have the second order of accuracy. In the current work, $\kappa = \frac{1}{3}$ is used, therefore the third order of accuracy is achieved for spatial differencing of the inviscid fluxes.

It should be mentioned that the third order accuracy is applied only in the interior cells. For the first two layers of boundary cells, the left and right side values on the the control volume surface use the center values of the left and right neighboring cells, which have the first order accuracy. The MUSCL differencing is employed with the Minmod limiter [122] and the limiter suggested by Anderson et. al [123].

## 4.4 Boundary Conditions

In the following chapters, numerical simulation is carried out for both transonic and high subsonic systems with various boundary conditions, which include supersonic inlet, supersonic outlet, subsonic inlet, subsonic outlet, wall surface, symmetric boundary, periodic boundary. In the parallel computation cases, a two-layer domain partitioning boundary condition is applied for the interface between two neighboring sub-domains. These boundary conditions are indexed in the code as Table. 4.1 .

To apply these boundary conditions, additional cells are attached to the outside of the domain boundary. There are two cell layers added for the periodic and the domain partitioning boundaries. The coordinates and the variables in these additional cells are copied

Table 4.1: Boundary condition indexes in the RANS3D solver

| Index | Boundary Condition | Index | Boundary Condition |
|---|---|---|---|
| 1 | Supersonic outlet | 6 | Subsonic inlet with velocity specified |
| 2 | Supersonic inlet | 7 | Domain partitioning boundary |
| 3 | Wall | 8 | Symmetry boundary |
| 5 | Subsonic outlet | 9 | Subsonic inlet with stagnation variables specified |
| | | 10 | Periodic boundary |

from other part of the domain or exchanged from the neighboring domains. For boundary conditions other than the periodic and domain partitioning boundaries, the domain is attached a single layer of phantom cells, where no coordinate information is required. The variables in the phantom layer are computed based on the inner neighboring cells or are specified based on different boundary conditions.

The numerical treatment of these boundary conditions are described as the following.

## Supersonic inlet

At supersonic inlet, all the primitive variables including the density $\rho$, velocity $\mathbf{V}$, energy $e$ are fixed as constants at the boundary. Other variables, for example the static pressure $p$, static temperature $T$, are computed from the specified variables.

## Supersonic outlet

At the supersonic outlet boundary, all waves are leaving the computation domain. Therefore all the primitive variables ($\rho$, $\mathbf{V}$ and $e$) are extrapolated with zero gradient from their inner counterparts,

$$\rho_b = \rho_i, \mathbf{V}_b = \mathbf{V}_i, e_b = e_i \tag{4.38}$$

where the subscript $b$ and $i$ stand for the boundary at outlet and first inner cell close to the boundary. The other derived flow variables are calculated based on these extrapolated primitive variables accordingly.

### Subsonic inlet

At subsonic inlet boundary, four of the five waves enter the domain and one leaves the domain. Therefore, there are 4 variables prescribed at the the boundary and 1 variable extrapolated from the inner neighboring cell. In the current work, there are two types of subsonic inlet boundary conditions employed in the code, based on the different variable combinations for the prescribed and extrapolated variables.

1. Subsonic inlets with velocity specified

   The 3 velocity components $u$, $v$ and $w$ and the density $\rho$ are given as constant at the subsonic inlet boundary. The extrapolated variable is the pressure $p$. All other flow variables at the boundary are then calculated from these 5 variables.

2. Subsonic inlet with stagnation variables specified

   In reality, at subsonic inlet boundary, the prescribed variables at the boundary are usually total pressure $p_t$, total temperature $T_t$ and the two flow angles $\alpha$ and $\beta$ instead of the velocity and density. The total variables are easier for experiment measurement and more common in engineering. In the current work, most of the subsonic inlet boundaries are treated in this way. In the code, a velocity component is extrapolated from the interior cells. The other variables are calculated based on these 5 variables.

   For example, in $x$ direction, when $P_t, T_t, \alpha, \beta$ are specified and $u$ is extrapolated from inner domain, the other normalized flow variables are calculated as the following. .

   Velocity:

$$u_b = u_i, \, v_b = u_b \tan\alpha, \, w_b = u_b \tan\beta \tag{4.39}$$

   Temperature:

$$T_b = T_t - \frac{1}{2}(\gamma - 1) M_\infty^2 \left(u_b^2 + v_b^2 + w_b^2\right) \tag{4.40}$$

Pressure:

$$p_b = P_t \left[ 1 + \frac{\gamma - 1}{2} \frac{M_\infty^2}{T_b} \left( u_b^2 + v_b^2 + w_b^2 \right) \right]^{\frac{\gamma}{\gamma-1}} \tag{4.41}$$

Density:

$$\rho_b = \frac{\gamma M_\infty^2 p_b}{T_b} \tag{4.42}$$

Energy:

$$e_b = \frac{1}{\gamma(\gamma-1)} \frac{1}{M_\infty^2} T_b + \frac{1}{2} \left( u_b^2 + v_b^2 + w_b^2 \right) \tag{4.43}$$

## Subsonic outlet

At the subsonic outlet boundary, four waves leave the domain and one wave enters the domain. Hence, the static pressure $p$ is specified for the wave entering the domain. Four of the five primitive flow variables, $u$, $v$, $w$ and $\rho$ are extrapolated from the inner domain. These five variables are then used to calculate other variables at the boundary.

## Wall boundary

The wall surface boundary is treated as no-slip and adiabatic wall in the current work. The velocity components in the phantom cells are determined with the no-slip condition. The temperature is determined by the adiabatic condition.

When the boundary is stationary, the velocity on the wall is simply set as zero. This is made possible by setting the phantom cell velocity to be the negative of the velocity at the first interior cell velocity,

$$\mathbf{V}_b = -\mathbf{V}_i \tag{4.44}$$

However, this method is only valid when the central differencing is applied on the boundary for viscous fluxes. For inviscid fluxes the contravariant velocity on the boundary is set as zero directly. For example, in the inviscid flux **F**, the contravariant velocity $V$ is directly set to be zero on the wall surface and the flux on the wall surface is calculated as,

$$\mathbf{F}_w = \begin{bmatrix} 0 \\ m_x p \\ m_y p \\ m_z p \\ 0 \end{bmatrix} \tag{4.45}$$

When the boundary is moving, the wall velocity is calculated based on the wall grid point moving distance between two sequential time levels. This prescribed wall velocity is enforced by extrapolating the velocity between the phantom and the neighboring interior cells. For example in 2D cases,

$$u_b = 2\dot{x}_w - u_i, \, v_b = 2\dot{y}_w - v_i \tag{4.46}$$

where $\dot{x}_w$, $\dot{y}_w$ are the velocity components of moving wall boundary.

$$\dot{x}_w = \frac{x_w^{n+1} - x_w^n}{\Delta t}, \, \dot{y_w} = \frac{y_w^{n+1} - y_w^n}{\Delta t} \tag{4.47}$$

The other two conditions to be imposed on the solid wall are the adiabatic wall condition and the inviscid normal momentum equation [124]. For example at the $\eta$ direction wall boundary in 2D calculation,

$$\frac{\partial T}{\partial \eta} = 0 \tag{4.48}$$

$$\frac{\partial p}{\partial \eta} = -\left(\frac{\rho}{\eta_x^2 + \eta_y^2}\right)\left(\eta_x \ddot{x}_{\mathrm{b}} + \eta_y \ddot{y}_{\mathrm{b}}\right) \tag{4.49}$$

where $\ddot{x}_w$ and $\ddot{y}_w$ are the wall surface acceleration components.

$$\ddot{x}_w = \frac{\dot{x}_w^{n+1} - \dot{x}_w^n}{\Delta t}, \, \ddot{y}_w = \frac{\dot{y}_w^{n+1} - y_w^n}{\Delta t} \tag{4.50}$$

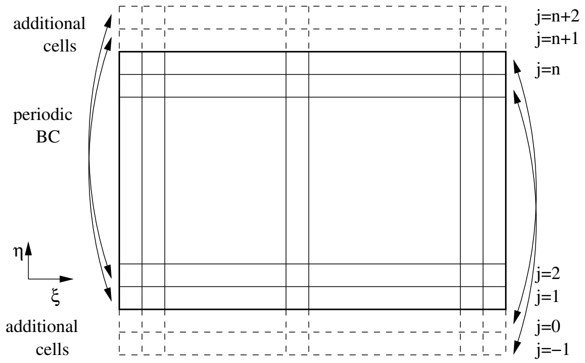


Figure 4.2: Periodic boundary condition

When the boundary is stationary, Eq. (4.49) is simplified to the zero gradient form,

$$\frac{\partial p}{\partial \eta} = 0 \tag{4.51}$$

## Symmetrical boundary

The velocity in the phantom cell is mirrored about the symmetrical boundary from its first neighboring interior cell. The velocity components parallel to the boundary are extrapolated with zero gradient. The normal velocity component is set as negative to its interior counterpart. Other parameters, such as the density $\rho$ and the energy $e$, the temperature $T$ and the pressure $p$ are extrapolated with zero gradient.

## Periodic boundary

On the periodical boundaries, the flow variable and the mesh coordinates are repeated periodically in a certain direction. Fig. 4.2 shows one periodical boundary example in $\eta$ direction. The flow variables and the mesh information on the additional cell layers ($j = n+1$, $n+2$) are copied directly from the corresponding first and second inner cells in the opposite boundary ($j = 1, 2$). When calculating **RHS** in the integral governing equation (4.18), the viscosity on the control volume surface is required and it is calculated as the average viscosity between the neighboring grid points with the weight of their volume magnitudes.

On the computation domain surface, the additional cells are real cells with real volumes. Therefore, the mesh coordinate information is also copied from the corresponding interior cells to the additional cells at the periodical boundary to compute the real volumes.

It should be noted that, because higher order differencing is used in the code, there are two additional cell layers ($j = n+1$, $n+2$ on top for example) attached on the boundary. In this way the accuracy of the surface flux computation for the periodic boundary points is kept the same as the inner points. When solving the discretized linear governing equations on the lines with periodical boundary at their end points (for example the vertical lines in Fig. 4.2), the periodic block triangular solver is used in the Gauss-Seidel line iteration.

### Domain partitioning boundary

This boundary condition only exists in parallel computation. When the computation domain is split into several sub-domains, two neighboring sub-domains share the interface boundary. Like the periodic boundary, there are two additional cell layers allocated for the boundary. The flow variables and the mesh information are copied between the two neighboring sub-domains. The implementation of the domain partitioning boundary is described later in this chapter.

## 4.5 Local Time Stepping

When the governing equations are solved, the physical time step is expected to be as large as possible to save the computation CPU time. However, in order to achieve the numerical stability, the time step is restricted by the numerical method and the flow field characteristics. The time step in the code is determined by the following,

$$\Delta t = CFL \frac{\Delta V_{ijk}}{\max\Big(|U| + |C_\xi|,\, |V| + |C_\eta|,\, |W| + |C_\zeta|\Big)} \tag{4.52}$$

where $CFL$ is the Courant-Friedrichs-Levy condition number, $\Delta V_{ijk}$ is the cell volume at location $(i,j,k)$, $U$, $V$ and $W$ are the contravariant velocities at the cell center in $\xi$, $\eta$ and $\zeta$ directions, $C_\xi$, $C_\eta$ and $C_\zeta$ are the corresponding contravariant speeds of sound.

$$C_\xi = c\sqrt{l_x^2+l_y^2+l_z^2},\ C_\eta = c\sqrt{m_x^2+m_y^2+m_z^2},\ C_\zeta = c\sqrt{n_x^2+n_y^2+n_z^2} \tag{4.53}$$

It is shown in Eq. (4.52) that the time step size is proportional to the volume magnitude of the mesh cell. In the viscous flow cases, the mesh is highly stretched in the near wall regions and the areas around the shock wave to resolve the high gradient. In these regions, the mesh size is several orders of magnitude smaller than the regions far away from the wall and shock wave. Hence, a large range of time step scales exist, which may vary in several orders of magnitude. The time step is very small in the regions with very fine mesh compared with that in the free stream.

If the time accuracy is required during the integrations, for example in explicit time marching, the time step should be uniform across the whole computation domain. Thus the minimum time step which is determined by the smallest volume should be used. The constrain of the uniform time step brings high computation cost. However, if only the steady state solution is required, the local time stepping method suggested by Li [73] can be used to accelerate the convergence of the iteration. The iteration at each point is advanced using the local maximum possible time step computed by Eq. (4.52). This method is used in the present work for convergence acceleration.

## 4.6 Dual Time Stepping

In the unsteady simulation, the physical time step in Eq. (2.44) should be uniform on all mesh cells in the computation domain. The local time stepping is not applicable any more. The computation cost will be very high if explicit scheme is used. The implicit scheme can have much larger time step. However, when parallel computation is used, it is difficult to preserve the time accuracy across the domain partitioning boundaries. The dual time stepping method suggested by Jameson [76] is then employed to resolve this problem.

For dual time stepping, a pseudo temporal term $\frac{\partial \mathbf{U}}{\partial \tau}$ is added to the left side of the governing equation(2.44). The new governing equations with the pseudo term are hence rewritten in the Cartesian coordinates as,

$$\frac{\partial \mathbf{U}}{\partial \tau}+\frac{\partial \mathbf{U}}{\partial t}=-\frac{\partial\left(\mathbf{E}-\mathbf{R}\right)}{\partial x}-\frac{\partial\left(\mathbf{F}-\mathbf{S}\right)}{\partial y}-\frac{\partial\left(\mathbf{G}-\mathbf{T}\right)}{\partial z}+\mathbf{D} \tag{4.54}$$

The physical time marching is still conducted with the physical time $t$ as discussed above. However, for each physical time step $\Delta t$, the time marching is not finished until the iteration over the pseudo time $\tau$ is converged, that is,

$$\frac{\partial \mathbf{U}}{\partial \tau}=-\frac{\partial\left(\mathbf{E}-\mathbf{R}\right)}{\partial x}-\frac{\partial\left(\mathbf{F}-\mathbf{S}\right)}{\partial y}-\frac{\partial\left(\mathbf{G}-\mathbf{T}\right)}{\partial z}+\mathbf{D}-\frac{\partial \mathbf{U}}{\partial t}=0 \tag{4.55}$$

Then, the new equation (4.54) is reverted to the original governing equation (2.44). Because the pseudo time term vanishes in the final converged solution in each physical time step, the accuracy of the physical time marching is not affected. Since the pseudo time iteration has no accuracy requirement, the convergence acceleration techniques, such as the local time stepping can still be used. The physical time step can be very large since the implicit scheme is used.

The dual time stepping is implemented in the present work as the following.

Eq.(4.54) is integrated on the control volume as,

$$\frac{\partial \mathbf{U}}{\partial \tau}+\frac{\partial \mathbf{U}}{\partial t}=-\frac{1}{\Delta V}\oint_S \mathbf{R}_F\cdot \mathrm{d}\mathbf{s}+\mathbf{D} \tag{4.56}$$

where $\mathbf{s}$ is the surface area vector, $\mathbf{R}_F$ is the net flux vector,

$$\mathbf{R}_F=\left(\mathbf{E}-\mathbf{R}\right)\mathbf{i}+\left(\mathbf{F}-\mathbf{S}\right)\mathbf{j}+\left(\mathbf{G}-\mathbf{T}\right)\mathbf{k} \tag{4.57}$$

Let

$$\mathbf{RHS}=-\frac{1}{\Delta V}\oint_S \mathbf{R}_F \mathrm{d}\mathbf{s}+\mathbf{D} \tag{4.58}$$

The $\frac{\partial \mathbf{U}}{\partial t}$ is discretized using the second order 3 point backward differencing. Use the first order discretization for $\frac{\partial \mathbf{U}}{\partial \tau}$ on two pseudo time steps ($m$ and $m+1$) on the same physical time step $n+1$, the implicit discretization of the governing equations (4.54) is written as,

$$\frac{\mathbf{U}^{n+1,m+1}-\mathbf{U}^{n+1,m}}{\Delta\tau}+\frac{3\mathbf{U}^{n+1,m+1}-4\mathbf{U}^{n}+\mathbf{U}^{n-1}}{2\Delta t}=\mathbf{RHS}^{n+1,m+1} \tag{4.59}$$

Let

$$\Delta\mathbf{U}^{n+1,m+1}=\mathbf{U}^{n+1,m+1}-\mathbf{U}^{n+1,m} \tag{4.60}$$

$$\Delta\mathbf{RHS}^{n+1,m+1}=\mathbf{RHS}^{n+1,m+1}-\mathbf{RHS}^{n+1,m} \tag{4.61}$$

Therefore

$$\begin{aligned}\frac{\Delta\mathbf{U}^{n+1,m+1}}{\Delta\tau}+\frac{3\mathbf{U}^{n+1,m}+3\Delta\mathbf{U}^{n+1,m+1}-4\mathbf{U}^{n}+\mathbf{U}^{n-1}}{2\Delta t}&\\ =\mathbf{RHS}^{n+1,m}+\Delta\mathbf{RHS}^{n+1,m+1}&\end{aligned} \tag{4.62}$$

The final formulation for the dual time stepping is,

$$\begin{aligned}\left[\left(\frac{1}{\Delta\tau}+\frac{1.5}{\Delta t}\right)I-\left(\frac{\partial\mathbf{RHS}}{\partial\mathbf{U}}\right)^{n+1,m}\right]\Delta\mathbf{U}^{n+1,m+1}&\\ =\mathbf{RHS}^{n+1,m}-\frac{3\mathbf{U}^{n+1,m}-4\mathbf{U}^{n}+\mathbf{U}^{n-1}}{2\Delta t}&\end{aligned} \tag{4.63}$$

When the iteration is converged within one physical time step, $\Delta\mathbf{U}^{n+1,m+1}\rightarrow 0$.

The implementation of Eq. (4.63) is based on Eq. (4.18). If divide Eq. (4.18) and then compare it with Eq. (4.63), the only difference for the LHS is the temporal term. $\frac{1}{\Delta t}$ in Eq. (4.18) is replaced by $\frac{1}{\Delta\tau}+\frac{1.5}{\Delta t}$ in Eq. (4.63). On the RHS, an extra term, $-\frac{3\mathbf{U}^{n+1,m}-4\mathbf{U}^{n}+\mathbf{U}^{n-1}}{2\Delta t}$ is added to Eq. (4.18).

## 4.7 Parallel Computation

In turbomachinery, the periodic boundary condition is usually used for steady state simulation and the computation is carried out in a single passage. However, when the periodicity is lost, such as in the mistuned rotor vibration, the full annulus rotor calculation is required. Even though the multi-passage simulation can be conducted with a single processor by going through each blade sequentially, it would be too CPU expensive. Another disadvantage of this method is that, the flows in all passages are highly coupled in reality, solving these passages sequentially may be difficult to get converged solutions.

A better way for the turbomachinery unsteady simulation is using parallel computer, which has multiple CPUs installed on a single machine with shared memory. Most of parallel computation algorithms apply the approach of “divide and conquer”. The computation job is split into multiple sub-jobs and these sub-jobs can be processed by multiple processors simultaneously. However, this method has its own disadvantages. The hardware will be too costly especially when a large number of processors are needed to be installed on a single machine. Besides, the original single passage code will be updated based on the specific hardware and operating system configuration. This is expensive and inconvenient. The final code will become poor in portability.

The distributed computation with the MPI standard makes the parallel computation of the turbomachinery more economic and portable [90]. Each subdomain is computed by a processor. The two topologically neighboring subdomains communicate with each other using the MPI protocol. These two processors do not have to be neighbors physically in the computer system. The location for a specific process is maintained by the MPI routines for the highest efficiency. Currently, there are several popular implementations of the MPI standard. In current work, the parallel simulation is implemented using the LAM/MPI, which is a high quality implementation of the MPI Standard. It provides high performance on a variety of platforms.

A sample 2D parallel computation algorithm is shown in Fig. 4.3 . These two subdo-

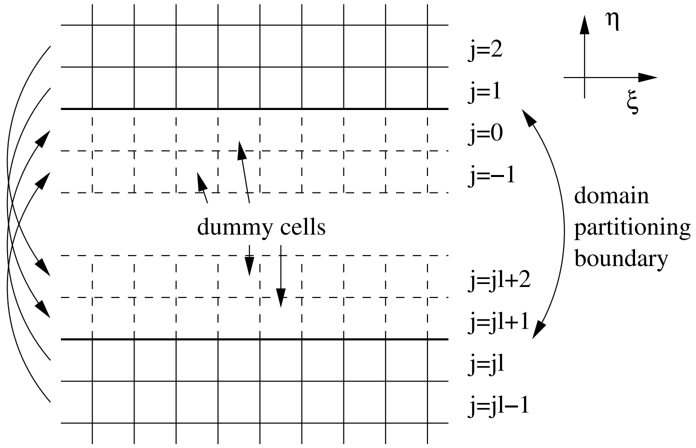


Figure 4.3: Sketch of MPI data exchange

mains are decomposed in $\eta$ direction. The two domain partitioning boundaries are identical in reality. Two rows of additional dummy cells are added below the upper domain and above the lower domain. This is needed by the higher order differencing schemes used to discretize the governing equations. The data in the dummy cells are received from the corresponding cells directed as the arrows on the left side, that is, from upper domain to lower domain, $(1 \rightarrow jl+1)$, $(2 \rightarrow jl+2)$ and from lower domain to upper domain $(jl \rightarrow 0)$, $(jl-1 \rightarrow -1)$. The information exchanged includes the primitive flow variables $\rho$, $\mathbf{V}$, $e$ and the coordinates $x$, $y$, $z$.

The decomposition can be extended to 3D cases as shown in Fig. 4.4 . In stead of passing 2 lines of data as in Fig. 4.3, two planes of data are exchanged in the horizontal and vertical directions as the arrows indicate.

When the above data exchange is implemented for an explicit time marching method across the partitioning boundary, the parallel computation is as efficient as the single processor computation. The variable on each cell center is updated based on the variables on the neighboring cells from the last time step. On the partitioning boundary cells, the variable update is the same as the inner cells. The variables on the additional cell layers are exchanged from the neighboring domain right after the inner cells are updated. The boundary additional cells and the inner cells are updated synchronously and no efficiency

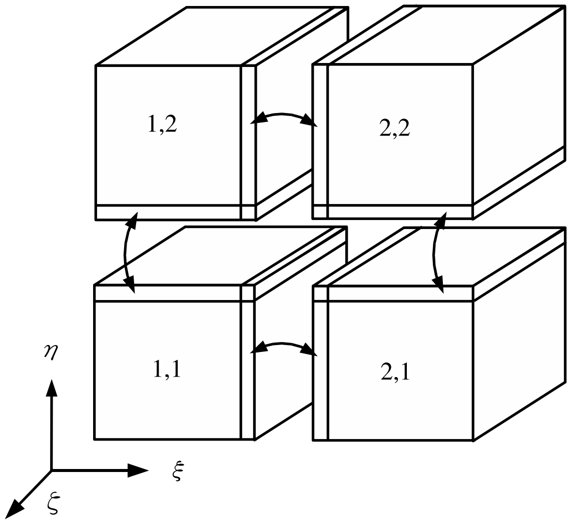


Figure 4.4: 3D MPI data exchange

is lost because of parallel computation.

However, when the implicit method is used for time marching, the computation efficiency and the time marching accuracy will be affected by the existence of the partitioning boundary. Different from the explicit method, in implicit solver, the variables on the neighboring cells are related to each other and are updated synchronously by matrix inversion. In the current study, the line Gauss-Seidel iteration is carried out on single lines in the computation domain. For example, in $\eta$ direction, the variables $\Delta\mathbf{U}$ on a single line $\xi = i$ and $\zeta = k$ is solved based on Eq. (4.31). When written in a matrix form, the equation is expressed as,

$$\left[\begin{array}{cccc|ccc} \bar{B}_1 & B_1^+ & & & & & \\ B_2^- & \bar{B}_2 & B_2^+ & & & 0 & \\ & & \ddots & & & & \\ & & B_m^- & \bar{B}_m & B_m^+ & & \\ -- & -- & -- & -- & -- & -- & -- \\ & & & B_{m+1}^- & \bar{B}_{m+1} & B_{m+1}^+ & \\ & 0 & & & & \ddots & \\ & & & & & B_n^- & \bar{B}_n \end{array}\right] \left[\begin{array}{c} \Delta\mathbf{U}_1 \\ \Delta\mathbf{U}_2 \\ \vdots \\ \Delta\mathbf{U}_m \\ -- \\ \Delta\mathbf{U}_{m+1} \\ \vdots \\ \Delta\mathbf{U}_3 \end{array}\right] = \left[\begin{array}{c} \mathbf{RHS}'_1 \\ \mathbf{RHS}'_2 \\ \vdots \\ \mathbf{RHS}'_m \\ -- \\ \mathbf{RHS}'_{m+1} \\ \vdots \\ \mathbf{RHS}'_n \end{array}\right] \tag{4.64}$$

If the $\eta$ line is split into 2 sections as $(0 \rightarrow m)$ and $(m+1 \rightarrow n)$ because of the $\eta$ direction domain decomposition, Eq. (4.64) will be split into two sub-matrices following the dashed lines,

$$\begin{bmatrix} \bar{B}_1 & B_1^+ & & \\ B_2^- & \bar{B}_2 & B_2^+ & \\ & & \ddots & \\ & & B_m^- & \bar{B}_m \end{bmatrix} \begin{bmatrix} \Delta\mathbf{U}_1 \\ \Delta\mathbf{U}_2 \\ \vdots \\ \Delta\mathbf{U}_m \end{bmatrix} = \begin{bmatrix} \mathbf{RHS}'_1 \\ \mathbf{RHS}'_2 \\ \vdots \\ \mathbf{RHS}'_m \end{bmatrix} \tag{4.65}$$

$$\begin{bmatrix} \bar{B}_{m+1} & B_{m+1}^+ & & \\ B_{m+2}^- & \bar{B}_{m+2} & B_{m+2}^+ & \\ & & \ddots & \\ & & B_n^- & \bar{B}_n \end{bmatrix} \begin{bmatrix} \Delta\mathbf{U}_{m+1} \\ \Delta\mathbf{U}_{m+2} \\ \vdots \\ \Delta\mathbf{U}_n \end{bmatrix} = \begin{bmatrix} \mathbf{RHS}'_{m+1} \\ \mathbf{RHS}'_{m+2} \\ \vdots \\ \mathbf{RHS}'_n \end{bmatrix} \tag{4.66}$$

The variables $\Delta\mathbf{U}_1 \rightarrow \Delta\mathbf{U}_m$ and $\Delta\mathbf{U}_{m+1} \rightarrow \Delta\mathbf{U}_n$ are correspondingly solved on two processors by conducting the matrix inversion on the above two sub-matrices. Comparing Eqs. (4.65) and (4.66) with the original Eq. (4.64) for the single processor computation, $B_m^+$ on the first section and $B_{m+1}^-$ on the second matrix are discarded in the two sub-matrices. They are the coefficients computed based on the variables from the neighboring sections. In the present work, these two coefficients are treated as zero in the implicit solver for the subdomain computations. The variables are updated independently on the two sub-domains. This LHS treatment does not preserve the exact matrices as in the single processor computation, which will negatively affect the convergence efficiency in steady state calculation and the time accuracy on the boundary cell variables in unsteady simulation. However, the computation experience indicates that the slow down of the convergence due to the non-exact LHS treatment is small, in particular when the mesh size is large. For unsteady time accurate calculation, the dual time stepping technique treats the partitioning boundary similar to the steady state calculation. The non-exact treatment of LHS occurs within the pseudo time step, which will not affect the time accuracy when the unsteady calculation is converged within each physical time step.

There are two important parameters to measure the parallel computation performance. The first is speedup, which is the serial execution time divided by the parallel execution time, given a fixed problem size and number of processors,

$$\text{speedup} = \frac{\text{time used for the job on a single CPU}}{\text{time used for the job on multiple CPUs}} \tag{4.67}$$

A perfect speedup equals the number of processors. The second parameter is scalability, which refers to how speedup changes as the problem size and number of processors increases. An ideal scalability is a linear increase of the speedup with the number of processors. However, when a large number of processors are used the scalability will become poor. The communication time will increase significantly with the increased processor number. If an implicit time marching is used, the iteration convergence will be influenced by the existence of the domain partitioning boundary. The domain decomposition should be carefully designed to minimize the influence.

In turbomachinery computation, the most straightforward domain decomposition is to follow the flow path and divide a blade row into subdomain passages. Then each passage is assigned to a single processor. In dual time stepping scheme, the flow variables of the domain partitioning boundary cells are exchanged across the boundaries after each pseudo time step. The mesh information of the boundary cells is exchanged every physical time step in the prescribed vibration and every pseudo time step in the vibration determined by the fully coupled fluid-structural interaction.

## 4.8 Moving Grid System

### 4.8.1 Algebraic mesh and elliptic mesh

The meshes in this work are generated algebraically or elliptically. In algebraic meshes, the mesh points are first distributed along the boundaries, such as walls, inlet, outlet etc. following a given stretch factor. Then the corresponding points on the boundaries are connected with lines. The cost of the elliptic mesh is much higher. Like the algebraic mesh, the grid points are first distributed along the given boundaries. Then the elliptic equation is solved to obtain the inner mesh. The grid lines in the elliptic mesh are more smooth compared with the algebraic mesh and hence have higher quality.

When the viscous boundary layer or wake need to be resolved, the mesh at these regions needs to have good orthogonality to assure the accuracy of the estimation of the turbulent viscosity, which will has a big influence on the flowfield results.

The algebraic mesh is simpler and more appropriate for cases within boundaries which are topologically simple, for example the flat plate boundary flow and the inlet diffuser flow. In the cylinder and airfoil cases with far field boundaries, the O-type meshes are also generated algebraically.

When the boundaries are complicated, the elliptic mesh is needed to keep the orthogonality in wall regions. For the cascade airfoil, the O-type mesh become inappropriate when the inlet and outlet are required to be extended. This brings difficulties for mesh generation. In the current work, the H-type elliptic meshes are used for the oscillating and stationary cascade flow simulations.

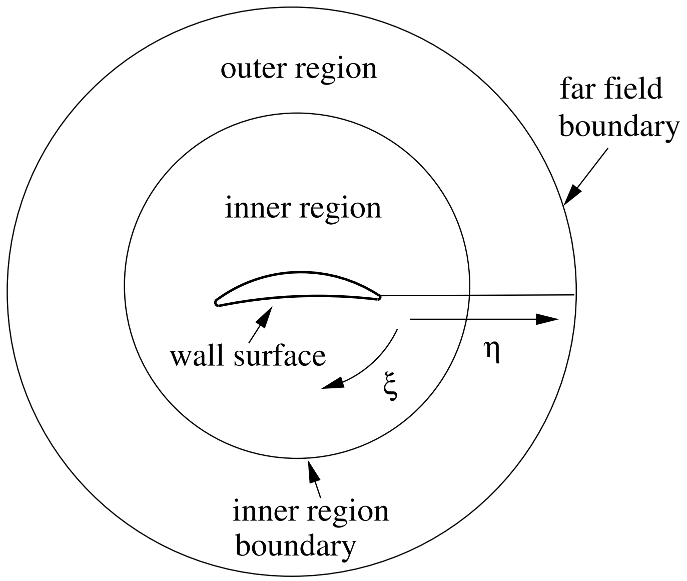


Figure 4.5: Multi-layer moving grid regeneration

### 4.8.2 Moving grid generation

When the moving grid system is simulated, the mesh will be regenerated every time when the solid wall boundary is moved. A straightforward method for the mesh update is to regenerate the mesh every time. However, several disadvantages are associated with this method. The mesh orthogonality in the wall region may not be maintained if the wall is distorted too much and the grid points distribution is kept unchanged on the wall boundaries. The cost of the mesh regeneration will be high for the elliptic equation solver. In the present work, the forced oscillating airfoil and the cascade use different moving grid generation models.

For the airfoil with far field boundary, the mesh is regenerated with a multi-region model. The computation mesh is split into an inner region and an outer region as shown in Fig. 4.5 . When the airfoil moves to a new position, the mesh in the inner region will move with the airfoil rigidly without deformation. Also kept unchanged is the outer far field boundary. The mesh in the outer region is however interpolated based on the far field boundary and the inner region boundary. The orthogonality is preserved in the inner layer.

In the cascade case, the H-type mesh is generated with a single-region model. As

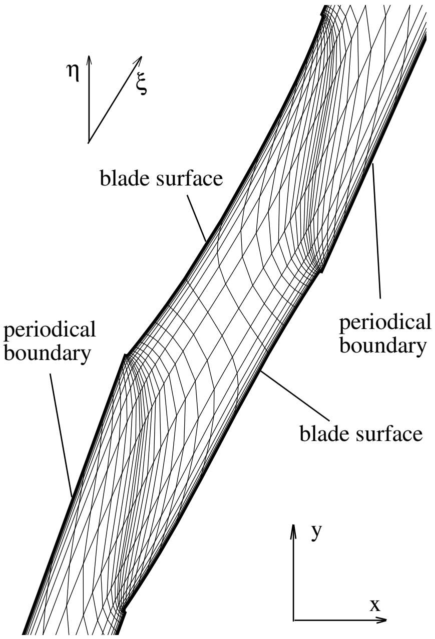


Figure 4.6: Sample cascade H-mesh

shown in Fig. 4.6 , the mesh is highly twisted in the computation domain because of its large stagger angle. The mesh is generated by solving the elliptic equations every time the wall surface moves.

## 4.9 The Fluid-Structural Interaction Procedure

The current solver is developed to calculate the unsteady flow with stationary or moving boundaries. As discussed above, the unsteady simulation is carried out using the dual time stepping scheme. The inner pseudo time iteration is conducted using the implicit line Gauss-Seidel iteration, accelerated by the local time step technique. The fluid-structural interaction procedure is outlined in Fig. 4.7 . On each physical time step, the computation starts from the previous flow field at time step *n*. When the grid is moving, the computation boundary is first deformed following the prescribed vibration. The mesh is then updated using the appropriate model mentioned in the previous section. The calculation is iterated with the pseudo time and is converged when the maximum residual is reached to the given

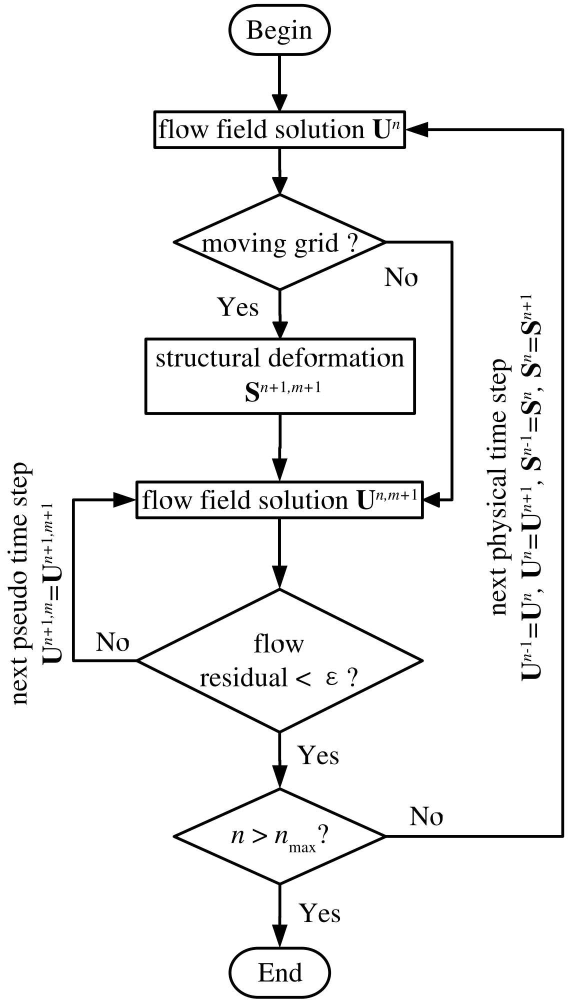

Begin
flow field solution $\mathbf{U}^n$
moving grid ?
No
Yes
structural deformation $\mathbf{S}^{n+1,m+1}$
flow field solution $\mathbf{U}^{n,m+1}$
next pseudo time step
$\mathbf{U}^{n+1,m}=\mathbf{U}^{n+1,m+1}$
No
flow residual < ε ?
next physical time step
$\mathbf{U}^{n-1}=\mathbf{U}^n$, $\mathbf{U}^n=\mathbf{U}^{n+1}$, $\mathbf{S}^{n-1}=\mathbf{S}^n$, $\mathbf{S}^n=\mathbf{S}^{n+1}$
Yes
No
$n > n_{\max}$?
Yes
End

Figure 4.7: Unsteady time marching procedure

convergence criteria ($\varepsilon$). Then the time marching for the current physical time step is completed and the updated mesh and flow field are saved. A new physical time marching step is repeated following the same procedure. The unsteady time marching is carried out until the specified maximum physical time step number is reached.

# Chapter 5

# Validation Calculation

In this chapter, the code is validated using several 2D cases for the turbulence modeling, parallel computation scalability, unsteady calculation capability and moving grid system. The turbulence modeling is validated with a subsonic flat plate boundary layer flow and an inlet-diffuser transonic flow. The parallel computation speedup scalability is validated with the flat plate boundary flow. The unsteady calculation capability is tested with the inlet-diffuser transonic flow. The moving grid system is tested on an forced oscillating airfoil.

## 5.1 Turbulence Model Validation

The first case is a subsonic flat plate turbulent boundary layer flow. The computation results are compared with the law of the wall. In the turbulent boundary layer on a flat plate without pressure gradient, the law of the wall is valid for the viscous layer, the buffer zone and the fully turbulent portions [125]. In terms of the non-dimensional velocity $u^+$ and the non-dimensional distance $y^+$ , the law of the wall is written as the following,

$$u^+ = \begin{cases} y^+ & \text{when } y^+ < 2\text{~}8 \text{ (viscous sublayer)} \\ \frac{1}{\kappa}\ln y^+ + B & \text{~}50 < y^+ < \text{~}500 \text{ (fully turbulent zone)} \end{cases} \tag{5.1}$$

where the non-dimensional $u^+$ and $y^+$ are defined as,

$$u^+ = \frac{u}{u_\tau} \tag{5.2}$$

$$y^+ = y\frac{u_\tau}{\nu} \tag{5.3}$$

$u$ is the velocity component parallel to the wall surface, $y$ is the distance to the wall, $\nu$ is dynamic viscosity, $u_\tau$ is the friction velocity given by

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{5.4}$$

$\tau_w$ is the wall shear stress.

The von Karman constants in Eq. (5.1) typically have values of $\kappa$ =0.4~0.41, $B$ =5.0~5.5. In the following comparison, $\kappa$= 0.4 and $B$ =5.5.

The computation mesh is distributed as 80 points uniformly allocated along the wall surface and 60 points allocated normal to the wall surface with a stretch factor of 1.1. The $y^+$ of the first cell center to the wall is kept under 0.2. The Reynolds number is $4\times10^6$ based on the plate length. The inlet Mach number is 0.5. The flow is subsonic at inlet and outlet. The upper boundary is zero gradient. The governing equation is solved implicitly with the Gauss-Seidel line relaxation method. The discretization scheme is the Zha CUSP2 scheme. The CFL number is set to be 100. The computed result is compared very well with the law of the wall as shown in Fig. 5.1 .

The second validation case for the Baldwin-Lomax turbulence model performance is a transonic inlet-diffuser flow, which was tested in wind tunnel [126]. The test section is designed to obtain a 2D flow condition. The flow enters the inlet diffuser under subsonic condition and accelerates to supersonic after the throat. By adjusting the back pressure level at outlet, different shock wave structures are obtained. The inlet diffuser has a height of $h_t$ =4.4 cm at the throat and a total length of $12.6h_t$. The inlet height is $h_{in} = 1.4h_t$. Fig. 5.2 is the mesh used for the calculation. The Roe scheme is used for the inviscid flux

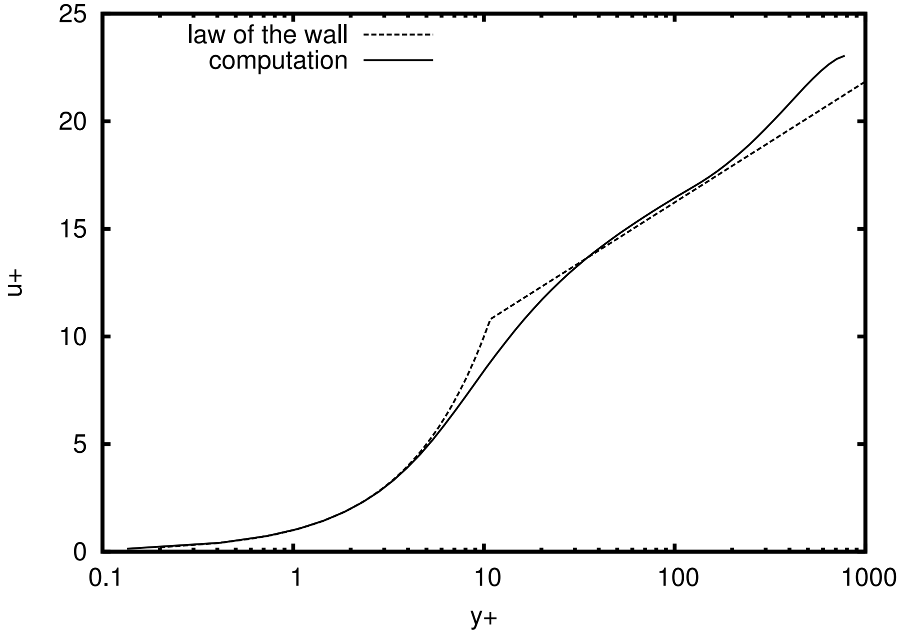


Figure 5.1: Computed velocity profile of the flat plate boundary layer flow compared with the law of the wall

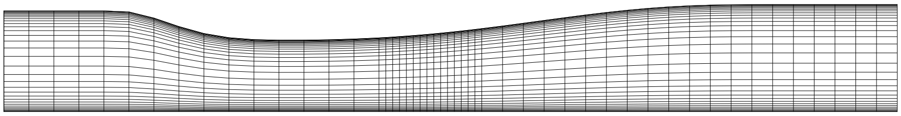


Figure 5.2: The transonic inlet-diffuser 2D mesh

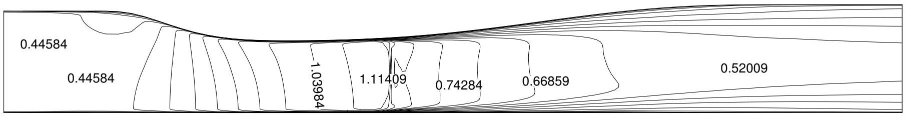


Figure 5.3: Mach number contours of the transonic inlet-diffuser with back pressure $0.83p_t$

calculation. The mesh size is 100×60. The mesh is uniformly distributed horizontally, except that a refinement is applied in the throat region, where a normal shock interacts with the turbulent boundary layer. The mesh is clustered close to upper and lower walls vertically with a stretch factor of 1.1. The maximum $y^+$ of the first cell center close to the wall is below 3. The Reynolds number is $4.34\times10^5$ based on the throat height. Uniform total pressure and total temperature are fixed at the inlet. The inlet Mach number in the computation result is 0.45. At the outlet, the static pressure is fixed as $p_{out}/p_t$=0.83. A CFL of 5 is used in the Gauss-Seidel iterations.

Figure 5.3 shows the computed Mach number contours, where the shock wave is clearly captured. The upper wall static pressure distribution is compared with the experimental data in Fig. 5.4 . A very good agreement is achieved.

## 5.2 Parallel Computation Speedup Scalability

When applying the MPI protocol for parallel computation, ideally the computation wall clock time will be reduced linearly with the number of processors employed in the computation. However the performance of the parallel computation is influenced by the implementation details. The data communication between subdomains will take CPU time, and the communication time increases with the exchanging data size. The domain decomposition will affect the convergence rate of the implicit Gauss-Seidel iteration. The inversion of a block tri-diagonal matrix along a grid line in single processor computation is split into multiple short grid lines due to the domain partitioning. At the domain partitioning boundary, the information from the neighboring grid lines is simply treated as zero. This

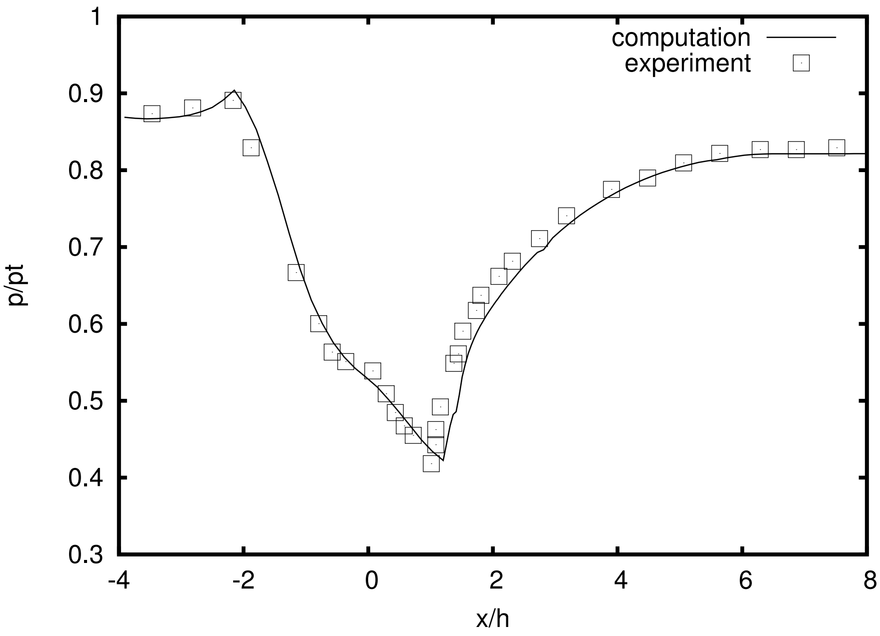


Figure 5.4: Upper wall pressure distribution of the transonic inlet-diffuser with back pressure $0.83p_t$

treatment is approximate and will affect the convergence rate compared with the single grid line solution. However, the accuracy of the converged results will not be affected. The extra cost can be minimized by optimizing the size and the structure of the interface message and the topology of the decomposition.

The parallel computation performance is tested on the flat plate boundary layer flow. Similar to the calculation above, a 80×60 rectangular mesh is applied. The inlet Mach number is 0.5. However, the calculation is carried out for a laminar flow condition. The Reynolds number based on the plate length is $4\times10^4$. The solutions are achieved with 1, 2, 4, 6 and 10 processors respectively. The Roe scheme is used for the inviscid flux computation. There are up to 5 machines (two CPUs on each machine) used in the calculation simultaneously. These machines are identical in hard ware configuration and operating system (Red Hat Linux 7.1). The solution is considered as converged when the maximum residual is below $1\times10^{-9}$. The iteration history and the final wall clock time are recorded for comparison. After each time marching iteration, the flow variables are exchanged be-

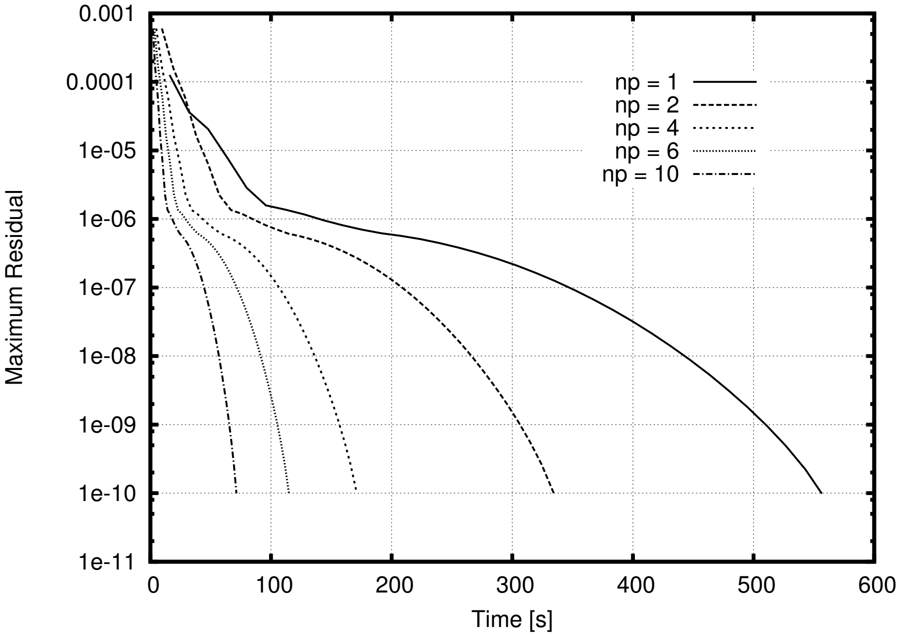


Figure 5.5: Boundary layer flow computation iteration histories

tween the two neighboring sub-domains. To minimize the MPI communication time, the computation domain is divided almost evenly in horizontal direction ($x$) only. Therefore, the communication only needed to be carried out in one direction and the interface vertical width of 60 cells is smaller than the horizontal width of 80 cells. The data on the partitioning boundary to be exchanged are not non-contiguously distributed in memory. They are arranged as a single block of derived date type and transfered in a single communication operation, which saves the communication time.

The computations with 1, 2, 4, 6, 10 processors converge in 1743, 1746, 1753, 1760, 1775 steps respectively. The domain decomposition increases the total iteration number slightly. Fig. 5.5 plots the iteration histories for all computation cases. The iteration history curves have similar shapes. The total computation wall clock time decreases dramatically with the increase of the processor number (np). The parallel computation speedup scalability is shown in Fig. 5.6 , where the computation speedup calculated based on Eq. (4.67) is plotted versus the number of processors. The speedup increases almost linearly with the number of processors. The discrepancy between the CFD results and the ideal linear-

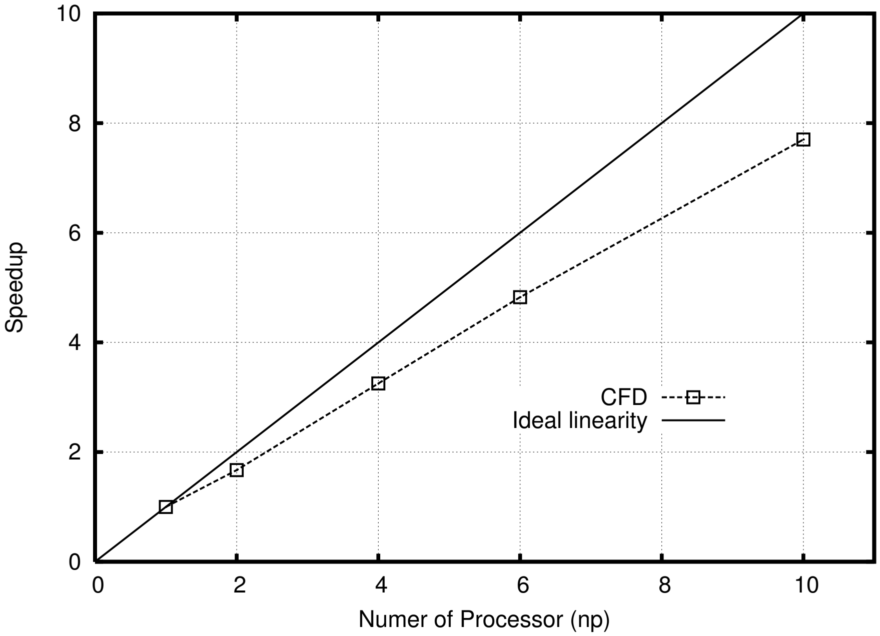


Figure 5.6: Parallel computation speedup scalability in boundary layer flow problem

ity indicates the time used for message communication also increases with the number of processors.

## 5.3 Unsteady Capability

The unsteady simulation capability of the solver is validated using two 2D cases. The first is the self-excited shock wave oscillation in the transonic inlet diffuser. The second is a forced oscillating airfoil. The flow interacts with the airfoil structure movement of forced vibration.

### 5.3.1 Transonic inlet-diffuser shock wave oscillation

The frequency characteristics of the shock wave oscillation in the same transonic inlet-diffuser discussed above is experimentally measured in reference [126]. The shock wave oscillation results from the interaction between the turbulent boundary layer and the shock wave.

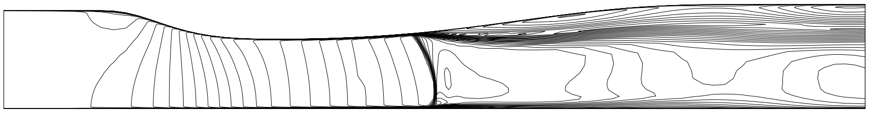


Figure 5.7: Transonic inlet-diffuser Mach contours with back pressure $0.72p_t$

The mesh is similar to that in Fig. 5.2. The mesh size in the unsteady simulation is $130(x)\times60(y)$. The near wall maximum $y^+$ is below 3. The Roe scheme is applied to evaluate the control volume inter surface flux. The total pressure $p_t$, total temperature $T_t$ and flow angle $\alpha$ are specified at the inlet. The no-slip adiabatic boundary condition is applied at top and bottom walls. The back pressure level is set as $p_{out}/p_t$ =0.72 as the experiment. The Reynolds number based on the throat height is $4.3896\times10^5$. The static pressure is fixed at the outlet. The physical time step is set as 0.28272 ms, which is about 7% of the shock oscillation cycle. The Gauss-Seidel iteration is swept forward and backward within each physical time step with a CFL of 5.0 for the pseudo time step. The converged solution for each physical time step is obtained after 50 pseudo time step iterations. The obtained inlet Mach number is 0.458.

As shown in Fig. 5.7 . The shock wave is clearly captured. Compared with the shock wave structure in Fig. 5.3, the shock in the case of $p_{out}/p_t = 0.72$ is more downstream. The flow is separated after the shock, which brings high unsteadiness to the flowfield. The location of the shock wave moves back and forth downstream of the throat region. The flow field parameters, including the shock wave location and static pressure, vary periodically with time in the region after the shock wave. Fig. 5.8 shows the pressure oscillation history on the upper wall at outlet. More clear frequency information is revealed using the Fourier transform technique. The static pressure frequency spectrum is shown in Fig. 5.9 on the left compared with experiment measurement based on the shock wave motion [126] on the right. The computed frequency spectrum peak is at 250 Hz, which is close to the experiment measurement frequency, 200 Hz. However, the computation has multiple peaks while the experiment only has one peak. The reason for the discrepancy between the computational

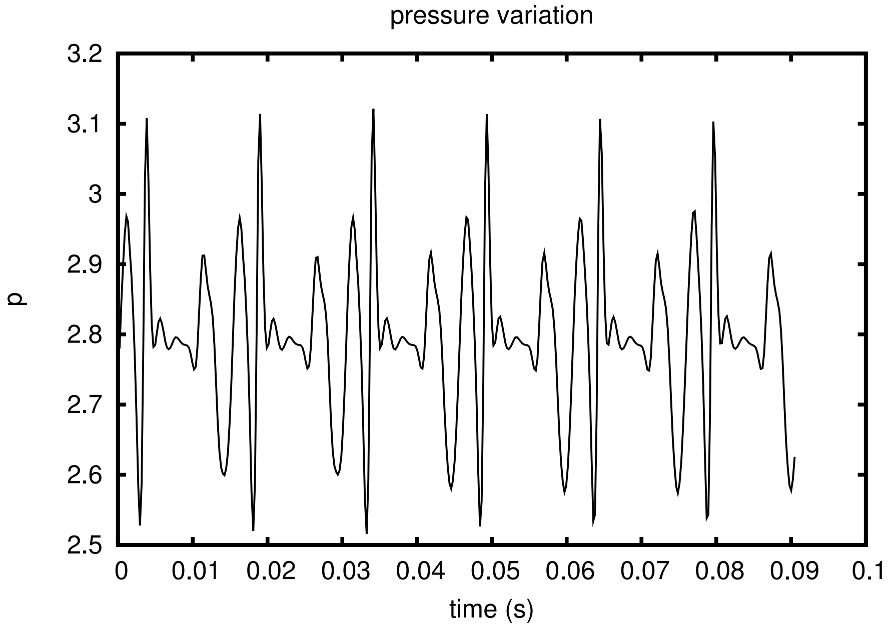


Figure 5.8: Transonic inlet diffuser outlet upper wall pressure variation history with back pressure $0.72p_t$

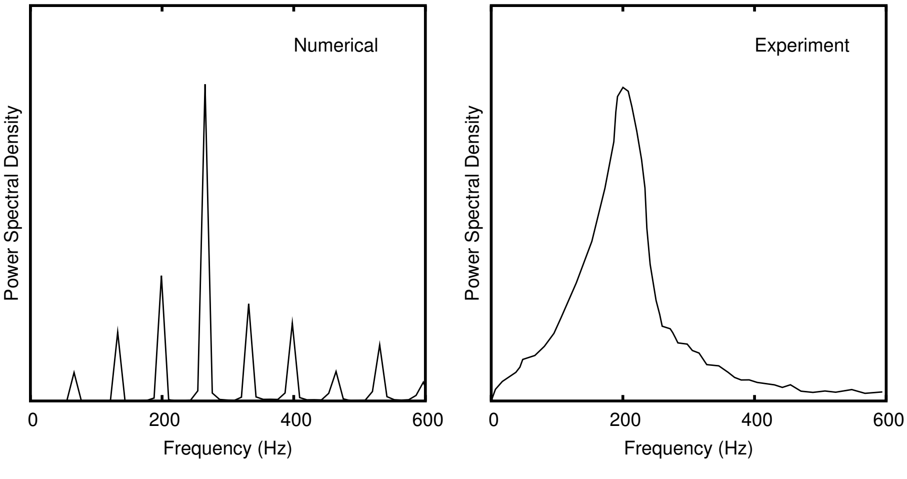


Figure 5.9: Transonic inlet-diffuser outlet upper wall unsteady pressure spectrum with back pressure $0.72p_t$

and experimental results may be the inadequacy of the turbulence model, which will affect the boundary layer thickness and hence the shock motion.

### 5.3.2 Forced oscillation of NACA 64A010 airfoil

In the second unsteady validation case, the flow around a forced pitching airfoil is calculated. The airfoil is NACA 64A0101 and the experiment measurement is reported by Davis [127]. In the experiment, the airfoil is forced in pitch about its quarter chord sinusoidally. The airfoil oscillation is defined by function of the time dependent variation of its angle of attack,

$$\alpha(t) = \alpha_0 + \alpha_A \sin(\omega t) \tag{5.5}$$

where $\alpha\,(t)$ is the time dependent angle of tack.$\alpha_0$ are $\alpha_A$ are the mean and the amplitude of the oscillating angle respectively, $\omega$ is the angular frequency which is directly related to the reduced frequency

$$k_c = \frac{\omega C}{2U_\infty} \tag{5.6}$$

where $C$ is the airfoil chord, and $U_\infty$ is the free-stream velocity.

An O-type mesh consisting of 280×65 cells is employed for calculation. When the airfoil is oscillating, the moving grid system is dynamically generated based on a two-layer model. The fine mesh zone or non-deforming part of the mesh is depicted in Fig. 5.10 . This mesh layer oscillates with the airfoil surface. The outer layer of the mesh is linearly interpolated between the fixed far field outer boundary and the outermost grid line on the fine mesh zone every time the the airfoil oscillates to a new position. The smallest grid spacing attached to the airfoil surface is about $8.0\times10^{-6}$ chord in radial direction. To be consistent with the experiment, the following primary parameters are employed in the unsteady calculation: $\alpha_0$ = 0, $\alpha_A = 1.01°$, Reynolds number (based on chord), Re $= 1.256\times10^7$, free-stream Mach number, $M_\infty$ = 0.8, reduced frequency, $k_c = 0.202$.

The inlet boundary is subsonic with velocity specified and the outlet is subsonic with pressure specified. The computation begins with the stationary airfoil model with a dimen-

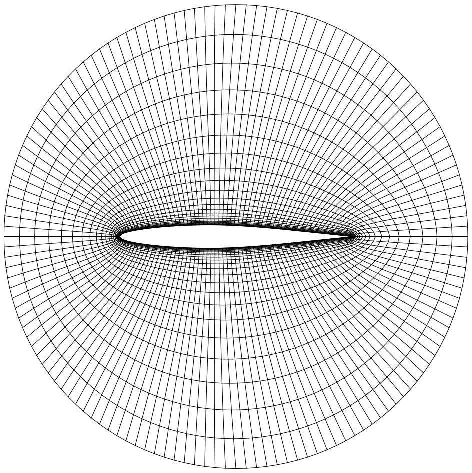


Figure 5.10: O-type mesh around the NACA 64A010 airfoil

sionless time step $\Delta t = 0.1$. The typical Mach contours at an arbitrary time is plotted in Fig. 5.11 . There are two shock wave observed on the upper and lower surfaces of the airfoil. They move back and forth during the airfoil oscillation. From the stationary airfoil model, it can be easily switched to the forced pitching airfoil model. The experimental results given by Davis [127] are reproduced using rectangle symbol in Figs. 5.12 and 5.13 . The computation result is also plotted using a solid line in the figures for comparison. Fig. 5.12 plots the history of the computed lift coefficient varying with the angle of attack. The computation agrees quite well with the experiment. Fig. 5.13 plots the history of the computed moment coefficient varying with the angle of attack. The agreement of the moment coefficient is not as good as that of the lift coefficient. However, the agreement in the current results is better than the recent result computed by McMullen et al. in 2002 [128]. The discrepancy between the computation and the experiment in the moment coefficient may be caused by the inadequacy of the turbulence modeling, which may not predict the surface friction accurately.

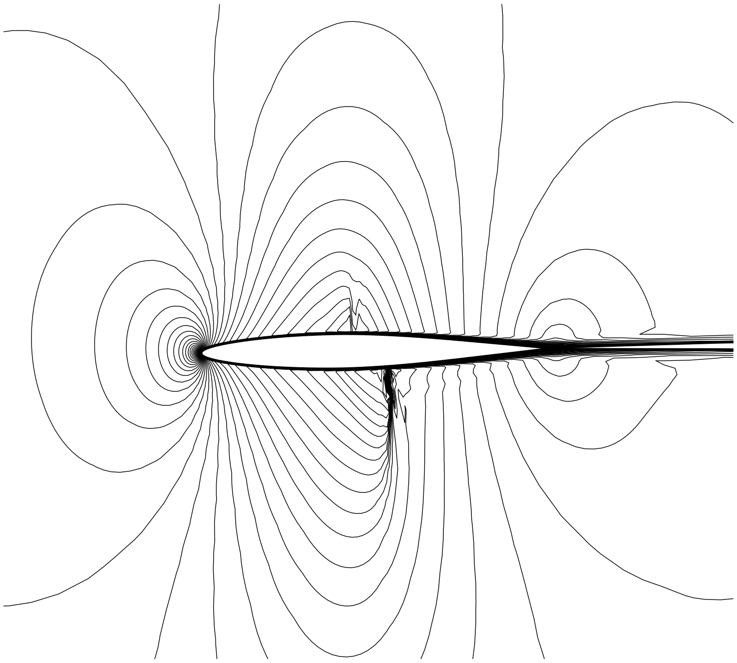


Figure 5.11: Instantaneous Mach contours around the forced pitching NACA 64A010 airfoil

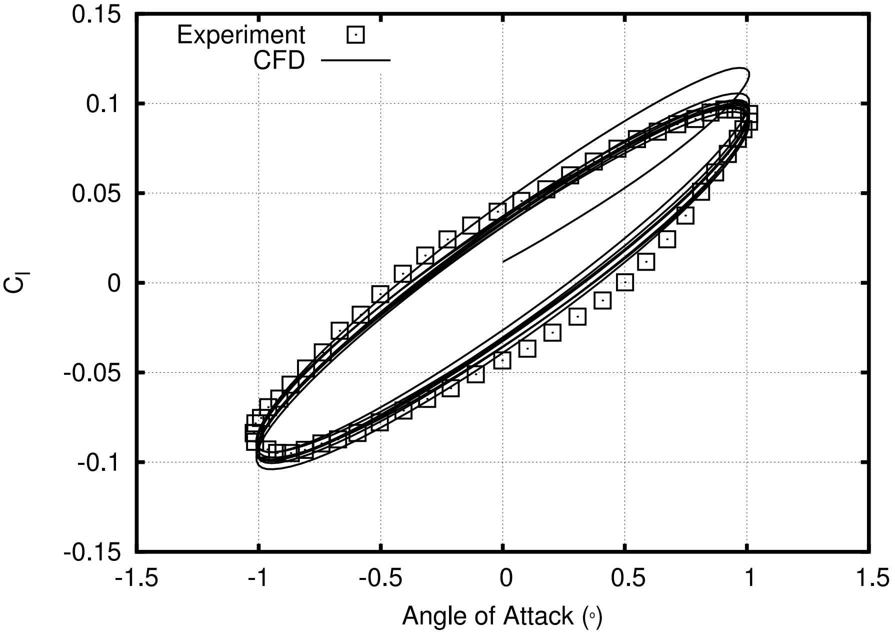


Figure 5.12: Comparison of computed lift coefficient with experimental data for the forced pitching airfoil

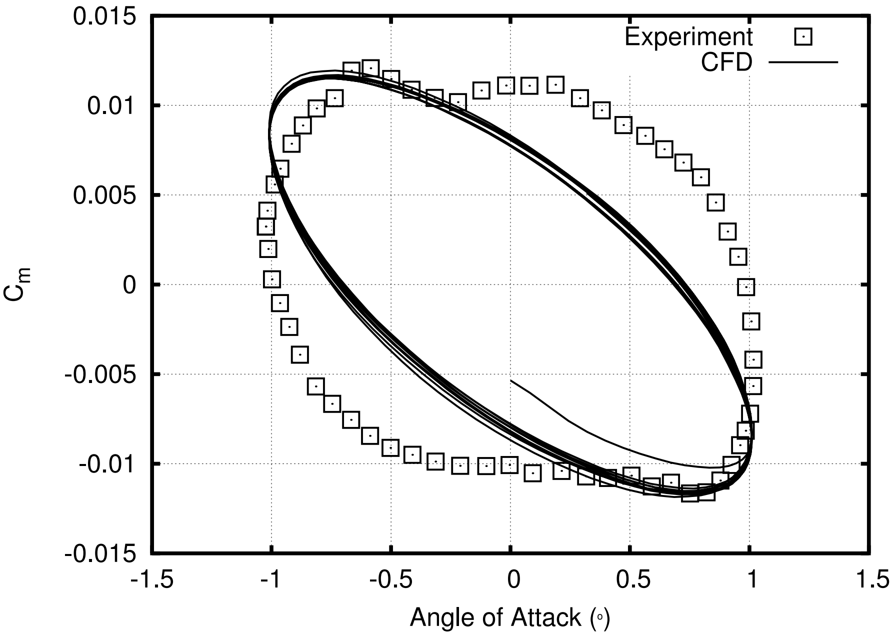


Figure 5.13: Comparison of computed moment coefficient with experimental data for the forced pitching airfoil

# Chapter 6

# Performance of Zha CUSP Schemes

The accuracy of the newly developed Zha CUSP schemes are compared with other popular upwind schemes including the Roe scheme, the Liou's AUSM$^+$ scheme, the van Leer scheme and the van Leer-Hänel scheme. The Zha CUSP schemes are tested in both 2D and 3D cases. These tests show that the new schemes are accurate, robust and efficient.

## 6.1 Zha CUSP Scheme in 2D Cases

The Zha CUSP scheme is tested in several 2D cases including a flat plate supersonic boundary layer flow, a transonic nozzle flow with oblique shock waves and reflections and the transonic inlet diffuser flow with shock wave-turbulent boundary layer interaction.

### 6.1.1 Wall boundary layer

To examine the numerical dissipation of the Zha CUSP scheme, a laminar supersonic boundary layer on an adiabatic flat plate is calculated using first order accuracy. The incoming Mach number is 2.0. The Reynolds number based on the length of the flat plate is 40000. The Prandtl number of 1.0 is used in order to compare the numerical solutions with the analytical solution. The baseline mesh size is 80×60 in the direction along the plate and normal to the plate respectively.

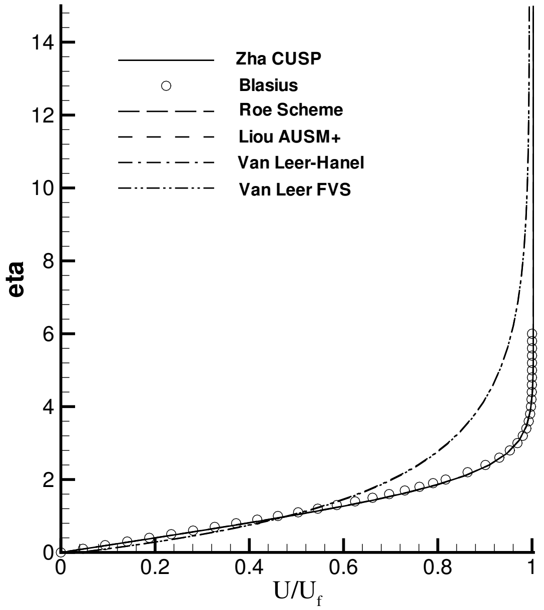


Figure 6.1: Computed velocity profiles of the laminar boundary layer using 1st order schemes

Fig.6.1 is the comparison between the computed velocity profiles and the Blasius solution. The solutions of the Zha CUSP scheme, Roe scheme, and AUSM$^+$ scheme agree very well with the analytical solution. The van Leer scheme significantly thickens the boundary layer. The van Leer-Hänel scheme does not improve the velocity profile.

Fig.6.2 is the comparison between the computed temperature profiles and the Blasius solution. Again, the Zha CUSP scheme, Roe scheme, and AUSM$^+$ scheme accurately predict the temperature profiles and the computed solutions basically go through the analytical solution. Both the van Leer scheme and the van Leer-Hänel scheme significantly thicken the thermal boundary layer similarly to the velocity profiles.

Table 6.1 shows the wall temperatures predicted by all the schemes using the baseline mesh and refined mesh. The predicted temperature value by the van Leer scheme has a large error. The van Leer- Hänel scheme does predict the wall temperature accurately even though the overall profile is nearly as poor as that predicted by the van Leer scheme. The

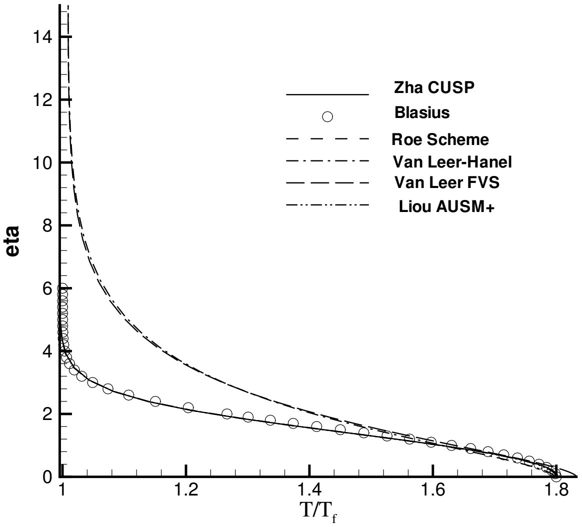


Figure 6.2: Computed temperature profiles of the laminar boundary layer using 1st order schemes

Table 6.1: Computed dimensionless wall temperature using first order schemes with the baseline mesh and refined mesh

| Scheme | $T_{\text{wall}}$, Mesh 80 × 60 | $T_{\text{wall}}$, Mesh 160 × 120 |
|---|---|---|
| Blasius analytical solution | 1.8000 | 1.8000 |
| Zha CUSP | 1.8025 | 1.8018 |
| Roe scheme | 1.8002 | 1.7996 |
| Liou's AUSM$^+$ | 1.8000 | 1.8000 |
| van Leer | 1.8328 | 1.8333 |
| van Leer-Hänel | 1.7970 | 1.7996 |

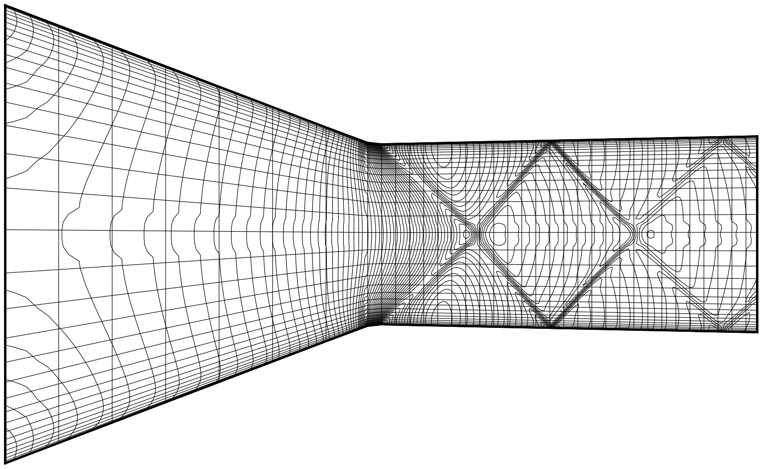


Figure 6.3: Computed Mach number contours using the Zha CUSP scheme

Zha CUSP scheme, Roe scheme and AUSM$^+$ scheme all predict the temperature accurately.

All the results mentioned above are converged based on mesh size. The wall temperatures using the refined mesh of 160×120 are also given in table 6.1. There is little difference between the results of the baseline mesh and the refined mesh. The refined mesh does not help to reduce the large numerical dissipation of the van Leer scheme. When the 2nd order schemes are used, both the velocity and temperature profiles of the van Leer scheme and the van Leer- Hänel are improved (not shown).

### 6.1.2 Transonic converging-diverging nozzle

To examine the performance of the Zha CUSP scheme in two-dimensional flow and the capability to capture the shock waves which do not align with the mesh lines, a transonic converging-diverging nozzle is calculated as inviscid flow. The nozzle was designed and tested at NASA and was named as Nozzle A1 [129]. Third order accuracy of MUSCL type differencing is used to evaluate the inviscid flux with the Minmod limiter. Fig. 6.3 is the computed Mach number contour using the Zha CUSP scheme with the mesh size of

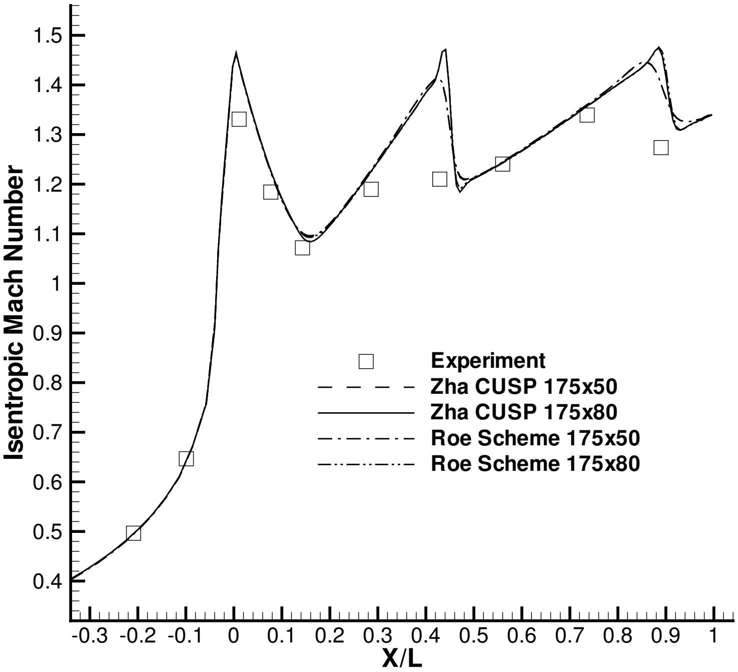


Figure 6.4: Adiabatic Mach number distribution computed on the wall surface of the nozzle

175 × 80. In the axial direction, there are 140 mesh points distributed downstream of the nozzle throat, where the oblique shock waves are located. The grid is clustered near the wall. For clarity, the coarsened mesh is drawn as the background with the Mach contours to show the relative orientation of the shock waves and the mesh lines. The nozzle is symmetric about the centerline. Hence only the upper half of the nozzle is calculated. The upper boundary uses the slip wall boundary condition and the lower boundary of the center line uses the symmetric boundary condition.

As indicated by the wall surface isentropic Mach number distribution shown in Fig. 6.4 , the flow is subsonic at the inlet with the Mach number about 0.22 and is accelerated to sonic at the throat, and then reaches supersonic with Mach number about 1.35 at the exit. Fig. 6.3 shows that right after throat, an expansion fan emanates from the wall and accelerates the flow to reach the peak Mach number about 1.5. Due to the sharp throat turning, an oblique shock appears immediately downstream of the expansion fan to turn the flow to axial direction. The two oblique shocks intersect at the centerline, go through

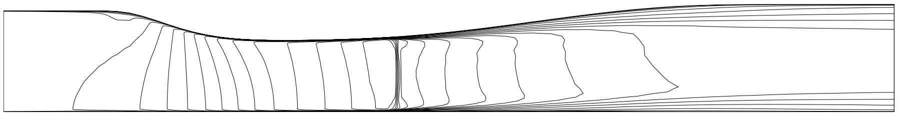


Figure 6.5: Computed Mach number contours using the Zha CUSP scheme with back pressure $0.83p_t$

each other, hit the wall on the other side, and then reflect from the wall. Such shock patten is repeated to the exit and the shock strength is weakened with the flow going downstream. As shown in Fig. 6.4, the isentropic Mach number distributions predicted by the Zha CUSP scheme and the Roe scheme agree fairly well with the experiment. The Zha CUSP scheme and the Roe scheme have virtually indistinguishable results.

The mesh refinement study indicates that the mesh resolution in the axial direction does not affect the shock resolution much. The axial mesh size of 280 downstream of the throat yields only slightly better shock resolution than the size of 70. However, the mesh size in the vertical direction dramatically changes the shock resolution. The mesh size of 80 in the vertical direction yields much better resolution than the mesh size of 50. This can be seen from the isentropic Mach number in Fig. 6.3, which shows that the mesh size of 175×80 generates much sharper profiles than those of the mesh 175×50 for the first and second shock reflections.

For this transonic nozzle with the mesh size 175×80 on an Intel Xeon 1.7 GHz processor, the CPU time per time step per node to calculate the inviscid flux is $2.5871\times10^{-6}$s for the Zha CUSP scheme, which is about 25% of the CPU time of $1.0284\times10^{-5}$s used for the Roe scheme. This is a significant CPU time reduction.

### 6.1.3 Transonic inlet-diffuser

To examine the performance of the Zha CUSP scheme for shock wave-turbulent boundary layer interaction, a transonic inlet-diffuser [126] is calculated as shown by the Mach number contours in Fig. 6.5 , which has the exit back pressure $p_{out}/p_t = 0.83$. The Reynolds

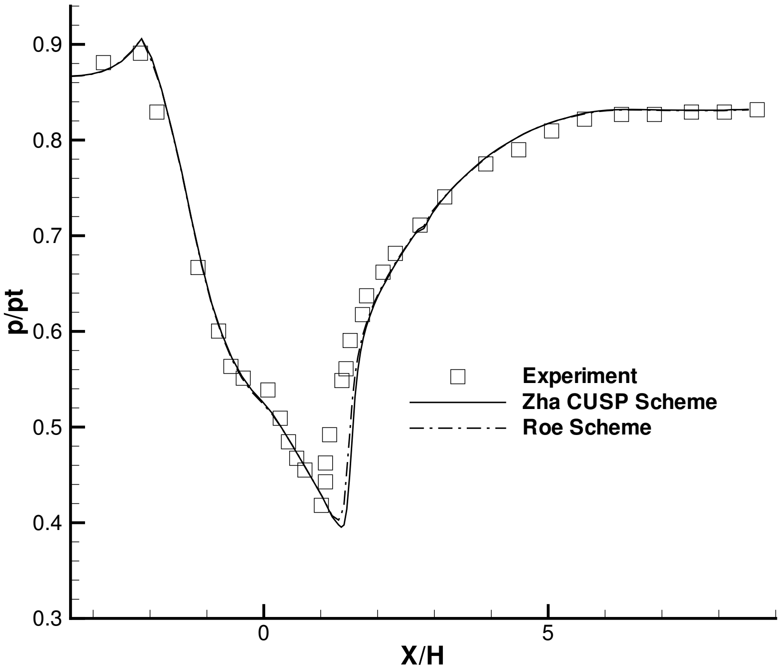


Figure 6.6: Static pressure distribution computed on the upper surface of the inlet-diffuser with back pressure $0.83p_t$

number based on the throat height is $4.38 \times 10^5$. The Baldwin-Lomax algebraic turbulence model [56] is used. The third order accurate MUSCL-type differencing with the Minmod limiter is used for the inviscid fluxes and the second order central differencing is used for the viscous terms.

A normal shock is located downstream of the throat as shown in Fig. 6.5. No flow separation is generated under this back pressure. The baseline mesh size is $100 \times 60$. When $y^+$ is held as constant and the mesh is refined in both the horizontal and vertical direction, the results have little variation. The inlet-diffuser results presented in this section are based on the mesh size $100 \times 120$. The mesh in the horizontal direction is clustered around the shock location to better resolve the shock profile.

Fig. 6.6 is the comparison of the upper wall surface pressure between the experiment and the computation. The agreement is very good except that the computation predicts the shock location a little downstream of the experimental shock location and the shock

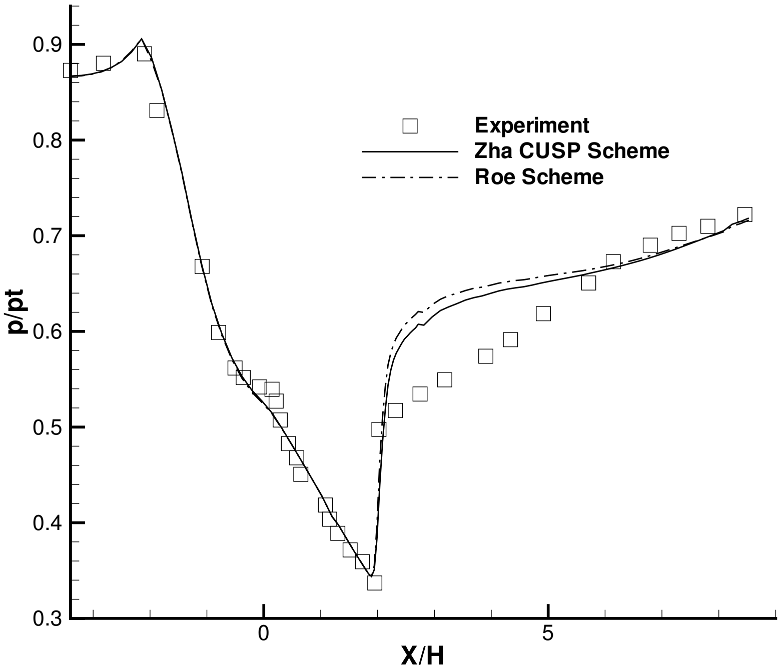
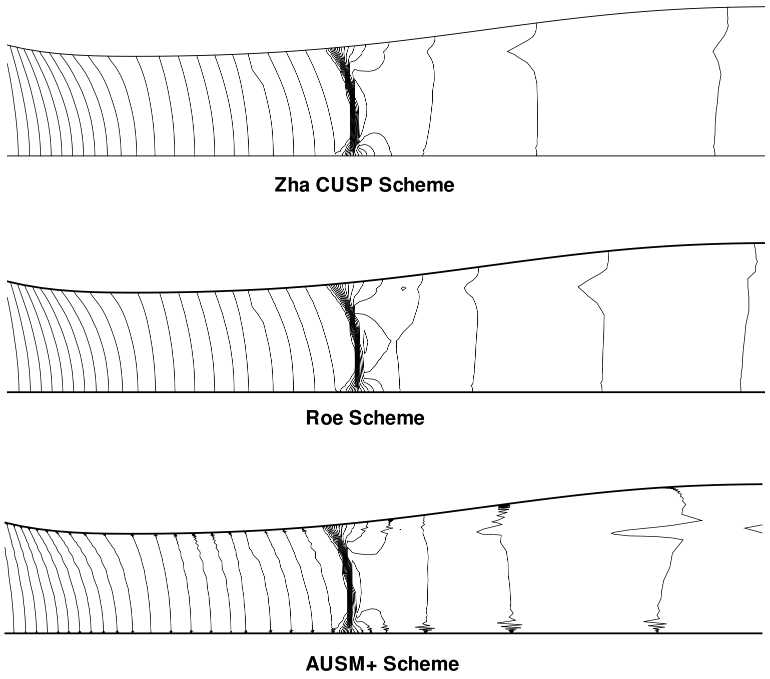


Figure 6.7: Static pressure distribution computed on the upper surface of the inlet-diffuser with back pressure $0.72p_t$

strength a little too strong. It is found that the shock profile is sensitive to the $y^+$. The $y^+$ value of $2.2\times10^{-4}$, $7\times10^{-6}$ are tested. The smaller $y^+$ yields a little closer shock location to the experiment. The results shown in Figs. 6.5 and 6.6 have the $y^+$ value of $2\times10^{-4}$. The small $y^+$ effect is believed due to the first order extrapolation of the pressure on wall surface instead of the requirement of the turbulence modeling. In the region with no shock, the first order pressure extrapolation on the wall is insensitive to the distance of the first cell to the wall, while in the shock region it is sensitive due to the large streamwise gradient. As indicated in Fig. 6.6, the Roe scheme predicts the shock location slightly closer to the experiment than the Zha CUSP scheme.

When the back pressure is reduced to $p_{out}/p_t = 0.72$. The normal shock is stronger and the flow separation is induced. The same mesh as the previous case is used for this case. Fig. 6.7 is the predicted pressure distribution compared with the experiment. Both the Zha CUSP scheme and the Roe scheme predict the shock location accurately, but the shock



Zha CUSP Scheme

Roe Scheme

AUSM+ Scheme

Figure 6.8: Comparison of computed pressure contours using the Zha CUSP scheme, Roe Scheme, and Liou's AUSM$^+$ scheme with back pressure $0.72p_t$

strength predicted is too strong. However, the Zha CUSP scheme has the pressure profile in the separation region downstream of the shock noticeably closer to the experiment than that predicted by the Roe scheme.

It should be pointed out that the turbulence modeling is a critical factor for the prediction accuracy of the shock wave-turbulent boundary layer interaction. Hence the discrepancy between the calculation and experiment shown above is only partially attributed to the different discretization schemes.

Fig. 6.8 is the pressure contours computed using $p_{out}/p_t = 0.72$ with the Zha CUSP scheme, Roe Scheme, and Liou's AUSM$^+$ scheme. A curved $\lambda$ shock is formed due to the shock wave-turbulent boundary layer interaction. The shape of the Mach contours of the Zha CUSP scheme and the Roe scheme are very much alike. The contours computed by the AUSM$^+$ scheme has significant oscillations near the wall.

## 6.2 Zha CUSP2 Scheme in 3D Cases

The Zha CUSP2 scheme is tested in 3D cases for a transonic nozzle with circular-to-rectangular cross-section, a transonic duct with shock wave-turbulent boundary layer interaction, and a subsonic compressor cascade.

### 6.2.1 Circular-to-rectangular nozzle

A transonic nozzle with circular-to-rectangular cross section tested at NASA [130] is calculated. The transition duct connects the axisymmetric engine to the non-axisymmetric nozzle through a smooth progression of geometrically similar cross sections. There is no swirl flow at the inlet. The Reynolds number based on the inlet diameter is Re=$7.7058 \times 10^5$. Due to its symmetric structure, only a quarter of the nozzle geometry is computed as shown in Fig. 6.9 . Two symmetric boundaries are located at the bottom and back sides. The wall is divided into two parts to generate the H-type mesh. For clarity, every two other grid line is omitted in each direction in the plot. The baseline mesh size is $100\times50\times50$ and is highly stretched near the wall. No shock wave exists in the flow field. The total pressure, total temperature and flow angles are fixed at the inlet boundary. Because of the supersonic flow at the outlet, the zero-gradient boundary condition is used at the nozzle exit. No slip and adiabatic wall conditions are used for the wall. The optimum CFL number used is 200.

Fig. 6.10 shows the contour lines of Mach number on the bottom symmetric plane for the Zha CUSP2 scheme. The flow accelerates from subsonic at the inlet, reaches sonic at the nozzle throat, and becomes supersonic at the exit. The top wall and side wall static pressure distributions are compared with experimental results [130] in Fig. 6.11 and Fig. 6.12 respectively. Good agreement is obtained.

The Zha CUSP2 scheme is also more CPU efficient than the Roe scheme in the 3D computation. On an Intel Xeon 1.7G Hz processor, the CPU time used to compute the inviscid flux per step per node is $1.84\times10^{-5}$s for the Zha CUSP2 scheme and $2.9723\times10^{-5}$s for the Roe scheme . The Zha CUSP2 scheme is about 40 percent more efficient.

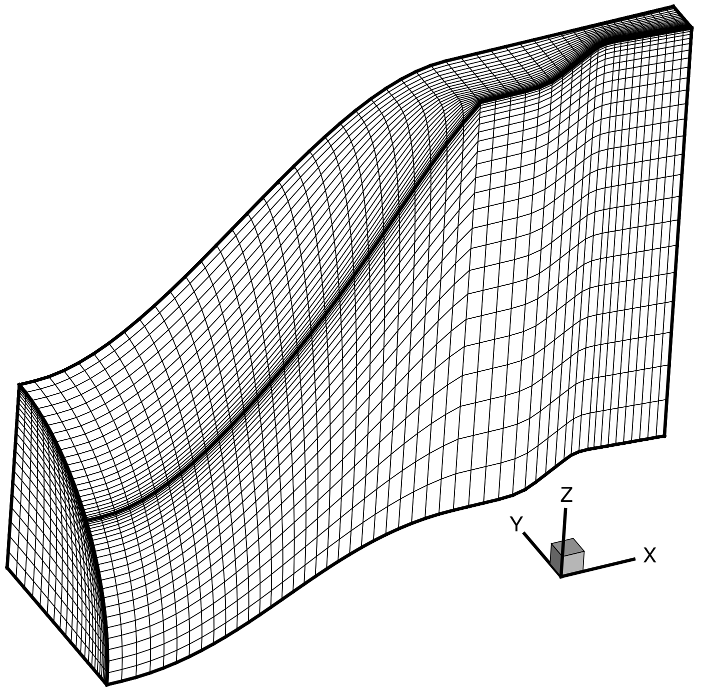


Figure 6.9: The mesh of the nozzle with circular-to-rectangular cross section

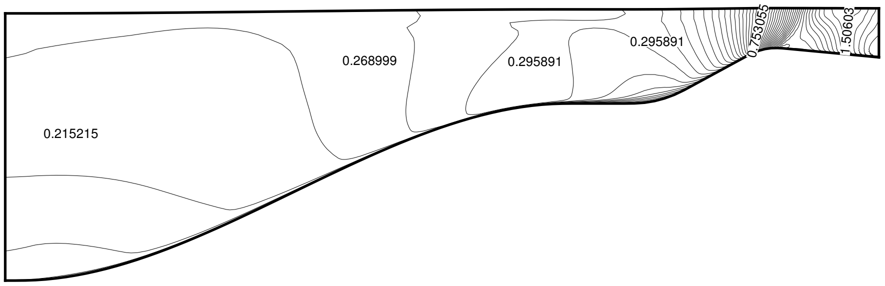


Figure 6.10: Mach number contours of the nozzle with circular-to-rectangular cross section

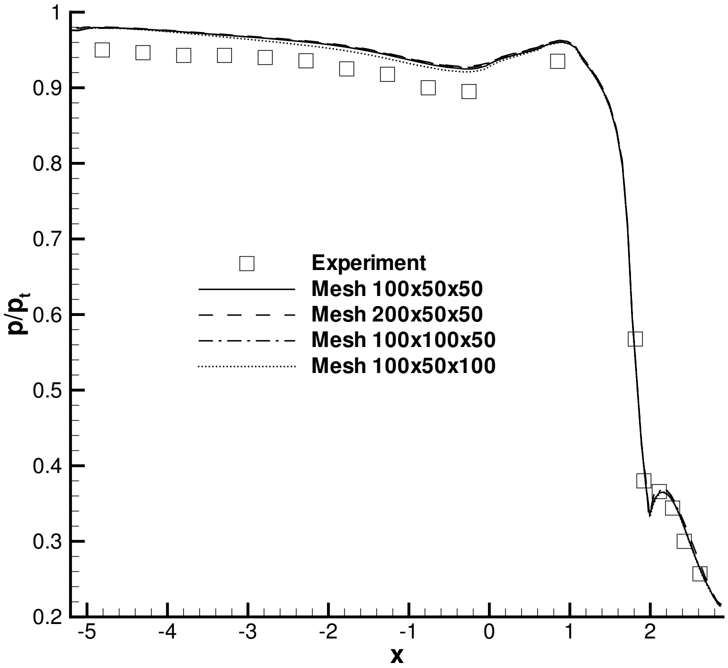


Figure 6.11: Top wall surface pressure distributions of the circular-to-rectangular nozzle compared with the experiment

Mesh refinement study is carried out by doubling mesh density in $\xi$, $\eta$ and $\zeta$ directions respectively. The computation results with mesh size 200×50×50, 100×100×50 and 100×50×100 are also plotted in Figs. 6.11 and 6.12. The same mesh height is kept on the first inner cell close to the wall boundary for the mesh refinement. The mesh refinement gives about the same results as the original baseline mesh as shown in Figs. 6.11 and 6.12, which indicate that the solution is converged based on the mesh refinement.

### 6.2.2 3D compressor cascade

The second 3D case is to calculate a 3D subsonic compressor cascade tested at NASA GRC [131]. The cascade has a chord length of 8.89 cm, a stagger angle of 60° and a solidity of 1.52. The flow under two incidences (0° and 10°) are measured in the experiment. The low incidence ( 0°) case is calculated here for its steady state results. More detailed steady and unsteady simulation under the high incidence, 10° is carried out in Chapter 7.

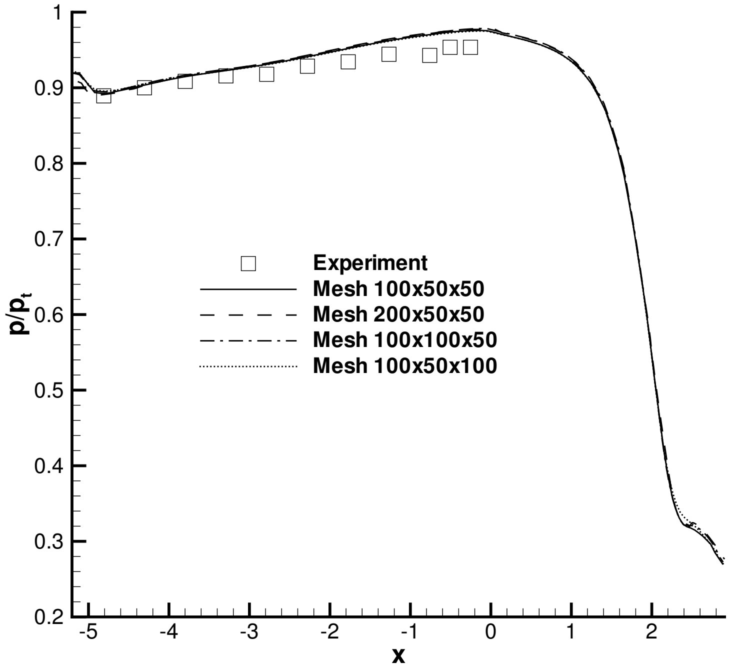


Figure 6.12: Side wall surface pressure distributions of the circular-to-rectangular nozzle compared with the experiment

Even though the geometry incidence is set to 0° in the experiment for the low incidence case, the actual incidence angle is considered to be 1° to 1.5° higher, due to the side wall boundary layer effect. In the current numerical simulation, the incidence angle is set as 1.5°. The Reynolds number based the chord length is $9.67\times10^5$. As shown in Fig. 6.13 , a mesh of $100(x)\times60(y)\times60(z)$ is used in the computation. In the streamwise direction, the numbers of mesh cells allocated for the upstream, wall surface and downstream are 15, 70 and 15 respectively. For clarity, every other mesh line is omitted in the plot.

The simulation includes the top and bottom end walls, where the cascade airfoil shape is very different from the one in the midspan as shown in Fig. 6.14 . The computation starts an initial flow field with zero velocity. The total pressure, total temperature and flow angles are fixed at the inlet. The static pressure is specified at the outlet. No slip and adiabatic wall boundary conditions are applied on blade surfaces, top and bottom end walls. Periodic boundary conditions are applied upstream and downstream of the blade in pitch direction.

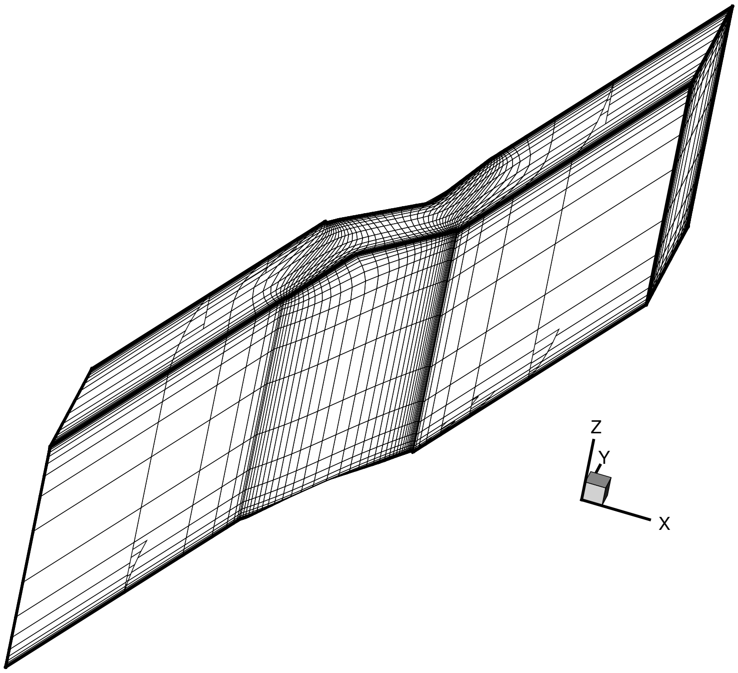
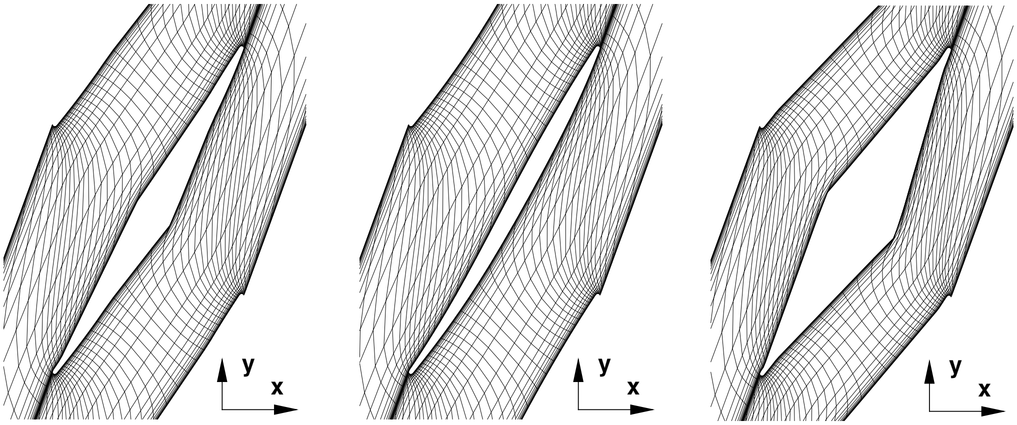


Figure 6.13: 3D cascade mesh



bottom mid-span top

Figure 6.14: Cascade mesh on the bottom, mid-span, and top planes

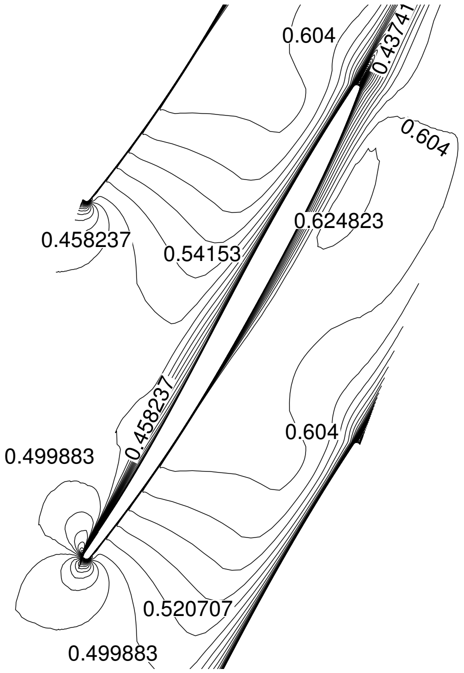


Figure 6.15: Cascade Mach contours on the midspan plane at low incidence

Fig. 6.15 shows the Mach number contours based on the Zha CUSP2 scheme on the mid-span plane. Fig. 6.16 shows that the computed surface pressure ($C_p = \frac{p_\infty - p}{\rho_\infty U_\infty^2}$) distribution agrees very well with the experiment [131]. The reference pressure $p_\infty$, density $\rho_\infty$ and velocity $U_\infty$ are the average values at the inlet. $p$ is the local static pressure on the blade surface. The result of the Zha CUSP2 scheme is also virtually identical to that of the Roe scheme.

### 6.2.3 Transonic channel flow

The last 3D case is a transonic channel flow with shock wave-turbulent boundary-layer interaction and is studied experimentally in reference [132]. The test section of the transonic channel has an entrance height of 100 mm and a width of 120 mm. It is composed of a straight top wall, two straight side walls. A bump with varying shape in span-wise direction is mounted on the bottom wall. The boundary conditions are to fix the total pressure, total

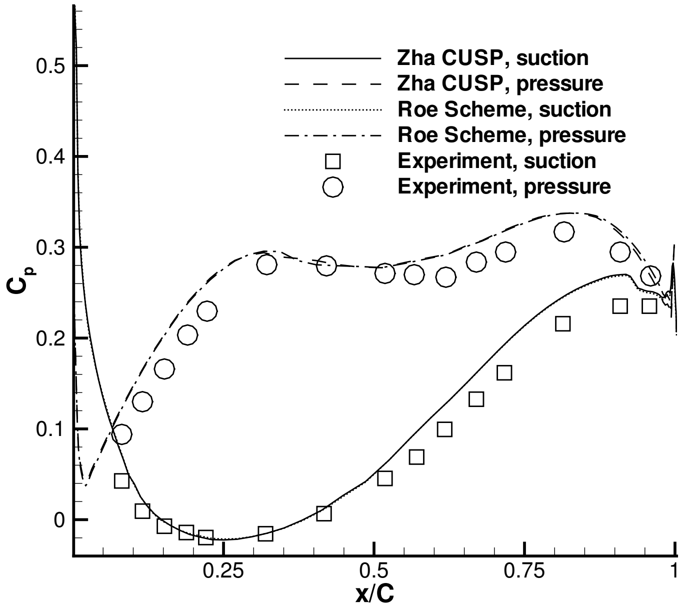


Figure 6.16: Cascade midspan plane surface pressure coefficient distributions compared with experiment measurement

temperature and flow angles at the inlet and static pressure at the outlet. No slip adiabatic wall boundary conditions are used on the walls.

In the present computation, the inlet Mach number is about 0.5. The Reynolds number based on the entrance height is $5 \times 10^5$. A mesh of $90 \times 60 \times 60$ is used for computation. The mesh is mostly uniformly distributed in horizontal direction, but clustered in the bump region to better resolve the shock wave. Fig. 6.17 shows the 3D mesh with every two other grid line omitted for clarity. To resolve the turbulent boundary layer, the mesh is clustered near the four walls.

Fig. 6.18 shows the computed shock wave structure (Mach number contours) compared with the experiment [133] at 3 spanwise planes. They are located at $Z = 60$ mm, 75 mm and 90 mm away from the back wall respectively. The plane at $Z$ =60mm is the central plane of the channel. The outlet static pressure is adjusted to achieve the same shock location as that in the experiment. The back pressure has the value of $p_{out}/p_t = 0.53$. The

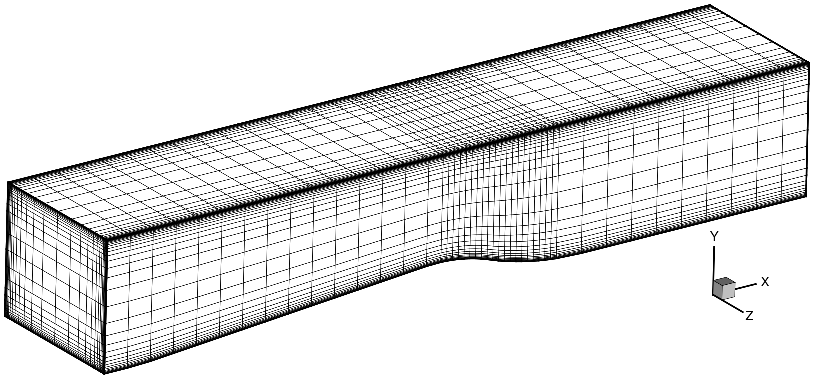
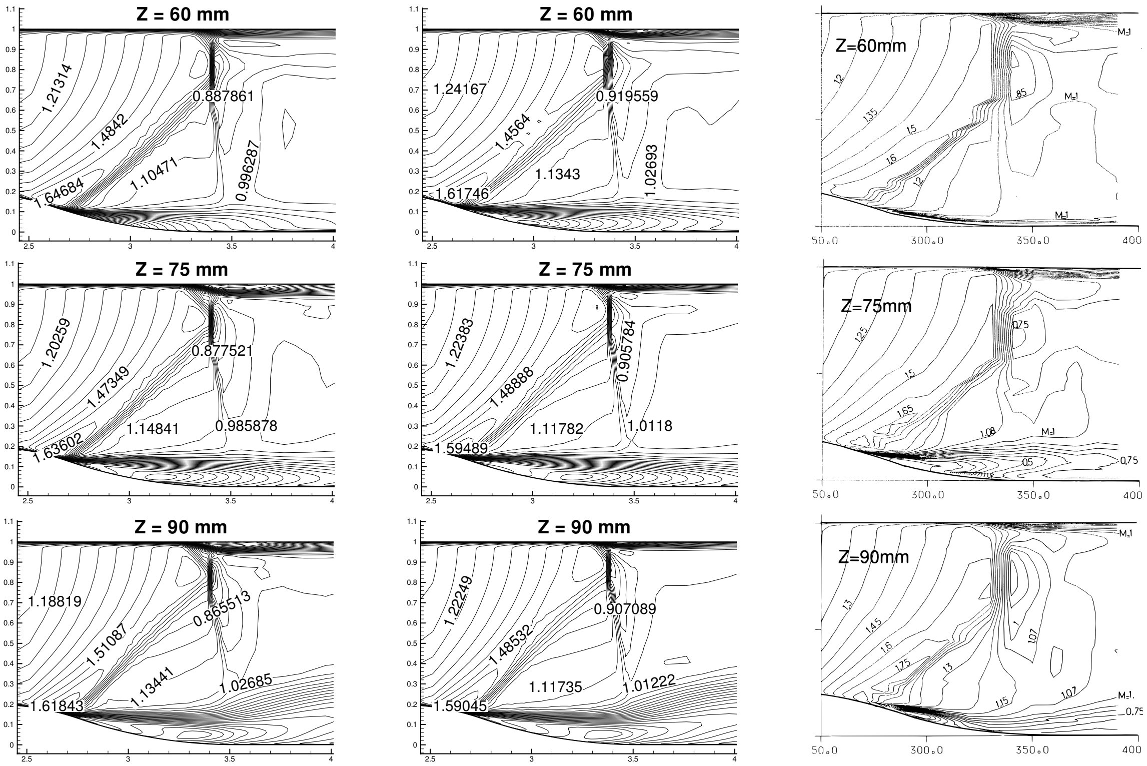


Figure 6.17: Transonic duct 3D mesh



(A) Zha CUSP 2 scheme (B) Roe scheme (C) Experiment

Figure 6.18: Transonic duct Mach number contours

Zha CUSP2 scheme and the Roe scheme result in the shock wave structure very similar to the experiment. The computed maximum Mach numbers using the Roe scheme and the Zha CUSP scheme are a little lower than that in the experiment. However, the Zha CUSP2 scheme gives the maximum Mach number closer to the experiment than the Roe scheme. Both the schemes predict the boundary layer thicker than that measured in the experiment. This may be mainly due to the inadequacy of the turbulence modeling. The Baldwin-Lomax model is based on the empirical mixing length assumption and is more reliable for attached flows in equilibrium [134]. In the transonic channel, the flow is separated after the throat, which makes the performance of the Baldwin-Lomax model not as good as the first two cases with flow attached.

At $Z$ =60 mm, the experiment shows no flow separation, so does the Zha CUSP2 scheme. However, the Roe scheme predicts a flow separation at that location, which is different from the experiment. At location Z =75mm, both the Zha CUSP2 scheme and the Roe scheme predict the flow separation similar to the experiment. However, at the location Z =90 mm which is close to the side wall, the results computed by both the schemes predict larger separation zone than that of the experiment.

The mesh refinement study with the mesh size of $180 \times 60 \times 60$, $90\times90\times60$ and $90\times60\times90$ gives very similar results, which indicates that the solution is mesh independent.

# Chapter 7

# Stationary Cascade Separation

The flow separation of the NASA flutter cascade with stationary blades is numerically studied in this chapter. The separation is caused by its high incidence angle. The van Leer scheme is used for the inviscid flux evaluation because it gives better agreement with the experiment measurement in the size of the separation region. The time-averaged separation characteristics under 3 different inlet Mach numbers is first studied with the 3D steady state simulation. Then the unsteady simulation is carried out based on the steady state results. The vortex shedding, which is the mechanism behind the unsteady flow oscillation is studied and described in details for the case of $M$=0.5. The frequency characteristics of the unsteady results is analyzed with Fourier transform technique.

## 7.1 NASA Flutter Cascade

The test section of the NASA transonic flutter cascade facility is shown in Fig. 7.1 . The test section has a rectangular cross section of 5.84 cm wide (pitch $s$) by 9.59 cm high (height $h$). The aerodynamic chord $C$ is 8.89 cm with a maximum thickness of 0.048$C$ at 0.625$C$ from the leading edge. This results in a solidity $C/s$ of 1.52. The experimental blades have constant cross sections in the span-wise direction, except near the end walls, where they have large, diamond-shaped fillets to support the attachment shafts. The blades

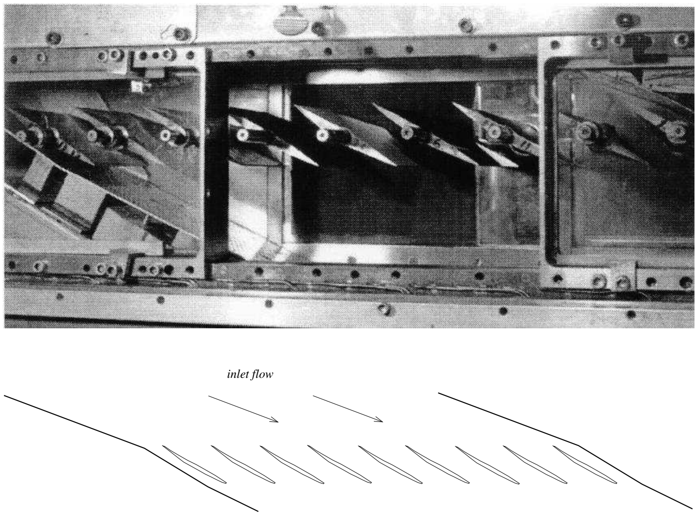


Figure 7.1: NASA transonic flutter cascade structure

are not exactly symmetric about the mid-span plane. The fillet on the drive-side is larger than the one on the free side. This makes the full 3D calculation necessary.

The 3D mesh used here is similar to the cascade 3D mesh used in last chapter as shown in Figs. 6.13 and 6.14. The end walls are located at the bottom and the top. The mesh size is 100 (streamwise)×60 (pitchwise)×60 (spanwise) with 15, 70 and 15 mesh cells allocated upstream, on and downstream the blade surface. The mesh is clustered in the area close to the blade surface on the pitch direction and in the area close to the top and bottom end walls in the spanwise direction. In the streamwise direction, the mesh is clustered in the area close to the leading and trailing edges of the blade, where the flow structure is complicated. The maximum $y^+$ at all wall boundaries is under 3.

The inlet and the outlet of the computational domain are located at $1.2C$ upstream and downstream of the blade leading edge and trailing edge. No-slip adiabatic boundary condition is applied on the blade surfaces and on the top, bottom end walls. The inlet and outlet boundary conditions vary with the local Mach number. At the inlet, if the flow is supersonic, all parameters, including the total pressure $p_t$, total temperature $T_t$, the streamwise velocity and the flow angles $\alpha$, $\beta$ are given at the inlet boundary. If the inlet flow is subsonic, the velocity is extrapolated from the inner domain. At the outlet boundary, if the flow is supersonic, all parameters are extrapolated from upstream to the boundary with zero gradient. If the flow is subsonic, a constant static pressure $p_{out}$ is given. In the pitch direction, the periodical boundary condition is applied on the boundaries upstream and downstream of the blade surfaces.

The incidence angle and the inlet Mach number are the two important factors to determine the flow separation pattern in the cascade. In the following simulations, the cases with incidence angles of 0° and 10°, and inlet Mach numbers of 0.5, 0.8 and 1.18 are computed and compared with the experimental results. As suggested in [135], the flow incidence angle ahead of the blade will not exactly follow the chordal incidence angle. The incidence uncertainty in the experiment is up to 1.5°. Therefore, the incidence angle is adjusted for each case in the computation based on the experiment results. During the process of the Gauss-Seidel iterations, a smaller CFL number of 1.0 is used in the first 50 steps and then a higher CFL number of 5.0 is used for the rest computation. For the 3D cascade simulation, the van Leer scheme instead of the Roe scheme is used to evaluate the inviscid flux. The Roe scheme predicts the separation region significantly larger than that in the experiment.

## 7.2 Steady State Results

### 7.2.1 Cascade flow without separation

Before studying the high incidence angle cases, the zero incidence angle case is computed to validate the van Leer scheme in 3D condition. In the calculation, the flow incidence is set as 2.5° for the case of incidence 0° in the experiment due to the experimental uncertainty. The inlet Mach number of 0.5 is achieved by adjusting the back pressure at the outlet. The Reynolds number based on the chord length $C$ is $9.67\times10^5$.

Figure 7.2 shows the computed mid-span stream lines at incidence angle 0°. The flow goes through the cascade passage smoothly without separation. The distribution of the computed static pressure coefficients, $C_p = \frac{p-p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}$, at the mid-span plane is shown in Fig. 7.3 , which is in good agreement with the experiment. In the $C_p$ formulation, $p$ is the

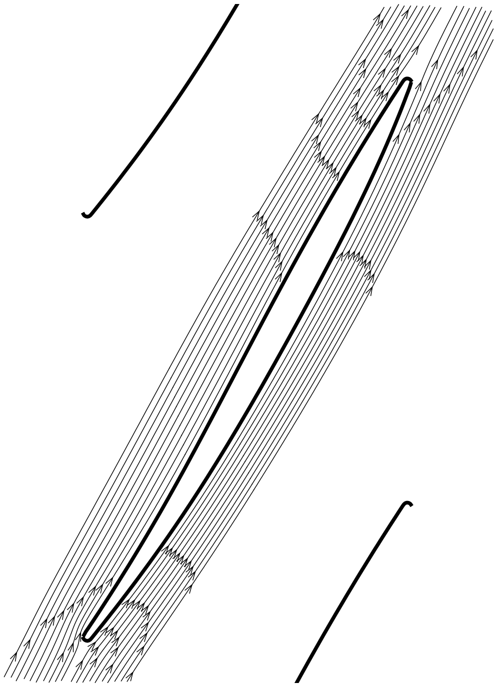


Figure 7.2: Flow pattern of the inlet-diffuser of the NASA transonic cascade at incidence 0°

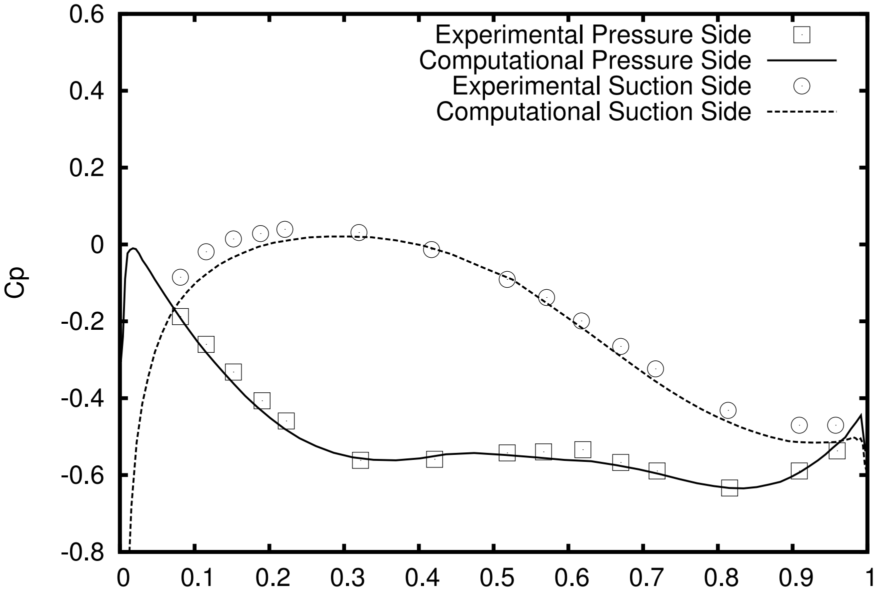


Figure 7.3: Mid-span static pressure distribution of the NASA transonic cascade at 0°, $M$=0.5

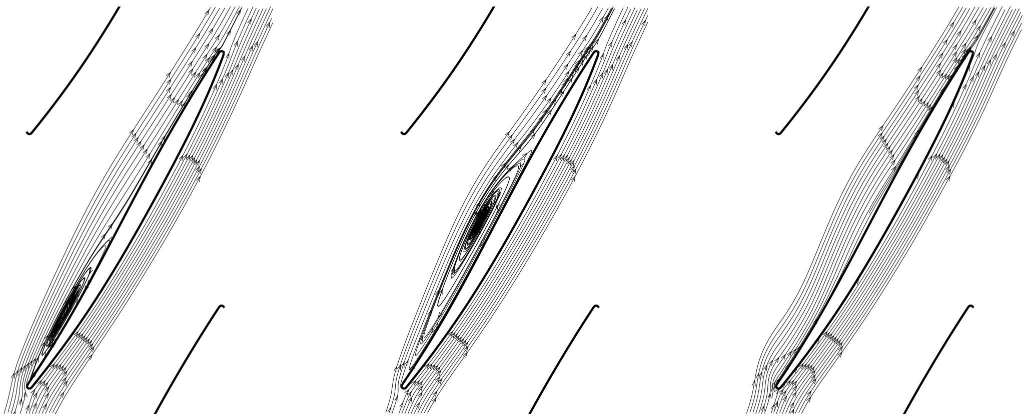

(a) M=0.5 (b) M=0.8 (c) M=1.18

Figure 7.4: Mid-span flow pattern of the NASA transonic cascade at 10° incidence under different inlet Mach numbers

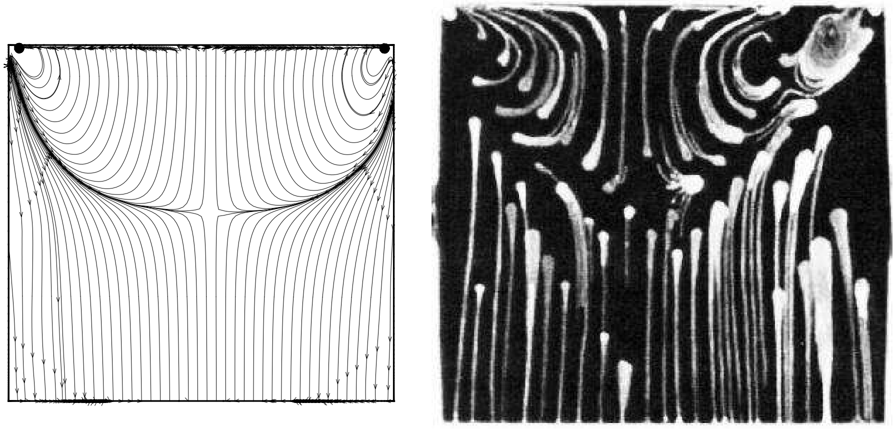
local static pressure, $p_\infty$, $\rho_\infty$ and $U_\infty$ are the averaged static pressure, density and velocity at the inlet.

### 7.2.2 Cascade flow with separation

The flow is separated when the incidence is increased. The characteristics of the separation varies with the inlet Mach number. Mach numbers of 0.5, 0.8 and 1.18 are chosen for calculation since the corresponding experimental measurement results are available for comparison. The experimental incidence angle of 10° is used for all the calculations below.

**M = 0.5**

The stream lines on the mid-span plane with inlet Mach number 0.5 is shown in Fig. 7.4 (a) . It differs from the low incidence angle case (Fig. 7.2) with a large flow separation area on the suction surface, which starts immediately at the leading edge and extends down to 45% of the blade chord. The computed and measured flow patterns from the bottom wall (left) to the top wall (right) on the suction surface are plotted in Fig. 7.5 . The experimental flow pattern is obtained with the dye oil technique. The computation shows the similar flow pattern as measured in the experiment. The computed separation bubble length in the



(a) Computed flow pattern (b) Experiment visualization

Figure 7.5: Suction surface flow pattern of the NASA transonic cascade at 10° incidence, *M*=0.5

midspan location agrees very well with the experiment, but the computed separation area is fuller in the span-wise direction. This indicates that the computed end wall boundary layer is thiner than that in the experiment. The possible reason is that the computational domain has shorter end wall length than the wind tunnel in the experiment.

The computed separation region has a parabola shape, which is approximately symmetric about the blade mid-span plane. Two counter rotating vortexes are formed downstream of the blade leading edge and the end wall corners.

The mid-span surface static pressure distribution is plotted and compared with the experiment measurement in Fig. 7.6 . The pressure on the pressure surface agrees fairly well with the experiment. In the area of the suction side leading edge, the computed pressure rises more steeply than that of the measurement. The separation bubble length represented by the cross of the surface pressure distribution on the suction and the pressure surface is predicted accurately compared with the experiment.

**M = 0.8**

At the inlet Mach number of 0.8, the separation exhibits the similar pattern to that in the case of Mach number 0.5. However, the size of the separation region increases significantly.

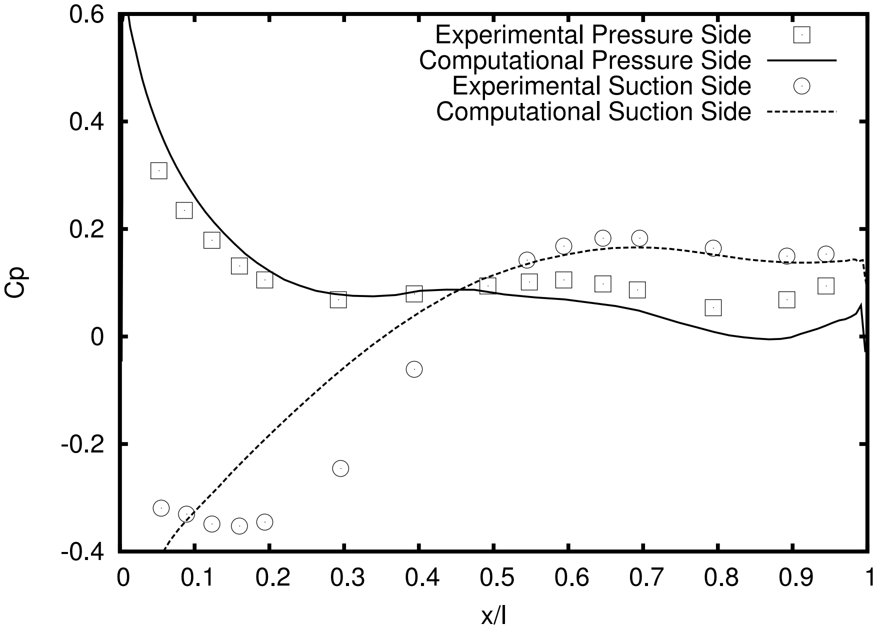
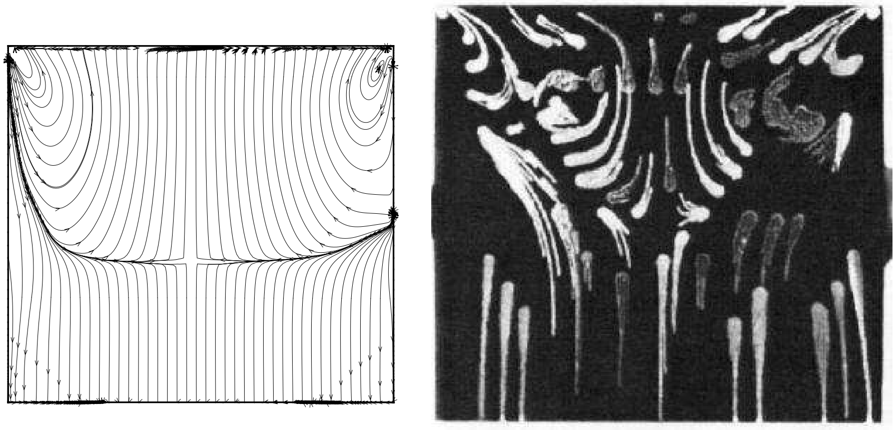


Figure 7.6: Mid-span static pressure distribution of the NASA transonic cascade at 10° incidence, *M*=0.5

As shown in the computed stream lines in Fig. 7.4 (b), the separation bubble is larger in both streamwise and pitchwise directions on the midspan plane. The separation region on the suction surface becomes as long as 60% of the chord length. The growth of the separation is also clearly shown in the suction surface flow pattern in Fig. 7.7 . The separation length is a little longer than the corresponding experiment visualization result on the mid-span plane and fuller in the spanwise direction. The asymmetry of the flow pattern due to the asymmetric geometry is more clearly seen in Fig. 7.7.

The mid-span surface static pressure is plotted and compared with the measurement in Fig. 7.8 . Similar to the case of *M*=0.5, the computed surface pressure agrees very well on the pressure side. The main difference remains in the leading edge region of the suction surface with the computed surface pressure rising too rapidly. Again, the separation bubble length is predicted fairly well compared with the experiment.



(a) Computed flow pattern

(b) Experiment visualization

Figure 7.7: Suction surface flow pattern of the NASA transonic cascade at 10° incidence, $M$=0.8

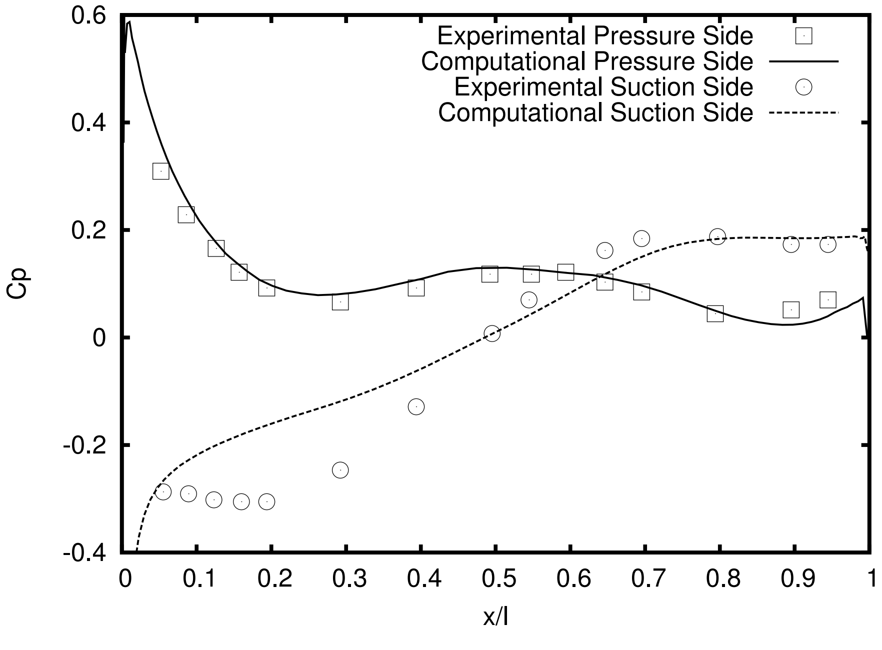
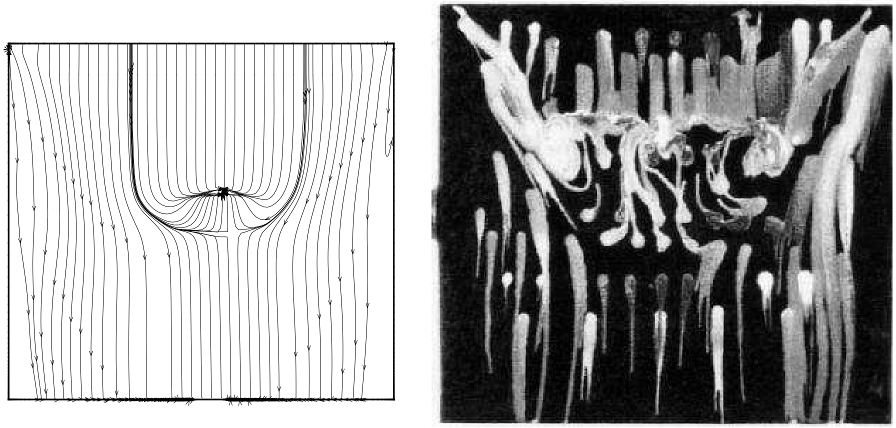


Figure 7.8: Mid-span static pressure distribution of the NASA transonic cascade at 10° incidence, $M$=0.8



(a) Computed flow pattern

(b) Experiment visualization

Figure 7.9: Suction surface flow pattern of the NASA transonic cascade at 10° incidence, $M$=1.18

**M = 1.18**

For the supersonic incoming flow with Mach number 1.18 and incidence of 10°, the flow pattern is very different from those in the subsonic cases. In Fig. 7.4 (c), the stream lines go through the cascade nearly smoothly. However, a small and thin separation region exists on the center part on the suction surface, where the stream lines deviate from the suction surface slightly. The separation region is clearly shown in the flow pattern on the suction surface in Fig. 7.9 . Compared with the results of the subsonic cases above, the separation is pushed more downstream with its size shrunk. The flow is attached before the separation and re-attached after the separation. The reduced size of the separation may be due to the increased kinetic energy with higher inlet Mach number compared with the subsonic flow. The separation is mostly induced by the shock wave-turbulent boundary layer interaction. The same flow pattern is observed in the experimental visualization (Fig. 7.9) with the separation bubble size a little larger than that obtained in the computation.

There are two shock waves captured in the computation, the passage shock and lip shock as shown in Fig. 7.10 . Compared with the experiment, the passage shock is located more upstream than the measured one. It is suspected that the computed boundary layer is predicted to be too thick and the excessive blockage pushes the shock more upstream.

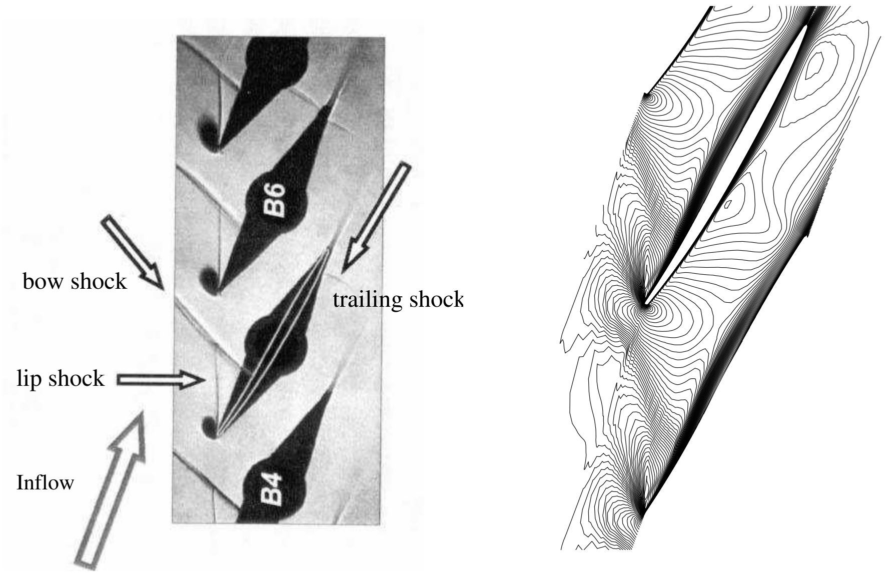


Figure 7.10: Experimental shock structure of the NASA transonic cascade at 10° incidence, *M*=1.18

The local adverse pressure gradient caused by the shock wave-turbulent boundary layer interaction is the reason to create the local separation shown in Fig. 7.9. In the experiment, there is a very weak trailing shock, which is not captured by the computation.

## 7.3 Unsteady Simulation

The separation is believed to bring high unsteadiness to the cascade flow pattern. The inlet Mach number is an important factor which affects the separation characteristics [136]. To study the influence of the inlet Mach number on the unsteady characteristics of the separated flow, numerical simulation is carried out for high incidence angle cases with Mach number 0.5, 0.8 and 1.18. Each unsteady calculation is carried out based on its corresponding steady state result.

Due to the limitation of the computation capability, the physical time interval is chosen as large as 10% of the characteristic time of the cascade, $t_c = \frac{C}{U_\infty}$. This time interval varies with the inlet Mach number. The CFL number used in the pseudo time Gauss-Seidel

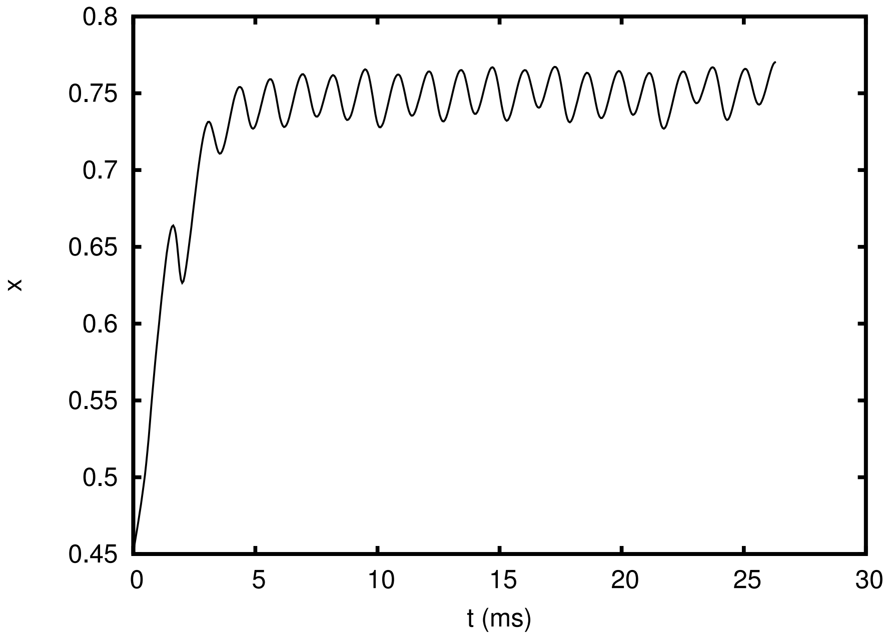


Figure 7.11: Separation zone length variation with time in 3D stationary cascade at 10° incidence, $M$=0.5

implicit iteration is 20. Twenty pseudo time steps are used for each physical time step. Two parameters are recorded to analyze the unsteady characteristics of the separation flow. The first is the mid-span separation bubble length($x$), which is marked by the streamwise zero velocity point along the suction surface. The second parameter is the unsteady static pressure ($p$) measured at the location of 13% downstream the leading edge on the suction surface, which is the same as the experiment measurement location. The unsteady pressure is referred as "checkpoint pressure" below.

**M = 0.5**

In the case of Ma = 0.5, the physical time interval is set as 0.052744 ms.

Fig. 7.11 shows the time history of the separation length oscillation in a time segment of 16 ms ($30t_c$). The separation length increases rapidly from 0.45$C$ to 0.66$C$ in the first 1.63 ms ($3t_c$) and then decreases to 0.63$C$ at $t$ =2 ms ($3.8t_c$). The separation region then grows again toward downstream to 0.73$C$ at $t$ =3 ms ($5.7t_c$). With the time progressing,

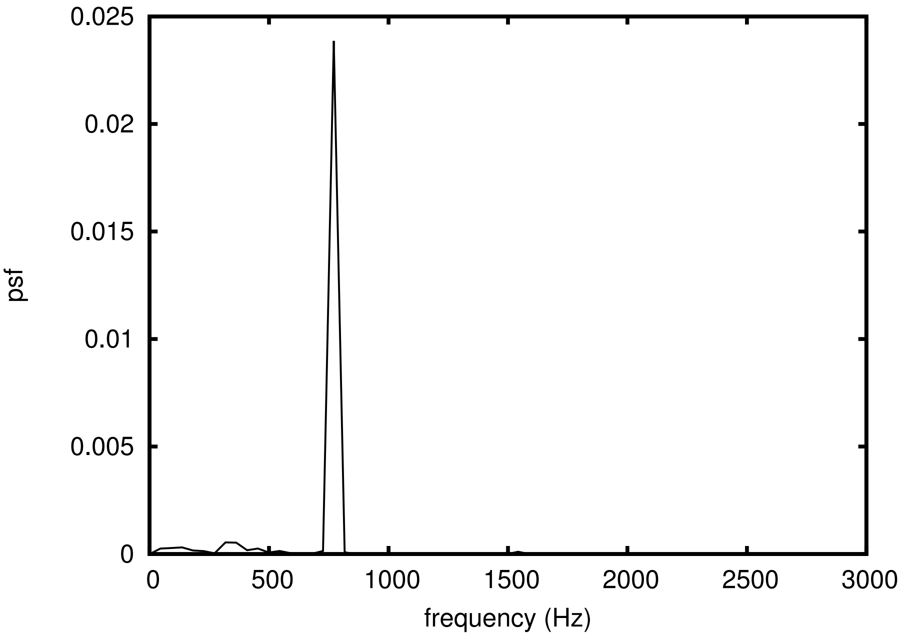


Figure 7.12: Separation zone length variation frequency spectrum in 3D stationary cascade at 10° incidence, $M$=0.5

the separation region boundary oscillates back and forth on the suction surface. The average length tends to increase gradually until a periodic state is reached after 4.4 ms ($8.3t_c$). The oscillation of the separation length is between $0.73C$ and $0.76C$ with a fixed cycle. The periodicity information is clearly extracted using the Fourier transform technique. The separation region oscillation spectra is calculated from the unsteady data after 4.4 ms ($8.3t_c$). The frequency spectrum is shown in Fig. 7.12 , which clearly shows a peak at 770 Hz. This indicates the separation length oscillates with a period of 1.25 ms ($2.37t_c$).

Compared with the steady state solution, the unsteady separation calculation results in larger separation size. The outlet boundary condition implementation is the possible reason for the difference. The fixed static pressure brings reflective wave back to the upstream and contaminates the computation result, which is more sensitive in the unsteady computation. The influence can be minimized by using the non-reflective boundary condition at outlet.

The unsteady checkpoint pressure data shows similar characteristics of the unsteady separation flow. Fig. 7.13 shows a segment of 16 ms ($30t_c$) pressure oscillation data. The

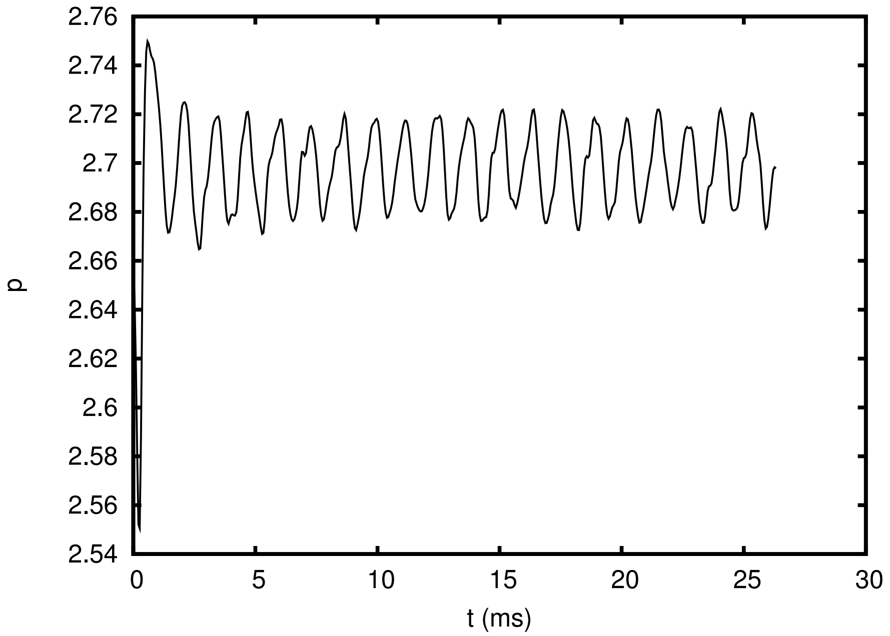


Figure 7.13: Suction surface checkpoint pressure variation with time in 3D stationary cascade at 10° incidence, $M$=0.5

pressure start at $p$ =2.659 from the steady state result. The oscillation is between $p$ =2.68 and $p$ =2.72 after $t$ =4.4 ms (8.3$t_c$). The oscillation amplitude is about 1.5% of the averaged pressure level. The frequency spectrum is shown in Fig. 7.14 with a peak at 770 Hz.

The mechanism behind the unsteady characteristics of the separation is illustrated in Fig. 7.15 , where the evolution of a separation oscillation cycle is plotted. The stream lines at 8 time levels show the leading edge vortex shedding development. There are 4 physical time steps (0.21 ms, 0.4$t_c$) between 2 sequential plots. The relationship between the oscillation of the pressure and the separation length is shown in Fig. 7.16 .

At the starting time level $a$ ($t$ =4.4305 ms, 8.4$t_c$), the separation region has just passed the maximum length location. There are 2 vortexes in the separation bubble. They are rotating in the same direction. The checkpoint pressure is going up. At time level $b$ ($t$ =4.6415 ms, 8.8$t_c$), the two vortexes are pushed toward downstream. The second vortex diminishes. The first vortex grows quickly and becomes the only vortex in the separation bubble. The separation region becomes thicker in the span-wise direction, but shorter in

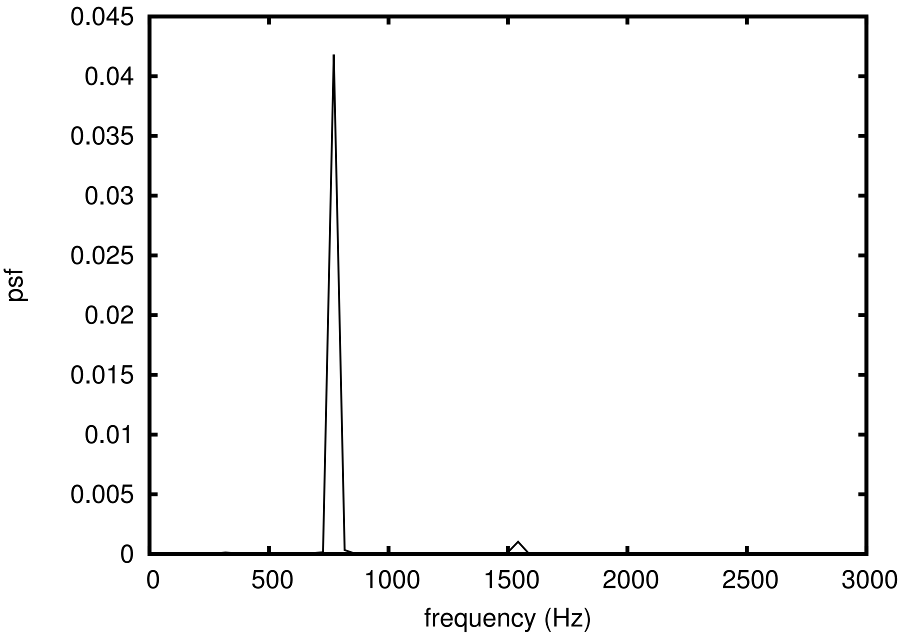
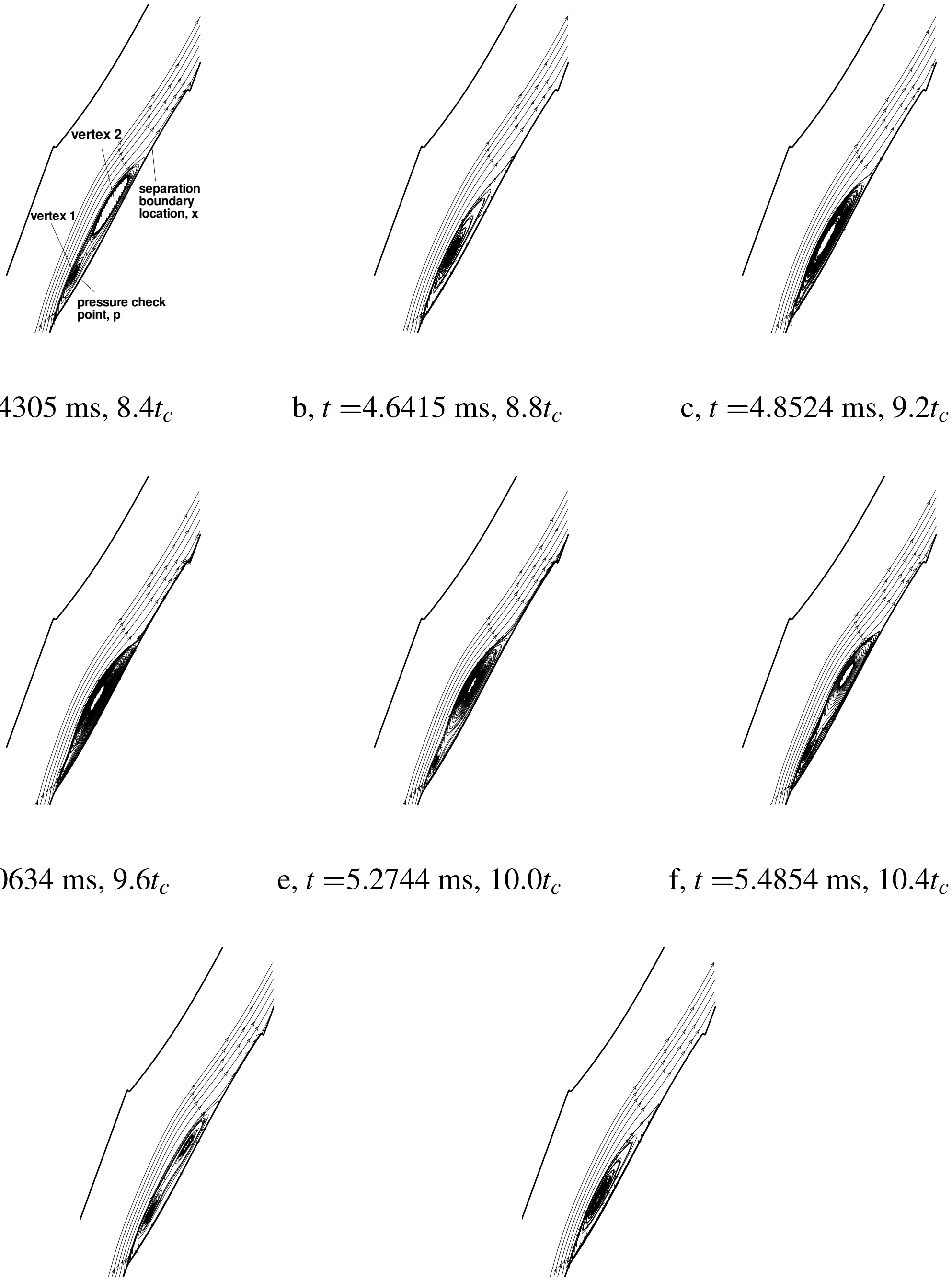


Figure 7.14: Suction surface checkpoint pressure frequency spectrum in 3D stationary cascade at 10° incidence, $M$=0.5

the stream-wise direction. The surface checkpoint pressure reaches its maximum level at this time level. At time level $c$ ($t$ =4.8524 ms, $9.2t_c$), the separation bubble approaches its shortest length in stream-wise direction, and maximum thickness in the span-wise direction. The checkpoint pressure is going down. At time level $d$ ($t$ = 5.0634ms, $9.6t_c$), the separation bubble has passed the minimum length location, and begins to extend toward downstream. The checkpoint pressure is still going down. At time level $e$ ($t$ =5.2744 ms, $10.0t_c$), a new vortex is generated at leading edge and becomes the first vortex. The checkpoint pressure reaches its minimum value. The separation length is still increasing. At time level $f$ ($t$=4854 ms, $10.4t_c$), the first vortex continues to grow. The second vortex is pushed toward downstream. Both the checkpoint pressure and the separation length are going up. The latter is approaching its maximum location. At time level $g$ ($t$ =5.6964 ms, $10.8t_c$), the two vortexes have almost the same size, the flow structure is close to the starting time level $a$. The separation boundary has passed its maximum location and begin to shrink toward upstream. The checkpoint pressure is going up. At time level $h$ ($t$ =5.9073 ms, $11.2t_c$), the



a, $t$ =4.4305 ms, 8.4$t_c$

b, $t$ =4.6415 ms, 8.8$t_c$

c, $t$ =4.8524 ms, 9.2$t_c$

d, $t$ =5.0634 ms, 9.6$t_c$

e, $t$ =5.2744 ms, 10.0$t_c$

f, $t$ =5.4854 ms, 10.4$t_c$

g, $t$ =5.6964 ms, 10.8$t_c$

h, $t$ =5.9073 ms, 11.2$t_c$

Figure 7.15: Separation bubble evolution in 3D stationary cascade at 10° incidence and $M$=0.5

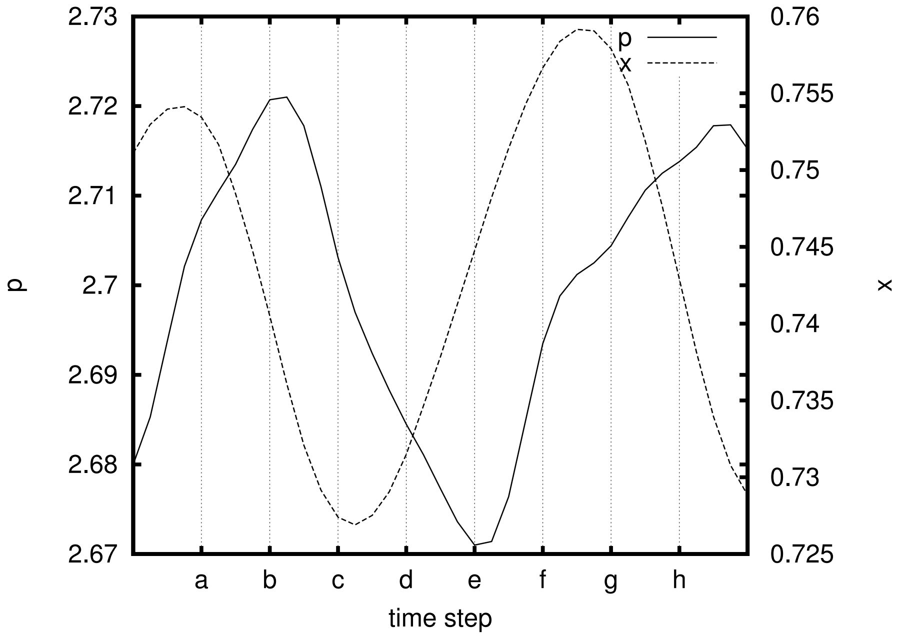


Figure 7.16: Checkpoint pressure and separation locations relation in 3D stationary cascade at 10° incidence, $M$=0.5

second vortex diminishes. A new cycle is started at this time level.

The leading edge vortex shedding exhibits obvious periodical pattern in its evolution process. The leading edge keeps generating new vortexes. The new vortex pushes the old vortex bubble toward downstream and the old vortex decreases in size at the same time. When the two vortexes become of the same size, the maximum separation length is reached, where the separation bubble has the thinnest size in span-wise direction. As the new vortex grows further, the old vortex will diminish. The separation bubble will move upstream and makes the bubble thicker. The leading edge surface checkpoint pressure reaches its maximum level when the separation bubble shrinks and reaches its minimum level when the separation bubble boundary extends. The vortex generation, pressure variation and separation length oscillation have the same frequency characteristics with a phase difference as shown in Fig. 7.16. Such oscillation may be one of the reasons to cause to blade vibration.

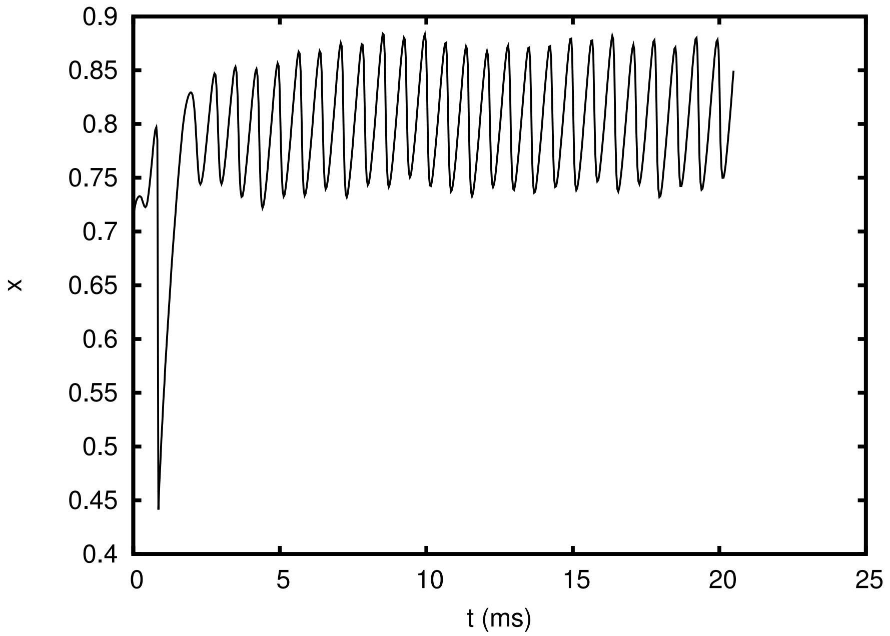


Figure 7.17: Separation zone length oscillation in 3D stationary cascade at 10° incidence, *M*=0.8

**M = 0.8**

The physical time interval used in the calculation for the case of Mach number 0.8 is 0.0342 ms. A similar flowfield unsteady characteristics is exhibited in the computation results.

Fig. 7.17 and Fig. 7.18 show the separation length and the checkpoint static pressure oscillation history in a time period of 21 ms. A clear periodicity is shown in these two figures. It is found in the time averaged steady state study in [136] that, the increase of the inlet Mach number will enlarge the separation bubble in size. Fig. 7.18 indicates that the inlet Mach number increase also increases the amplitude of the pressure oscillation. The oscillation amplitude is increased to about 5% of the averaged pressure level. The increased kinetic energy in the inflow bring higher unsteadiness intensity to the separated flow filed.

The corresponding frequency analysis is shown in Fig. 7.19 and Fig. 7.20 respectively. The frequency analysis is based on the oscillation data after *t*=5 ms. The unsteady separation flow exhibits higher oscillation frequency because of the increased inlet Mach number.

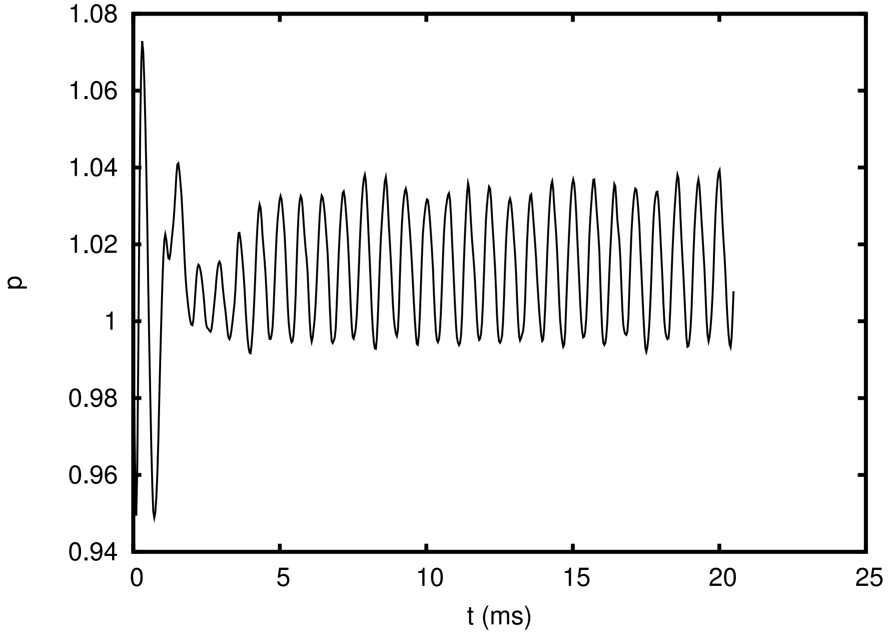


Figure 7.18: Checkpoint pressure oscillation in 3D cascade at 10° incidence, $M$=0.8

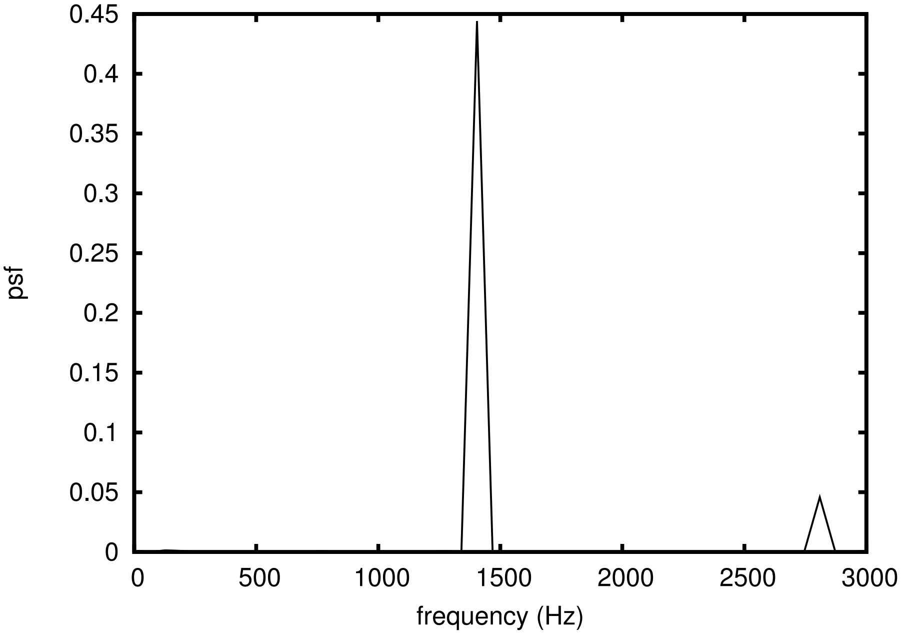


Figure 7.19: Separation zone length oscillation frequency spectrum in 3D cascade at 10° incidence, $M$=0.8

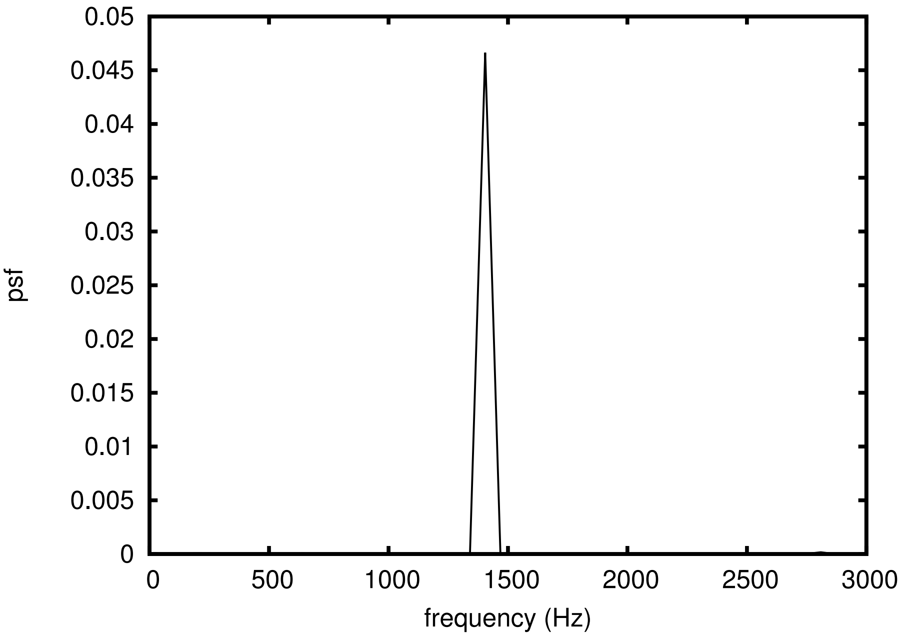


Figure 7.20: Checkpoint pressure oscillation frequency spectrum in 3D stationary cascade at 10° incidence, $M$=0.8

A clear frequency spectrum peak is shown at 1400 Hz in both figures, twice the frequency in the case of $M$ = 0.5.

### M = 1.18

In the steady state simulation of the cascade at $M$ = 1.18 in [136], the separation flow characteristics are very different from those at subsonic. The increased kinetic energy in the inflow makes the flow attached to the blade surface in the leading edge region. A smaller size separation region appears after the shock wave because of the interaction of the shock wave and the turbulent boundary layer. The separation bubble shrinks in size and is located only in a small region at the center of the suction surface region.

The physical time interval in the calculation is set as 0.02483 ms. The pressure checkpoint is located outside of the separation region in the supersonic case. The pressure oscillation history is shown in Fig. 7.21 . The oscillation amplitude is very small compared to the cases of $M$ = 0.5 and $M$ = 0.8. The flow tends to be stable at the leading edge. The

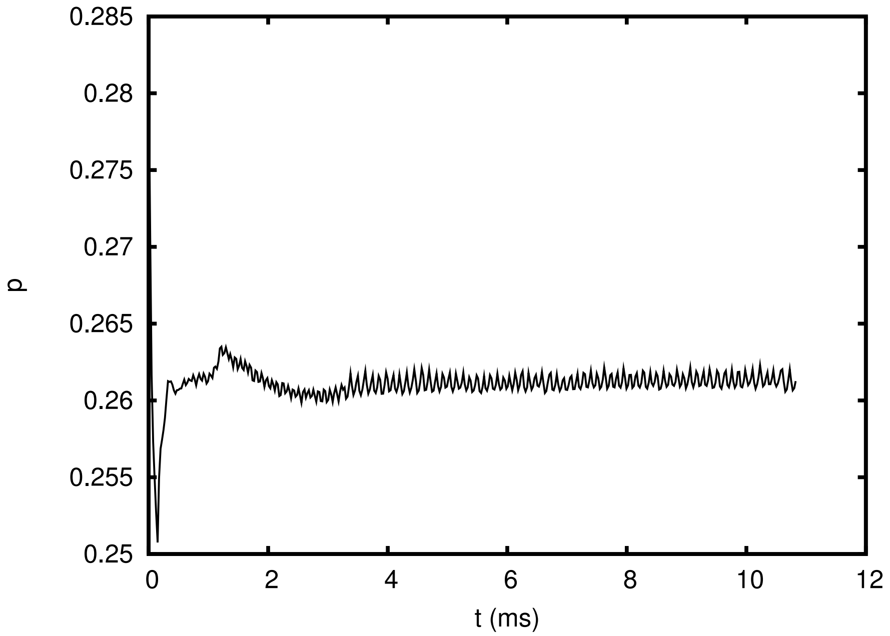


Figure 7.21: Checkpoint pressure oscillation in 3D stationary cascade at 10° incidence, $M$=1.18

pressure oscillation frequency spectrum is shown in Fig. 7.22 .

The computed characteristics of the separation flow above is similar to the experiment measurement in [137]. In [137], when the blade is fixed, the blade surface pressure for low subsonic inlet flow at $M = 0.5$ and low supersonic inlet flow at $M = 1.1$ exhibits low unsteadiness. Strong self-induced oscillations with a frequency of 110Hz is observed in high subsonic inlet flow at $M = 0.8$. However, the latter study indicated that the strong low frequency oscillation is attributed to the tunnel resonance characteristics instead of the flow unsteadiness due to the flow separation in [138]. The cascade flow separation is believed to have a direct relation with the wall surface pressure unsteady oscillation, which is an important factor to the blade vibration. Further detailed numerical research is necessary to discover the mechanism.

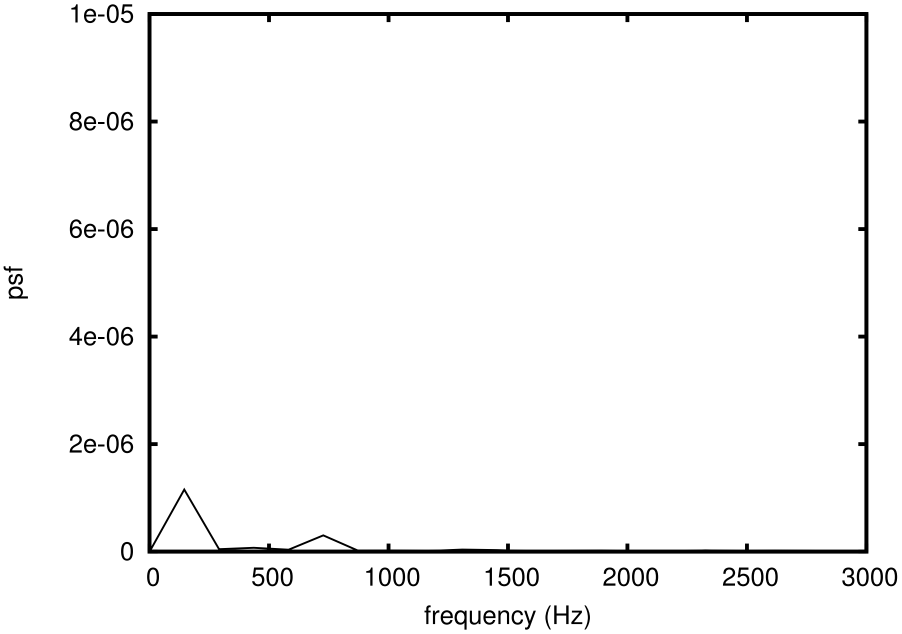


Figure 7.22: Checkpoint pressure oscillation frequency spectrum in 3D stationary cascade at 10° incidence, $M$=1.18

# Chapter 8

# Oscillating Cascade Unsteady Simulation

As part of the work to study the flutter phenomenon in cascade, the fluid-structural interaction in the unsteady cascade flow is simulated in this chapter. The experiment measurement is carried out in the same flutter cascade as Chapter 7, but with forced vibrating blades. To save computation time, the numerical studies conducted in this chapter is in 2D condition for the mid-span section. The flows in multiple passages are simulated simultaneously using parallel computation. The high resolution scheme, Zha CUSP2 scheme, developed in foregoing chapters is used to calculate the control volume interface inviscid flux. The cascade is first simplified as a 2-passage case to study the mesh dependency of the numerical results. Then the cascade is studied extensively in full scale under 3 oscillation frequencies and 2 incidences. The end wall influence on the flow pattern spatial periodicity is studied by comparing the 2-passage results with the full scale results. The unsteady results are analyzed to show the influence of the oscillation frequency and the incidence on the blade aeroelastic characteristics.

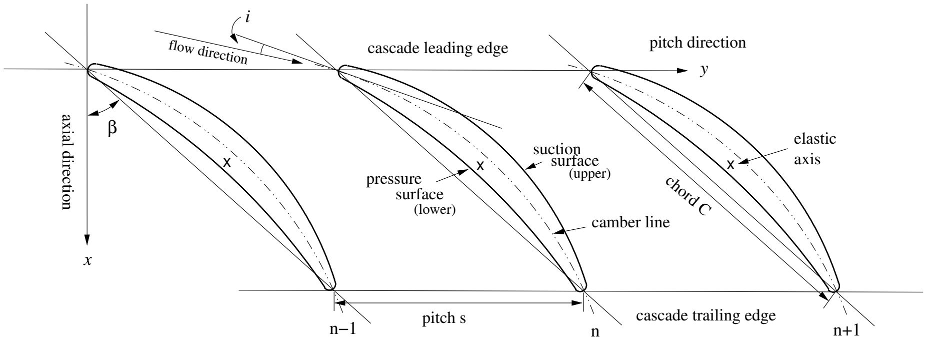


Figure 8.1: Tow-dimensional cascade geometry

## 8.1 Unsteady Performance Parameters

Unlike the stationary cascade simulation, in the forced oscillating cascade analysis, the oscillation of the flowfield and the blade stability are of the interest. The stability analysis is based on the aeroelastic unsteady parameters, for example, the unsteady pressure coefficient, the unsteady aerodynamic moment and the damping coefficient. More parameters are introduced as the following.

### 8.1.1 Inter-blade phase angle and reduced frequency

A 2D linear cascade is shown in Fig. 8.1 . Axis $x$ is along the axial direction and axis $y$ is along the pitch direction. All blades are identical in geometry. The chord length $C$ is defined as the distance between the leading edge (LE) and the trailing edge (TE). The cascade pitch is $s$. The flow incidence angle $i$ is defined as the angle between the flow direction and the camber line tangent direction at leading edge.

The blades vibrate about their elastic axes harmonically in pitching mode with a constant inter-blade phase angle (IBPA). The motion of the $n$th blade is determined by,

$$\alpha^n(t) = \alpha_0 + \alpha_A Re\left[\exp\left(i\left(\omega t + n\beta\right)\right)\right] \tag{8.1}$$

where $\alpha$ is the angular deflection about the elastic axis, $\alpha_0$ is the deflection at mean position, $n$ is the blade index, $t$ is the time, $\alpha_A$ is the amplitude of deflection angle variation,

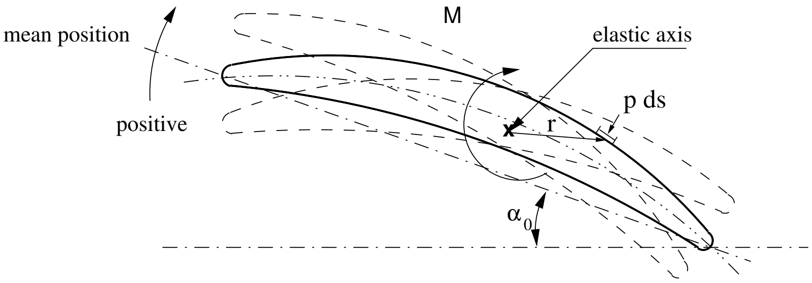


Figure 8.2: Unsteady two-dimensional cascade nomenclature

*Re* denotes the real part of a complex value, $\omega$ is the angular frequency, $\beta$ is the inter blade phase angle. The blade $n$ precedes the blade $n-1$ by phase angle $\beta$.

Fig. 8.2 shows the rotation of a single blade about its elastic axis. Attention should be paid to the definition of the deflection angle and moment used in aeroelasticity. The angular deflection $\alpha$ is defined as positive when the blade is nose up. It is opposite to the regular angle definition, which is positive in anti-clockwise direction.

The aerodynamic moment is defined as,

$$M = -\oint \mathbf{r} \times p(x)\,\mathrm{d}\mathbf{s} \tag{8.2}$$

where d**s** is the differential surface area vector with positive direction pointing into the blade, $p(x)$ is the local static pressure, **r** is the vector pointing from the elastic axis pivot to the local surface pressure acting point. The moment is also positive in clockwise direction, instead of the anti-clockwise direction in regular definition.

To be consistent with the experiment report, the reduced frequency $k_c$ is defined based on the chord length $C$, instead of the half chord used in Eq. (5.6) on page 87,

$$k_c = \frac{\omega C}{U_\infty} \tag{8.3}$$

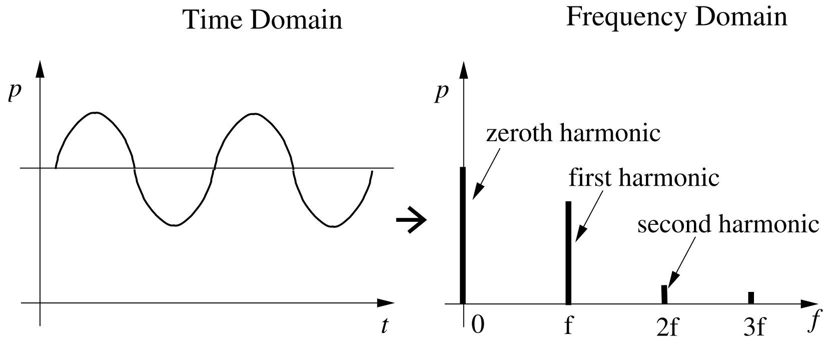


Figure 8.3: Pressure data Fourier transform

### 8.1.2 Unsteady pressure coefficient and moment coefficient

In the current unsteady fluid-structural interaction simulation, the pressure oscillation history is recorded during the unsteady time marching for post-processing. As the time marching goes on, the pressure oscillation becomes periodic with the same frequency as the blade motion. The pressure data in a certain oscillation cycle after the periodicity is achieved is then picked up for the Fourier transform analysis. If the quality of the periodicity is not good enough, pressure data over several sequential motion cycles are averaged. The averaged pressure data within one cycle is used for the Fourier transform. As shown in Fig. 8.3 , the pressure signal oscillates harmonically about its time average with a certain amplitude. After the Fourier transformation from time domain to frequency domain, the pressure oscillation frequency information is extracted and shown clearly in the $p-f$ plot.

In the forced vibration cases studied in this chapter, the most important parameter for stability analysis is the first harmonic pressure, which incorporates the pressure oscillation amplitude and phase angle. The zeroth harmonic pressure is the time averaged pressure, which is not used in the unsteady analysis. The first harmonic pressure is a complex variable. However, the start point of the unsteady pressure data may be randomly selected. In the resulted first harmonic pressure, the amplitude $p_{1A}$ is independent of the start point. However, the phase angle $\phi_{1_p}$ changes with the randomly selected starting point and is not unique. Similarly, the blade motion characterized by its deflection angle is analyzed using the Fourier transform. Because the blades under investigation are forced in a pure sinusoidal torsional motion, the frequency domain plot shows only two harmonics: the zeroth harmonic which is the mean deflection angle and the complex first harmonic which contains the information for the deflection oscillation. The amplitude is the prescribed deflection amplitude $\alpha_{1\mathrm{A}}$ and the phase angle $\phi_{1\alpha}$ is also determined by the selection of the unsteady deflection data. Usually, the pressure response, or the pressure oscillation, is not in phase with the blade motion. It may precede or lag behind the blade motion and the phase lead or lag varies along the blade surface.

In the present work, the first harmonic pressure complex is made independent of the unsteady pressure data starting point selection by normalizing the pressure phase angle $\phi_{1p}$ with the blade motion phase angle $\phi_{1\alpha}$. This is achieved by a vector division between the pressure complex value and the blade motion complex value,

$$p_{1N} = \frac{p_1}{\alpha_1/\alpha_{1\mathrm{A}}} = \frac{p_{1A}e^{i\phi_{1p}}}{e^{i\phi_{1\alpha}}} = p_{1A}e^{i\left(\phi_{1p}-\phi_{1\alpha}\right)} \tag{8.4}$$

where $p_1$ and $p_{1N}$ are the original and normalized pressure complex, subscript $A$ stands for the complex amplitude, 1 indicates the first harmonic. This phase normalization can also be considered as shifting the pressure complex and the blade motion complex by a phase angle of -$\phi_{1\alpha}$ simultaneously. The resulted blade motion has a phase angle 0 and the resulted pressure complex has a phase angle of $\phi_{1p} - \phi_{1\alpha}$, which is the phase angle relative to the blade motion. In the rest of the chapter, the subscript $N$ is omitted for clarity. If not specified, all first harmonic unsteady parameters are phase normalized by the blade motion phase angle by default.

An unsteady pressure coefficient is defined based on the first harmonic pressure $p_1$,

$$C_p(x,t) = \frac{p_1(x,t)}{\rho_\infty U_\infty^2 \alpha_\mathrm{A}} \tag{8.5}$$

where $x$ is the dimensionless chordwise distance normalized by the chord length $C$, $\rho_\infty$, $U_\infty$ are respectively the free stream density and velocity. Unlike the steady state pressure coefficient, the unsteady pressure coefficient is a complex value with pressure oscillation phase and amplitude information contained.

The aerodynamic moment computed by Eq. (8.2) on page 132 also varies with the blade motion. The unsteady complex aerodynamic moment coefficient is defined as,

$$\begin{aligned} C_M(t) &= \frac{M(t)}{\rho_\infty U_\infty^2 \alpha_\mathrm{A} C^2} = -\frac{\oint \mathbf{r} \times p(x,t)\,\mathrm{d}\mathbf{s}}{\rho_\infty U_\infty^2 \alpha_\mathrm{A} C^2} \\ &= -\frac{\oint \frac{\mathbf{r}}{C} \times p(x,t)\,\mathrm{d}\frac{\mathbf{s}}{C}}{\rho_\infty U_\infty^2 \alpha_\mathrm{A}} \end{aligned} \tag{8.6}$$

In reference [131], for a flat plate airfoil oscillating along its half chord axis, Eq. (8.6) is further simplified to the following,

$$C_M(t) = \int_0^1 \left(\frac{x_p}{C} - \frac{x}{C}\right) \Delta C_p \left(\frac{x}{C}\right) \mathrm{d}\frac{x}{C} \tag{8.7}$$

where $x_p/C = 0.5$. The pressure difference coefficient $\Delta C_p$ is defined as the difference between the pressure surface and suction surface unsteady coefficients,

$$\begin{aligned} \Delta C_p(x) &= C_{p,pre}(x) - C_{p,suc}(x) \\ &= C_{p,lower}(x) - C_{p,upper}(x) \end{aligned} \tag{8.8}$$

where the subscripts *pre* and *suc* stand for the pressure and suction surfaces, which are the corresponding *lower* and *upper* surfaces in the two-dimensional cascade nomenclature (Fig. 8.1).

### 8.1.3 Work per cycle and aerodynamic damping

When the blade oscillates in fluid flow, energy is exchanged between the blade and its surrounding airstream. The blade aerodynamic stability is determined by the work per motion cycle of the airstream acting on the blade. The blade is aerodynamically stable when the work is negative and aerodynamically unstable when the work is positive. The positive work means flow does work on the blade and the negative work means the blade does work on the flow.

The work per motion cycle is given by the cyclic integral of the product of the real parts of the moment and the differential deflection angle,

$$W = \oint M_{\mathrm{R}} \mathrm{d}\alpha_{\mathrm{R}} \tag{8.9}$$

where, the subscript R indicates the real part of the first harmonic complex of moment and pressure. Subscript 1 is omitted in this section for clarity.

The first harmonic angle deflection and the first harmonic moment are written as,

$$\alpha = \alpha_{\mathrm{A}} e^{i\omega t} \tag{8.10}$$

$$M = M_{\mathrm{A}} e^{i\omega t} = \left(M_{\mathrm{R}} + iM_{\mathrm{I}}\right) e^{i\omega t} \tag{8.11}$$

where $M_{\mathrm{A}}$ and $M_{\mathrm{I}}$ are the unsteady moment amplitude and the imaginary part of the first harmonic moment. The blade motion oscillation amplitude $\alpha_{\mathrm{A}}$ is a real value. However the moment amplitude $M_{\mathrm{A}}$ is generally a complex value, because the unsteady moment is not generally in phase with the blade motion.

Substitute Eqs. (8.10) and (8.11) into Eq. (8.9), and then use orthogonality to finish the integration, the work per cycle is obtained as,

$$W = \pi \alpha_{\mathrm{A}} M_{\mathrm{I}} \tag{8.12}$$

Because the work per cycle is proportional to $M_{\mathrm{I}}$, the out of phase part of the moment, the blade is unstable when $M_{\mathrm{I}} > 0$ and stable when $M_{\mathrm{I}} < 0$. The work coefficient is obtained by having Eqs. (8.9) and (8.12) divided by $\rho_\infty U_\infty^2 C^2$,

$$C_W = \frac{W}{\rho_\infty U_\infty^2 C^2} = \oint \alpha_\mathrm{A} C_{M_\mathrm{R}} \mathrm{d}\alpha_\mathrm{R} = \pi \alpha_\mathrm{A}^2 C_{M_\mathrm{I}} \tag{8.13}$$

In a linear, damped system surrounded by airstream, if the damping is small and the system is executing a single-degree-of-freedom torsional motion, the aerodynamic damping parameter $\xi$ of the system is related to the out of phase part of the unsteady moment as [139],

$$\xi = -\frac{M_\mathrm{I}}{\alpha_\mathrm{A}} \tag{8.14}$$

The dimensionless damping coefficient $\Xi$ is obtained by dividing Eq. (8.14) through $\rho_\infty U_\infty^2 C^2$,

$$\Xi = \frac{\xi}{\rho_\infty U_\infty^2 C^2} = -C_{M_\mathrm{I}} \tag{8.15}$$

Hence the system is stable when $\Xi > 0$ and is unstable when $\Xi < 0$.

The damping coefficient is related to the work coefficient as,

$$\Xi = -\frac{C_W}{\pi \alpha_\mathrm{A}^2} \tag{8.16}$$

Buffum et al. [131] use a work per cycle chordwise distribution to analyze the blade local stability,

$$C_w' = \left(0.5 - \frac{x}{C}\right) Im\left(\Delta C_p\right) \tag{8.17}$$

where *Im* stands for the imaginary part of a complex variable.

The blade is considered as locally unstable when the fluid does work on the blade ($C_w' > 0$). A negative $C_w'$ increases the blade local stability.

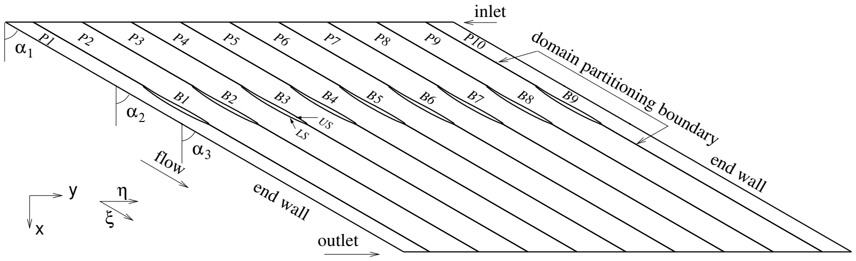


Figure 8.4: Full scale cascade computation domain configuration

## 8.2 Parallel Computation Implementation

The NASA Lewis Oscillating Cascade is basically designed to simulate the two dimensional flow conditions [131]. As shown in Fig. 8.4 , the cascade has 9 identical blades, marked as B1 through B9, surrounded by 2 end walls. The computation domain is decomposed to 10 subdomains according to the flow passages, which are separated by the blades and the streamlines going through the leading edge and trailing edge of each blade. Because the stream lines are unknown before calculation, the dividing interfaces upstream and downstream the blades are the straight lines aligning with the wall surfaces. The domain partitioning boundary condition is applied on these interfaces . The flow is going through the cascade from top-left to bottom-right in Fig. 8.4.

The computer system used in current work is a Beowulf cluster composed of 14 dual processor DELL workstations. The CPU speed varies from 1.7 GHz to 3.2 GHz. The built in memories vary from 1 GB to 3 GB. All the 14 machines are equipped with 1 GBps interface cards and are connected through cables. The operating system is the Red Hat Linux. The hard disk space is shared by all machines using the NFS file system. A single copy of the executable code is shared by all processors in the MPI group during computation.

In the current simulation, the 10 passages are computed simultaneously by 10 processors on 5 machines . The first processor P1 works as the coordinator processor to synchronize the activity of all the 10 processors during iteration. For stationary cascade com-

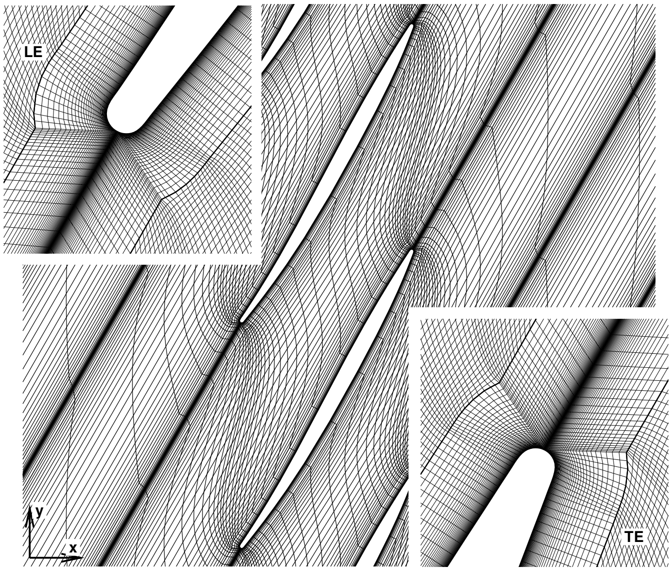


Figure 8.5: Full scale cascade 2D computation mesh

putation, the information exchanged across the domain partitioning boundary includes the primitive variables ($\rho$, $\mathbf{V}$, $e$) and the residual information. After each iteration, the coordinator processor gathers residual information from all subordinator processors and finds the universal maximum residual for the whole domain. When the universal maximum residual reaches the convergence criteria, the coordinator processor issues a termination order to all processors to terminate the computation. When local time stepping is turned off, the universal time interval for the whole domain is exchanged and determined similarly. When the blade is oscillating, the boundaries of the subdomains are also time dependent. The boundary mesh coordinates and the boundary moving velocities are exchanged through the domain partitioning boundaries.

The even load across the processors is achieved by applying the same mesh size on all subdomains. The baseline mesh size used in the computation is $195(\xi)\times180(\eta)$. The inlet and outlet boundaries are set as 1.5 and 3 times chord length away from the blade LE and TE in axial direction. A part of the mesh is shown in Fig. 8.5 with the regions of LE and TE zoomed in for more details. The H-type mesh is generated for each subdomain respectively.

Each subdomain shares the grid point distribution on the domain partitioning boundaries with its neighbors. As suggested in reference [14], an additional algebraic boundary layer mesh is generated in the wall surface region to achieve good orthogonality. The grid lines are made either parallel or normal to the solid surface. This is preferred by the Baldwin-Lomax model, in which the $F_{\max}$ and $y_{\max}$ based on Eq.(2.87) on page 2.87 is searched following a line normal to the solid boundary. As shown in Fig. 8.5, the grid lines are orthogonal on all blade surfaces except the small regions at LE and TE. For clarity, the mesh is plotted every 4 lines in the un-zoomed plots. The blade surface has 100 points in streamwise direction. The boundary layer has 40 points in pitchwise direction. Because of the high gradient of the flow variables in near wall region, the mesh is clustered near the wall surfaces. The grid points are also clustered toward the LE and the TE in streamwise direction.

When the blades vibrate at a prescribed frequency, the mesh in each subdomain moves accordingly. The boundary layer mesh is fixed to the solid surface and move with the blade. The inner mesh is recalculated by solving the elliptic equation. In the current study, the blades and the meshes vibrate harmonically. The mesh generation time is reduced by saving a series of meshes in a vibration cycle in advance and reusing them in the following vibration cycles. To save the storage, meshes in 20 uniform time intervals are generated before the simulation starts. The meshes at any time level during the vibration are linearly interpolated based on the previously-generated meshes.

## 8.3 Computation Configuration

To be consistent with the experiment [131], the simulation is configured as the following. All blades vibrate simultaneously along a pitching axis at 0.5 chord with a constant IBPA of 180°. The two neighboring blades always vibrate in opposite direction. The oscillating amplitude is 1.2° and the reduced frequency based on chord varies at $k_c$ =0.4, 0.8 and 1.2. The inlet Mach number is 0.5 and it is achieved by adjusting the outlet static pressure level.

The Reynolds number based on the chord length is $9\times10^5$. The flow incidences for the full scale simulation are 0° and 10°. In order to obtain good agreement with the steady state experimental surface pressure distribution, the inlet flow angles are adjusted to 61° in the cases of 0° incidence (60°flow angle) and 67.2° in the cases of 10° incidence (70°flow angle).

In order to study the end wall influence on the unsteady results, a simplified 2-passage cascade is computed before the full scale computation by applying the periodic boundary condition. The simplified cascade oscillates with frequency $k_c = 0.8$ and 0° incidence. The computation result mesh dependency is also studied. The multi-passage full scale simulation is then conducted for the 9-blade cascade with the wind tunnel end walls under 3 frequencies, $k_c$=0.4, 0.8 and 1.2 and 2 incidences, 0° and 10°, for more realistic results. The periodic boundary condition is not needed. The computation results are compared with the experiment measurement. The influence of the end walls are studied by comparing the the two cases with the same frequency ($k_c$=0.8) and incidence (0°) from both simulations.

For the unsteady dual-time stepping, one physical blade oscillation cycle is divided into 100 time intervals and 100 pseudo time Gauss-Seidel iterations are carried out for each physical time step. The 100 pseudo time iterations are sufficient to obtain a converged solution within a physical time step with the residual reduced by 3 orders in magnitude. Before the unsteady simulation, the corresponding steady state calculation is carried out to obtain the initial flow field for the unsteady computation.

## 8.4 Simulation in Two Passage Cascade

The two passage cascade simulation uses the meshes of two inner neighboring passages (P2 and P3) in the compressor cascade. As shown in Fig. 8.6 , The blade between the two passages is called BC (blade at center) and the two blade surfaces on the two outside periodic boundaries are treated as blade BP (blade at periodic boundary) .

The steady state pressure coefficient distributions along the blade surfaces are plotted

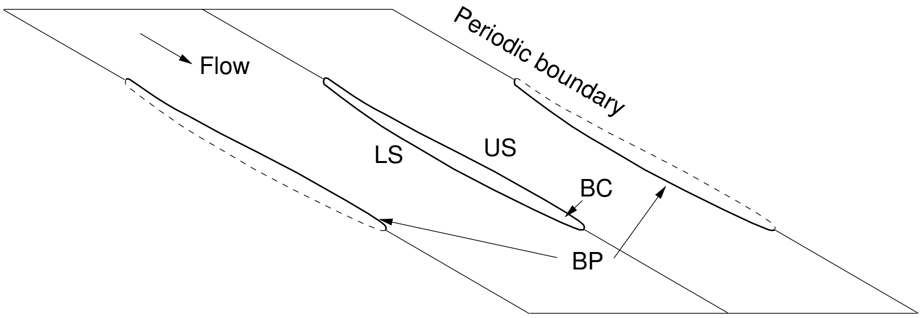


Figure 8.6: Two-passage cascade periodic computation domain

in Fig. 8.7 . The pressure coefficient prediction agrees well with the measurement. The result on BC agrees very well with that of BP, which shows good periodicity is achieved on the pitchwise direction.

The 2-passage cascade unsteady simulation is conducted with incidence 0° and $k_c$ =0.8. Fig. 8.8 shows the pressure variation history on two points on the suction surface (US) and the pressure surface (LS) respectively. The pressure on the suction surface is located at $x/C$ = 0.15 and the pressure on the pressure surface is located at $x/C$ = 0.1. The temporal periodicity is achieved very soon after the start of the vibration simulation. Because of the excellent temporal periodicity, the unsteady data extracted from a single blade motion cycle is enough for the unsteady Fourier transformation analysis. The IBPA of 180° is clearly shown by comparing the pressure maximums and minimums on BC and BP.

The unsteady pressure coefficients are plotted in Fig. 8.9 and Fig. 8.10 for suction surface (US) and the pressure surface (LS) respectively. The unsteady pressure coefficient $C_p$ is expressed in terms of the real part or in phase part and the imaginary part or out of phase part. On the suction surface, as shown in Fig. 8.9, the CFD results compare fairly well with experiment data after 30% chord. The real part of the coefficient is predicted lower than experiment data on leading edge, but the trend agrees very well with the experiment. The imaginary part is over predicted in the leading edge region. On the pressure surface, as shown in Fig. 8.10, the real part of the unsteady pressure coefficient agrees well with the experiment data. The imaginary part is under predicted compared with the measure-

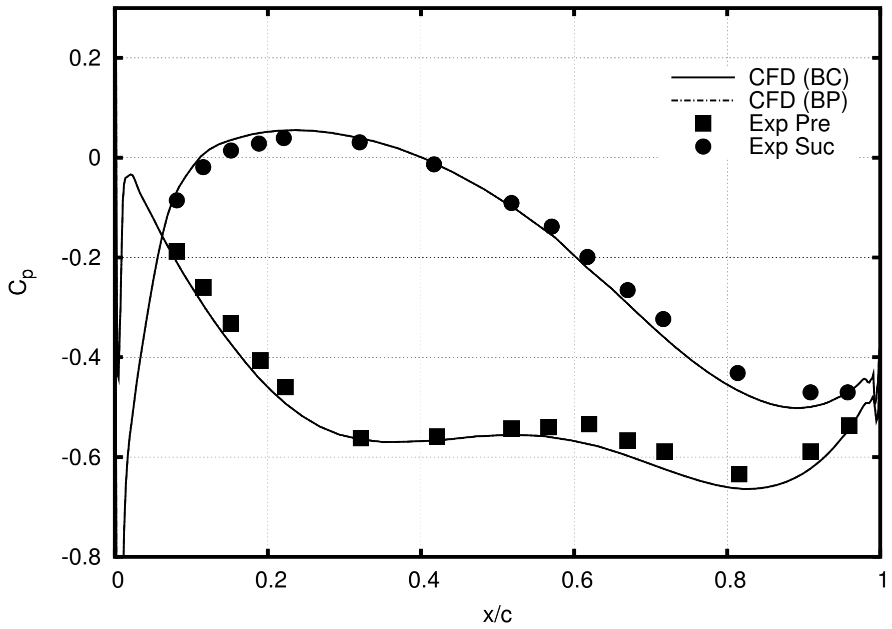


Figure 8.7: Steady state pressure coefficient in 2-passage cascade under low incidence

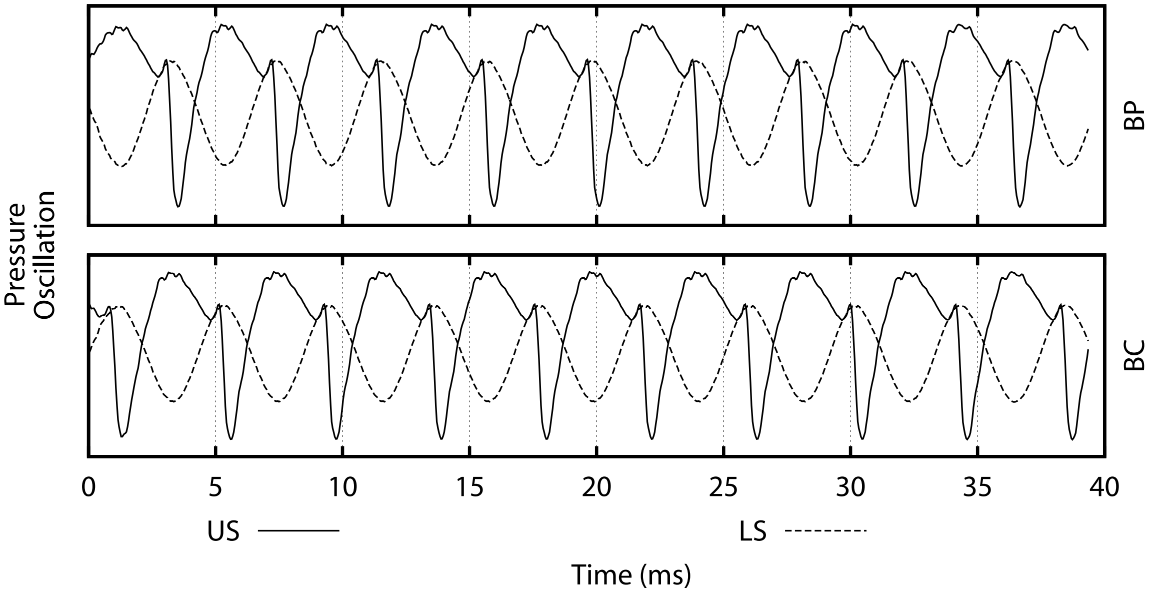


Figure 8.8: Pressure oscillation history in 2-passage cascade under low incidence and $k_c$= 0.8

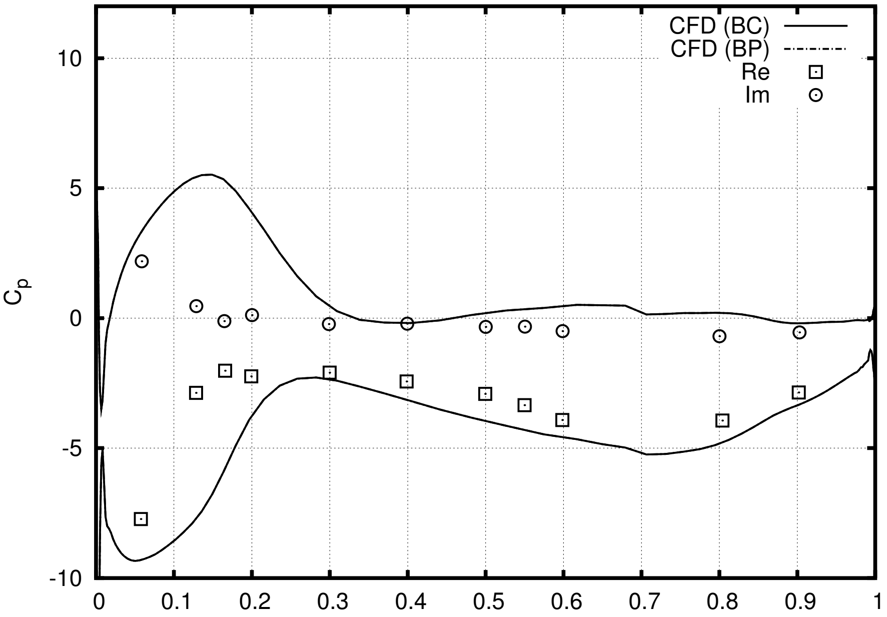


Figure 8.9: Unsteady pressure coefficient on upper surface in 2-passage cascade under low incidence, $k_c = 0.8$

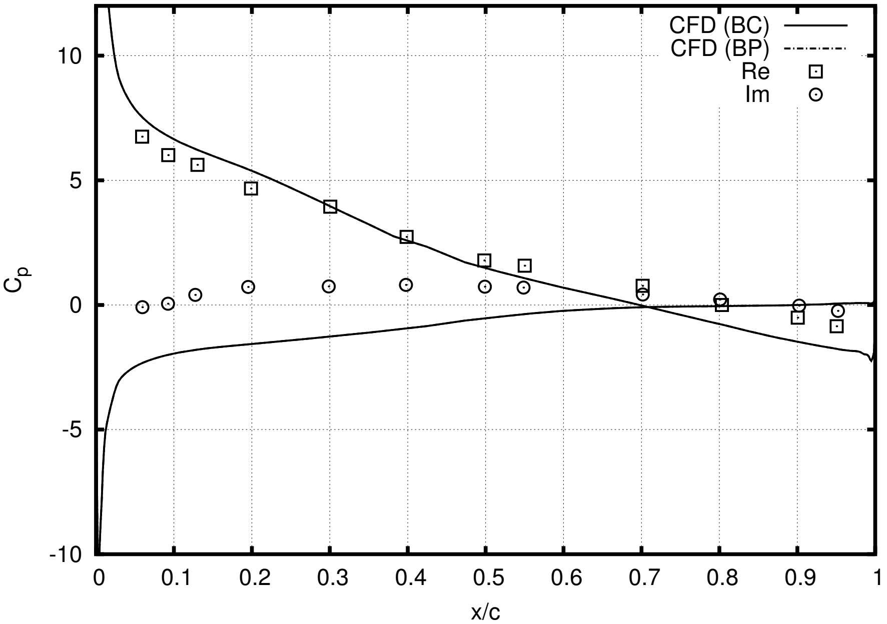


Figure 8.10: Unsteady pressure coefficient on lower surface in 2-passage cascade under low incidence, $k_c$= 0.8

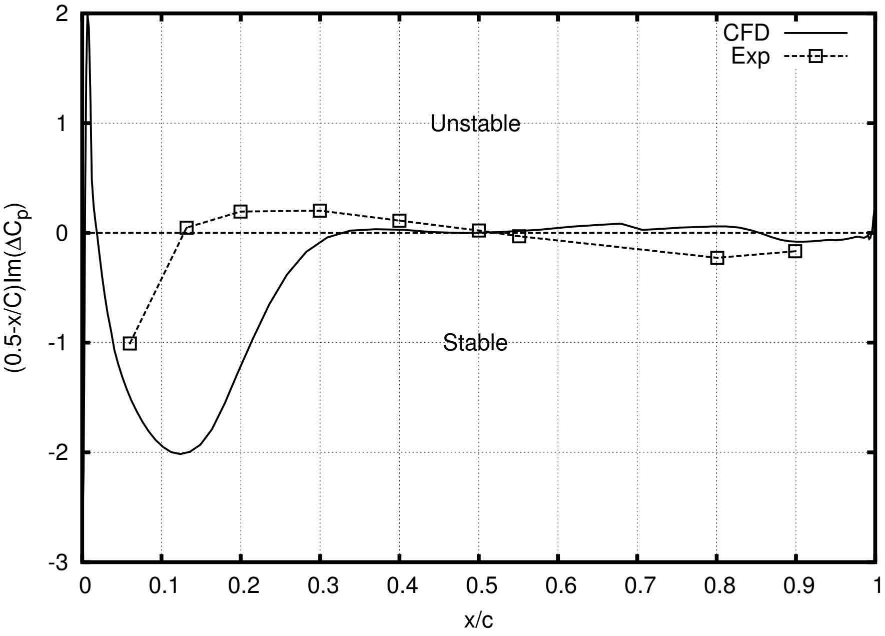


Figure 8.11: Local stability analysis in 2-passage cascade under low incidence, $k_c$=0.8

ment on the front part of the blade. This means that the CFD does not accurately capture the phase angle difference between the pressure response and the blade motion. A local flutter stability analysis based on the aerodynamic work per cycle chordwise distribution, calculated using Eq. (8.17) on page 137, is presented in Fig. 8.11 . The current numerical simulation predicts a larger local stable region on the front part of the blade. On the aft part, the experiment data indicates a shallow stable region. The CFD predicts the trend very well. The aft part stable region is predicted more shallow than the experiment.

## 8.5 Mesh Dependency

The Baldwin-Lomax turbulence model requires orthogonal mesh in the computation domain. This is partially achieved by generating an algebraic boundary layer orthogonal mesh on the blade surface in the current study. More numerical study on the computation mesh dependency is carried out for the simplified 2-passage cascade with $k_c$=0.8 and incidence 0°. The mesh size is adjusted in $\eta$ direction, with an emphasis on the boundary layer

Table 8.1: Mesh dependency study in 2-passage oscillating cascade under lower incidence

| mesh | $h_{\text{bnd}-\text{mesh}}$ | $N_{\text{bnd}-\text{mesh}}$ | $N_\eta$ | $\Xi$ |
|---|---|---|---|---|
| 4 | 0.00854 | 20 | 90 | 0.559 |
| 7 | 0.01139 | 40 | 180 | 0.556 |
| 9 | 0.02278 | 40 | 180 | 0.592 |
| 11 | 0.04556 | 80 | 360 | 0.527 |

algebraic orthogonal mesh property. The parameters considered in the study are the total mesh number in $\eta$ direction, the mesh cell number and the thickness in the boundary layer mesh. The mesh cell number in $\xi$ direction is kept constant as 195.

The meshes used for the dependency study are listed in Table. 8.1 . $h_{\text{bnd}-\text{mesh}}$ and $N_{\text{bnd}-\text{mesh}}$ are the thickness and mesh cell number of the boundary layer mesh. $N_\eta$ is the total mesh cell number in $\eta$ direction and $\Xi$ is the computed blade damping coefficient. The mesh used in the foregoing 2-passage cascade simulation is named as mesh 9. It is compared with two loose meshes, mesh 4 and mesh 7 and a dense mesh 11 in the following study. When the mesh cell number changes in the 4 meshes, the first inner cell close to the wall is kept the same height to make the comparison fair. This is achieved by applying appropriate stretch factor in $\eta$ direction.

As shown in Fig. 8.12 , the steady state computation results show very similar static pressure coefficient distribution in the 4 meshes. The discrepancy is negligible on most of the blade surface. Some small difference is found in the LE region on the suction surface ($x/C$=0.02-0.05).

The unsteady results are compared for the 4 meshes in Figs. 8.13 through 8.16 . These results are more sensitive to the mesh configuration than the steady state results. Fig. 8.13 shows the static pressure oscillation histories at the same location on the pressure surface (LS) as Fig. 8.8. The pressure oscillations coincide in dense meshes. The loose mesh 4 predicts slightly higher magnitude and average in the static pressure oscillation. The unsteady aeroelasticity analysis gives more comprehensive comparisons. As shown in Fig. 8.14, mesh 7, 9 and 11 give close results on the imaginary part of the upper surface unsteady

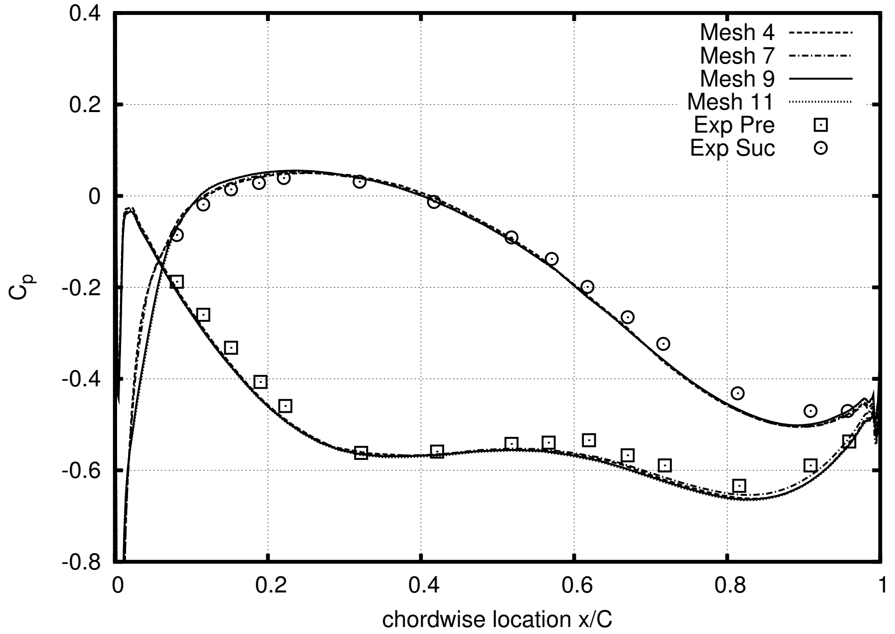


Figure 8.12: Steady state computation mesh dependency in 2-passage cascade under low incidence

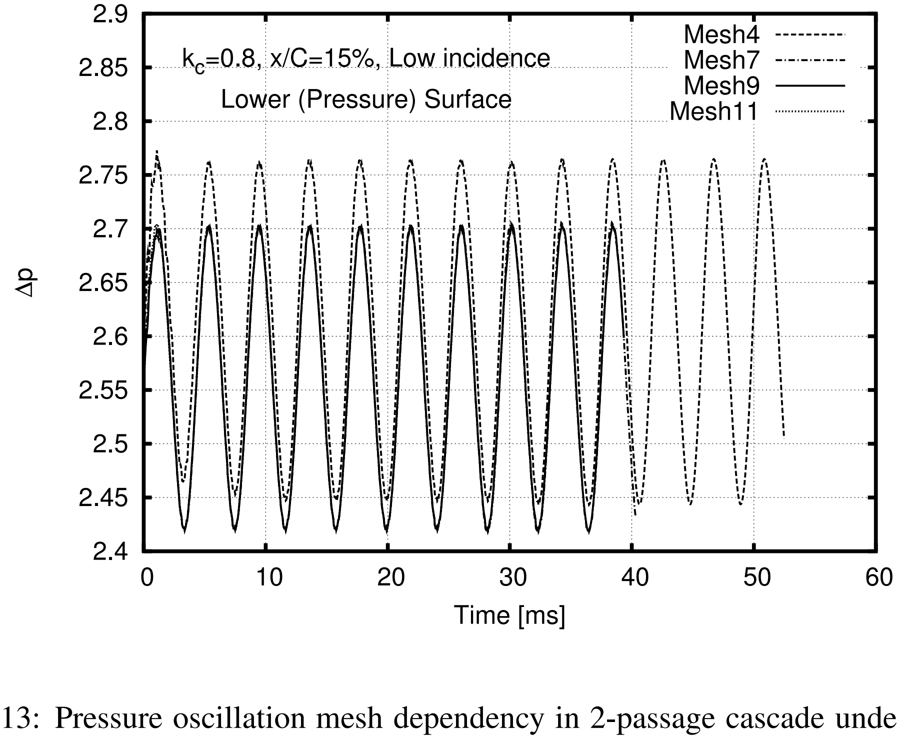


Figure 8.13: Pressure oscillation mesh dependency in 2-passage cascade under low incidence

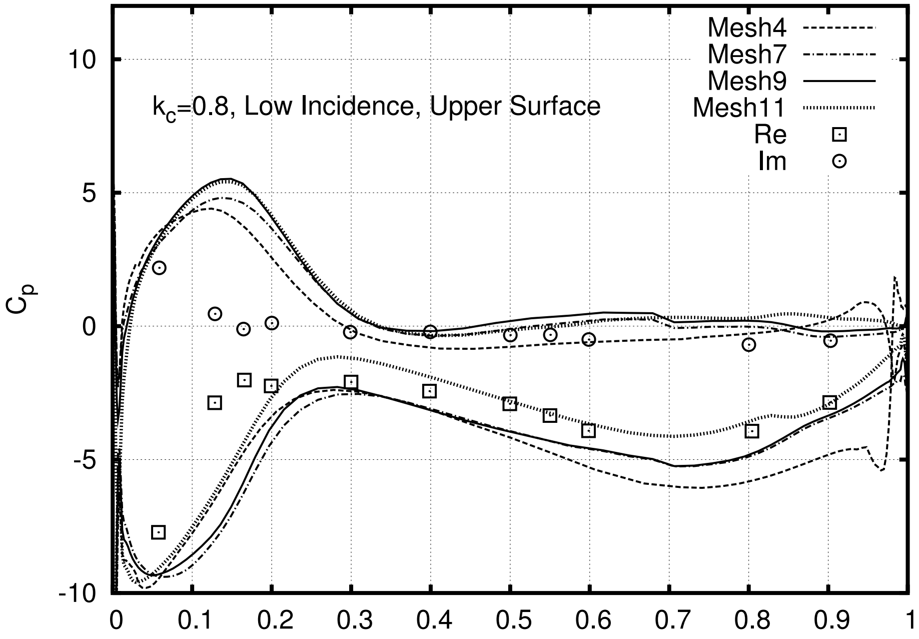


Figure 8.14: Upper surface unsteady pressure coefficient mesh dependency in 2-passage cascade under low incidence

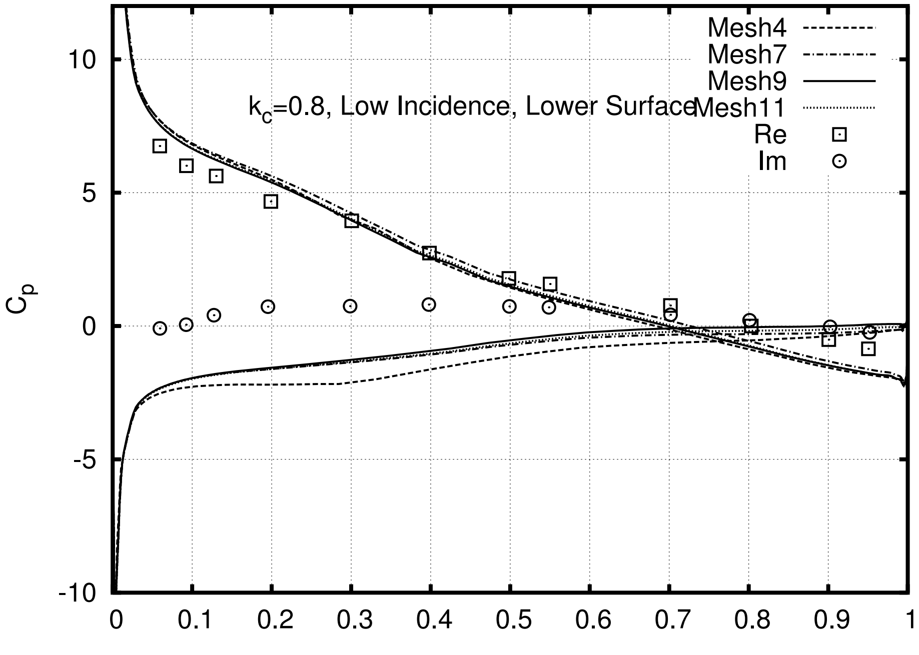


Figure 8.15: Lower surface unsteady pressure coefficient mesh dependency in 2-passage cascade under low incidence

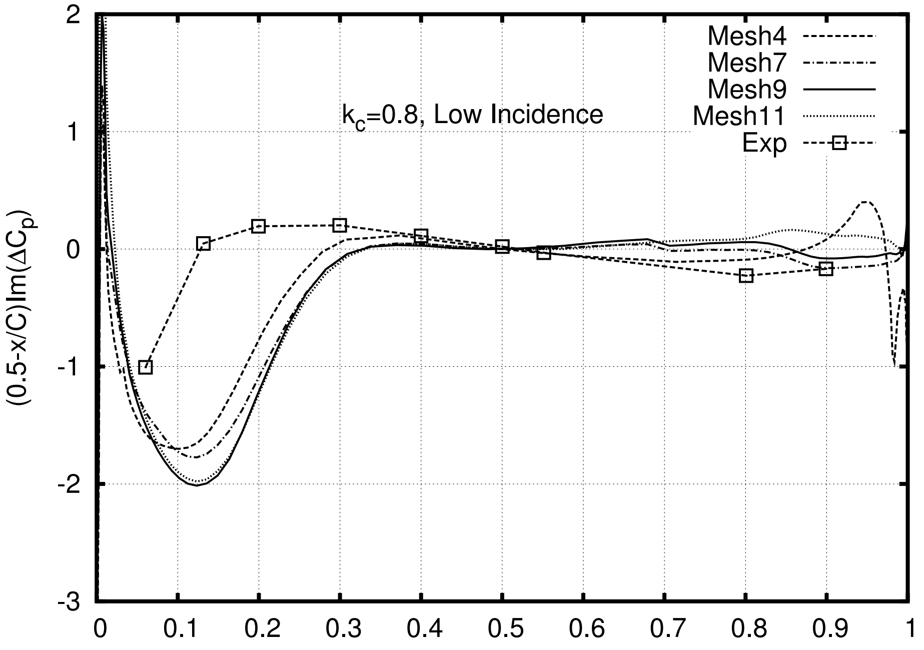


Figure 8.16: Local stability mesh dependency in 2-passage cascade under low incidence

pressure coefficient. The values predicted by the loose mesh 4 is a little lower on most of the chord. In the real part on the upper surface, mesh 7 and 9 gives similar results. The finest mesh 11 predicts higher imaginary pressure coefficient than the other 3 meshes and it is closer to the measurement. On the lower surface, as shown in Fig. 8.15, all 4 meshes give very similar results on the real part. The loose mesh 4 gives lower imaginary pressure coefficient than the other 3 meshes. The real and imaginary parts are similar on the other 3 meshes. The influence of the meshes on the blade local stability is shown in Fig. 8.16. In the front part of the blade, the 4 meshes predict similar trend in the local stability, which are all over predicted. In the aft part, the results based on mesh 4 are closest to the measurement at $x/C = 0.5 - 0.85$. In the TE region, mesh 7 and 9 predict stability, however mesh 11 predicts instability. Even though the finest mesh 11 works better in Fig. 8.14, it does not works as well as other 3 meshes in the local stability in Fig. 8.16. The damping coefficients computed from the unsteady numerical results are shown in Table. 8.1 for these meshes. The finest mesh 11 gives the lowest damping coefficient and the second finest mesh 9 gives the highest damping coefficient. No experimental damping coefficient

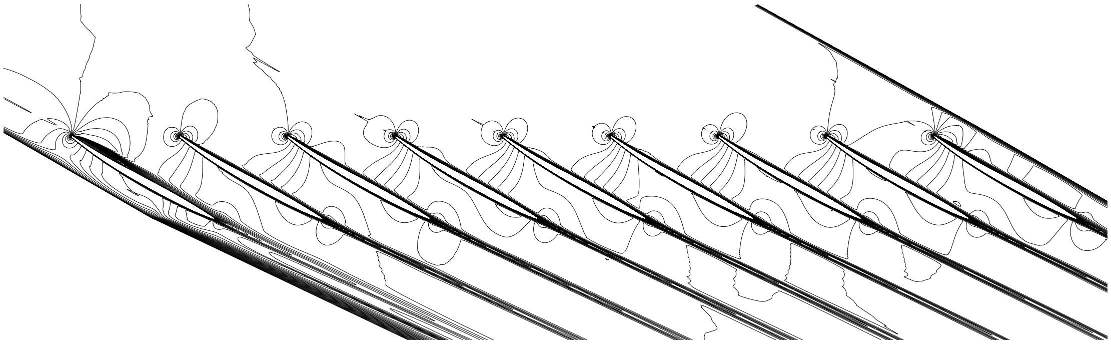


Figure 8.17: Steady state Mach contours in full scale cascade under low incidence

is available for comparison.

## 8.6 Simulation in Full Scale Cascade

### 8.6.1 Low incidence

The full scale steady and unsteady simulations use the same inlet flow angle as that of the 2-passage case. In the steady state calculation, all blade are parallel to each other at their mean positions. As shown in Fig. 8.4, the end wall is made up of 3 sections, which have different angles relative to the $x$ axis: $\alpha_1 = 61°$, $\alpha_2 = 60°$ and $\alpha_3 = 64°$. The middle section is parallel to the blade at its mean position. The front and aft sections follow the inlet and outlet averaged flow directions obtained in the 2-passage steady state computation.

The steady state Mach number contours for the full scale cascade is shown in Fig. 8.17 . The flow pattern is highly influenced by the end wall especially in the outer passages, P1 and P10. The influence is weakened quickly from the outer passages to the inner passages. Good periodicity in flow pattern is achieved among the inner passages (P3 through P8). Three center blades, B4, B5 and B6 are chosen to study the steady and unsteady periodicity in the rest of the paper. Even though the periodicity looks good in the Mach number contour plot, the static pressure distribution still shows the influence of the end walls on different blades. Fig. 8.18 shows the pressure coefficient chordwise distribution on the

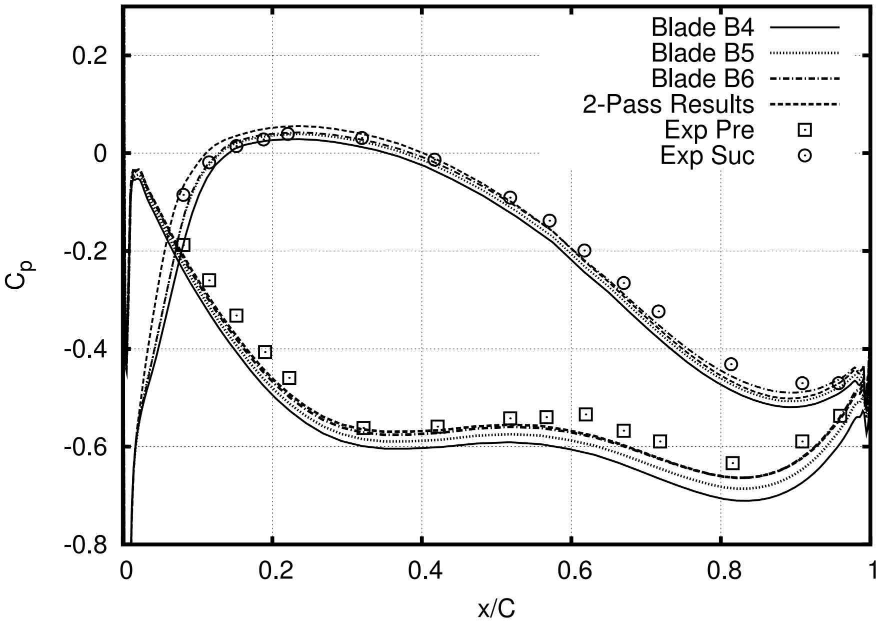


Figure 8.18: Steady state pressure coefficient in full scale cascade under low incidence

3 center blades. The experiment measurement and the 2-passage calculation results are also plotted for comparison. The surface pressure increases gradually from B4 to B6 on both the pressure surface and the suction surface. The pressure distribution on blade B6 is closest to the 2-passage periodic results on most part of the surfaces. The experiment measurement also shows the pressure variation on different blades [131], but its variation trend is opposite to the current numerical results. A possible reason for this difference is that the inlet and outlet end wall angles used in the experiment may differ from the values used in the current simulation. The experimental angles are not available. Such a pitchwise flow pattern difference is also expected in the following unsteady calculations.

The full scale unsteady simulation is first carried out for a reduced frequency $k_c = 0.8$ to study the end wall influence on the periodicity of the blade unsteady characteristics. Figs. 8.19 and 8.20 are the unsteady pressure coefficient chordwise distribution on the 3 center blades compared with the 2-passage cascade and the experiment results. On the upper surface, as shown in Fig. 8.19, the 3 blades have very similar unsteady coefficients on most of the chordwise distance. The results of blade B6 are closest to those of the 2-

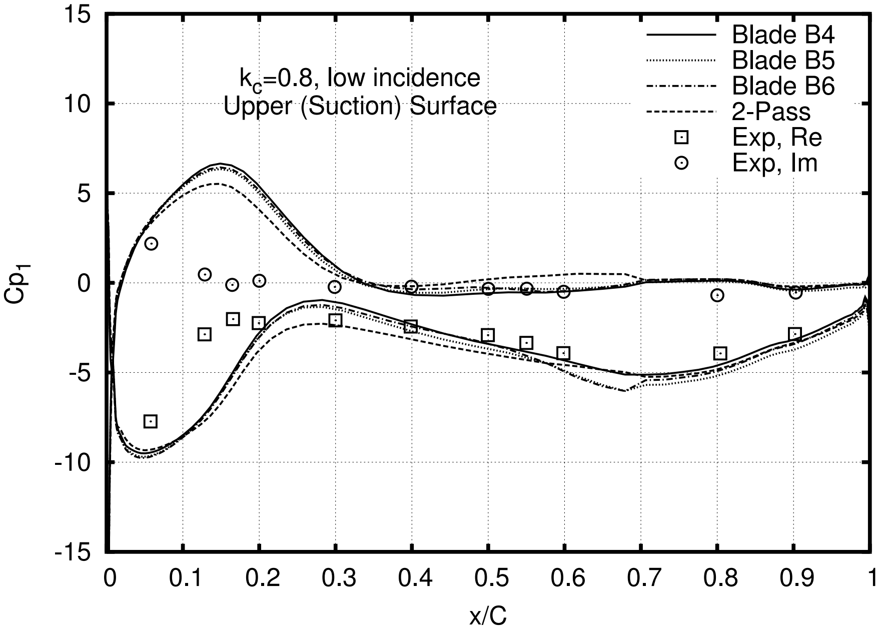


Figure 8.19: Unsteady state pressure coefficient on upper surface in full scale cascade calculation at low incidence, $k_c$ =0.8

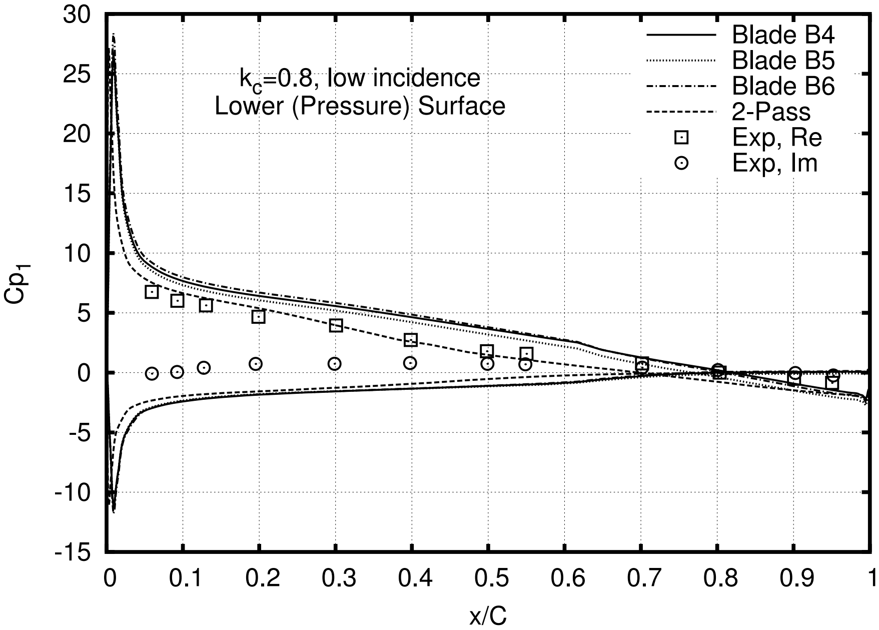


Figure 8.20: Unsteady state pressure coefficient on lower surface in full scale cascade calculation at low incidence, $k_c$ =0.8

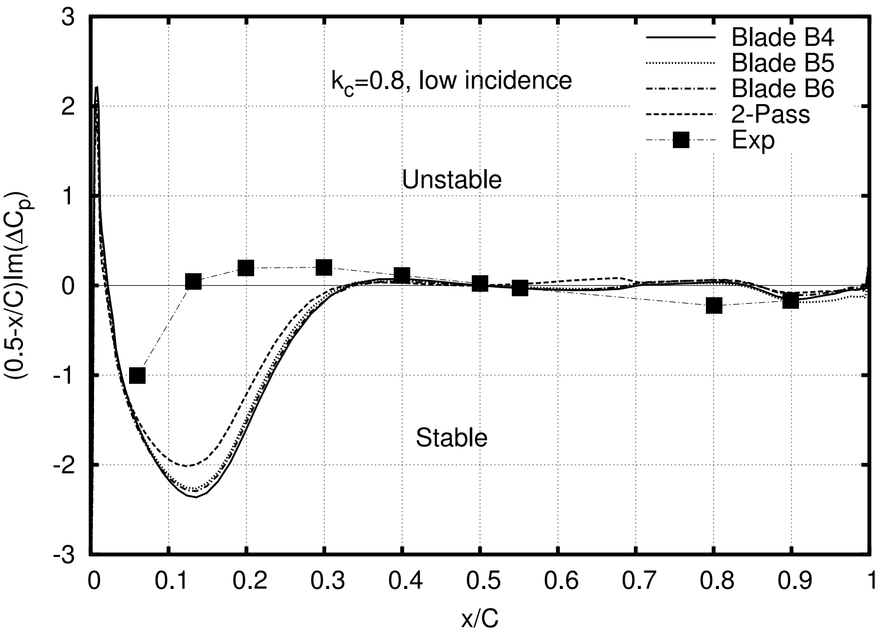


Figure 8.21: Local stability analysis in full scale cascade calculation at low incidence, $k_c$ =0.8

passage calculation. The difference between the full scale results and the 2-passage results mainly locate at the front and center part of the blade. The full scale results agree with the experiment better in the center part. On the lower surface, as shown in Fig. 8.20, the full scale results of the 3 blades are similar. The 2-passage results are closer to the experiment data in the real part. As shown in Fig. 8.21 , the full scale calculations predict higher stability on the front part of the blade compared with the 2-passage results. The full scale results are closer to the experiment measurement on the aft part of the blade. In the chordwise region of $x/C$ = 0.5 to $x/C$=0.7, the measured stability is better predicted in the full scale results. The 2-passage results shows instability in the same region. The end wall influence on the flow pattern periodicity is clearly shown in the unsteady aerodynamic moment oscillation plots within a single blade motion cycle in Fig. 8.22 . The moment is plotted versus the normal deflection angle, $\alpha' = \left(\alpha - \alpha_0\right)/\alpha_A$. Because of the end wall influence, the moment oscillations on the 3 center blades are different. They are also different from the 2-passage calculation results. The anti-clockwise direction of all the unsteady moment

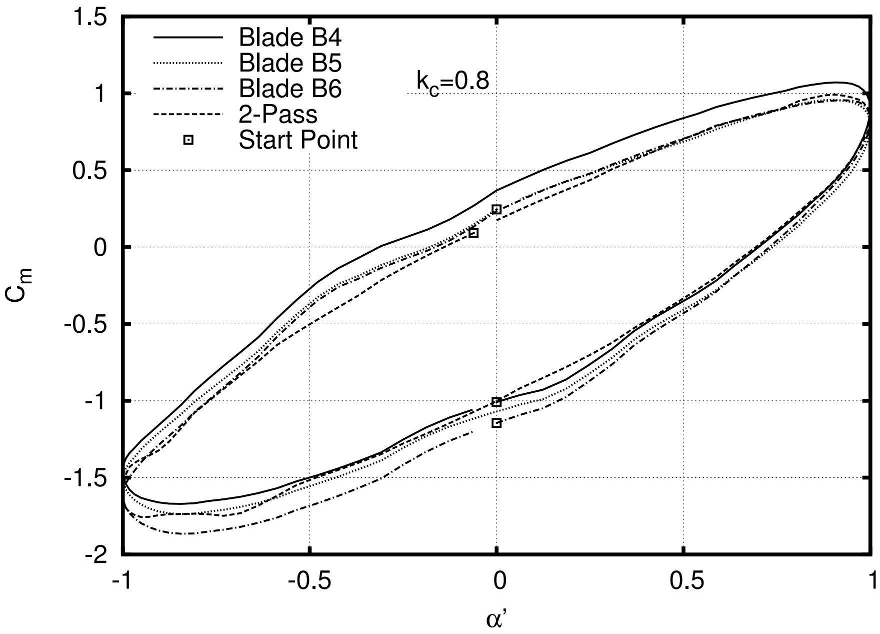


Figure 8.22: Unsteady aerodynamic moment oscillation in full scale cascade calculation at low incidence, $k_c$ =0.8

curves indicates negative work acted by the fluid on the blade. The blade motion is therefore damped down by the fluid flow. The blade motion is stable, which corresponds to a positive damping coefficient $\Xi$. The area enclosed by the moment curve indicates the magnitude of the work exchanged between the fluid and the blade, which is also proportional to the magnitude of the damping coefficient. The damping coefficients on all the 9 blades in the full scale calculation versus the blade number are shown in Fig. 8.23 . The damping coefficient varies among the blades. The damping coefficients for blade B4, B5 and B6 are 0.67, 0.65 and 0.68 respectively. Blade B1 has the lowest stability ($\Xi$ =0.45) and blade B9 has the highest stability ($\Xi$ = 1.4). The damping coefficient distribution is more uniform on the center blades (B3 through B7), even though small variation exists. The blade stability in the full scale cascade depends on the location of the blade. The damping coefficient obtained in the 2-passage periodic computation is 0.55.

Fig. 8.24 plots a series of Mach number contours around blade B5 and B6. A separation bubble is generated and grows periodically on the leading edge of the suction surface. At

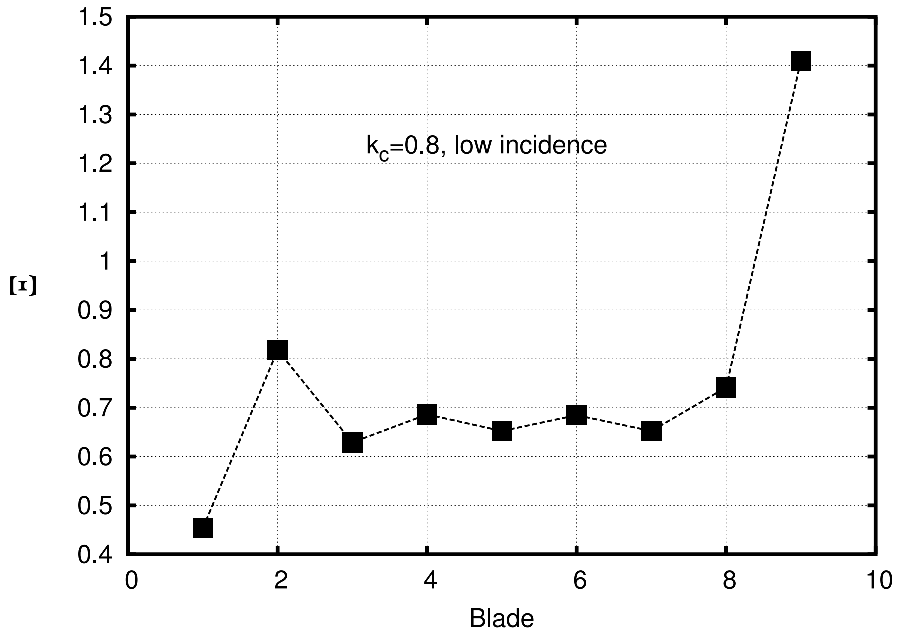


Figure 8.23: Damping coefficient distribution in full scale cascade calculation at low incidence, $k_c$ =0.8

$t$ =0, the two blades are initially located at their mean positions and are parallel to each other. Blade B5 then rotates in the anti-clockwise direction with a negative deflection angle (nose down). At the same time, blade B6 is rotating in the clockwise direction with a positive deflection angle (nose up). At $t$=0.2T, blade B5 is close to its minimum deflection position. The separation bubble at its LE is pushed downstream and shrinks in size. At $t$=0.4T, blade B5 is rotating back from its minimum deflection location towards its mean position, the separation bubble disappears from the suction surface. At $t$ = 0.8T, a new separation bubble is generated when blade B5 passes its maximum deflection position and rotates back toward its mean position. The bubble obtains its maximum size when the blade is close to its mean position. Similar phenomenon is observed on the neighboring blade B6, but with a phase difference of 180°.

More extensive unsteady simulations are carried out for reduced frequencies $k_c = 0.4$ and $k_c$ = 1.2. Figs. 8.25 and 8.26 show the unsteady pressure coefficient chordwise distribution of $k_c = 1.2$. Similar to the results of $k_c = 0.8$, the predicted unsteady complex

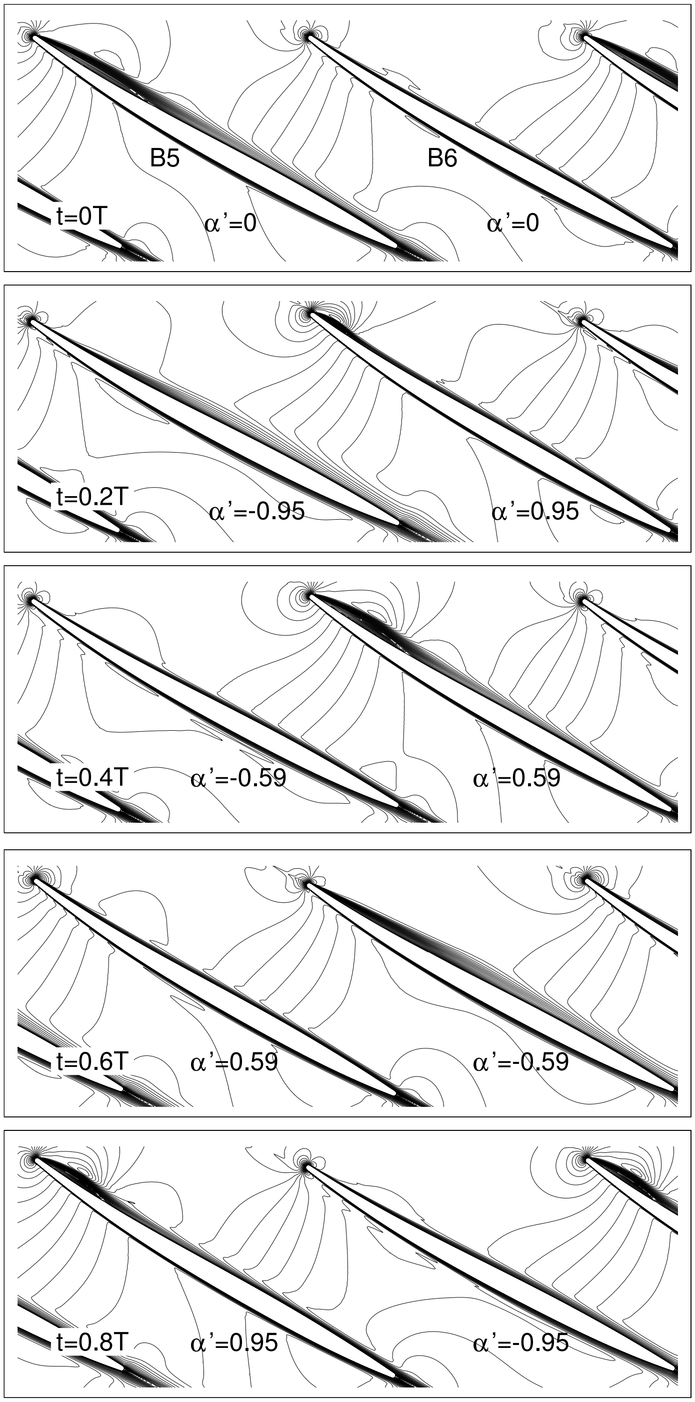


Figure 8.24: Unsteady Mach number contours in full scale cascade calculation at low incidence, $k_c = 0.8$

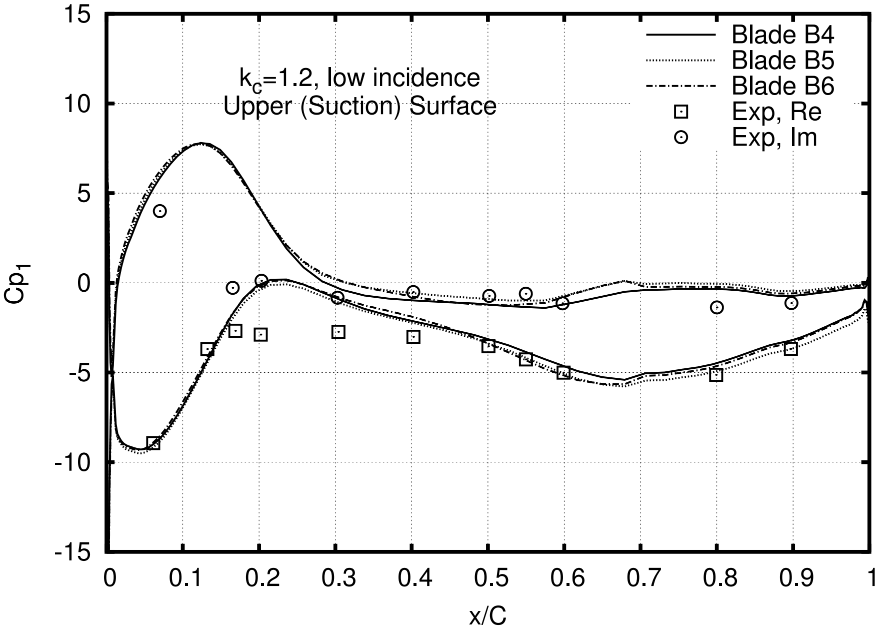


Figure 8.25: Unsteady pressure coefficient on upper surface in full scale cascade calculation at low incidence, $k_c$ =1.2

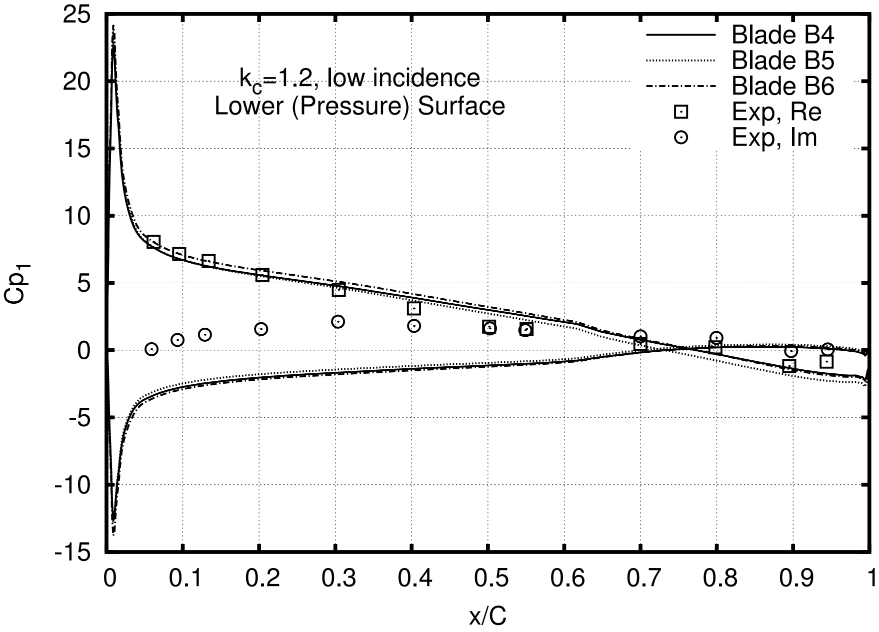


Figure 8.26: Unsteady pressure coefficient on lower surface in full scale cascade calculation at low incidence, $k_c$ =1.2

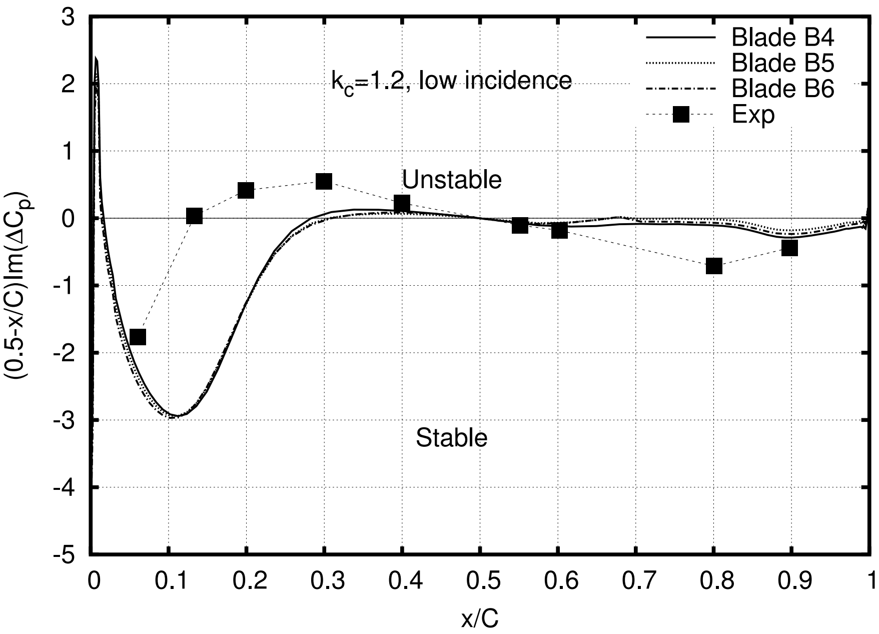


Figure 8.27: Local stability analysis in full scale cascade calculation at low incidence, $k_c$ =1.2

pressure coefficients are close to each other on the 3 center blades, even though some small difference exists. As shown in Fig. 8.25, on the upper surface, the imaginary parts of CFD results agree very well with the experiment results except that it is over-predicted in the region of $x/C$=0.15 to $x/C$=0.40. The real part is also predicted quite well on the middle and aft part of the blade. On the lower surface, as shown in Fig. 8.26, the predicted real parts compare very well with experiment. The Imaginary part is under-predicted from LE to $x/C$ = 0.7. The local stability analysis for $k_c$ = 1.2 is plotted in Fig. 8.27 . The correct trend is predicted compared with the experiment measurement, even though the magnitude does not agree very well. The stability is over-predicted in LE region. The unstable region predicted on the front part of the blade is smaller than the experiment results. The stability is predicted on the aft part, but the magnitude is smaller than the experiment data. The damping coefficients for all the blades are plotted in Fig. 8.28 . The damping coefficients for blade B4, B5 and B6 are 0.81, 0.78 and 0.84 respectively. The stability increases with the frequency. Similar to the results of $k_c$ = 0.8, the most stable blade is blade B9 ($\Xi$=1.5)

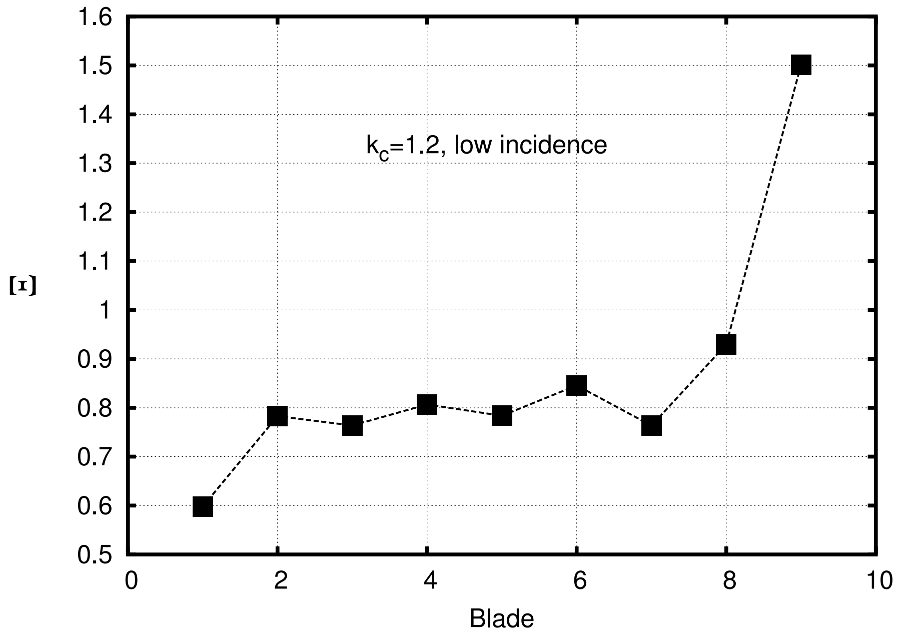


Figure 8.28: Damping coefficient distribution in full scale cascade calculation at low incidence, $k_c$ =1.2

and the least stable blade is blade B1 (Ξ=0.6). The damping coefficient is more uniformly distributed on the central blades. The variation of the damping coefficient on the center blades increases with the increasing frequency.

Because of the lack of experiment data, the unsteady pressure coefficient of $k_c$=0.4 is not presented for comparison. However, the local stability is analyzed and compared with the experiment data in Fig. 8.29 . Similar to the results of $k_c$ = 0.8 and $k_c$ = 1.2, the trend is predicted well, but the magnitude differs from the experiment. As expected, the damping coefficient distribution is more uniform on center blades (Fig. 8.30 ). The variation of their magnitudes decreases with the decreasing vibration frequency compared with the high frequency cases of $k_c$ = 0.8 and $k_c$ = 1.2. The damping coefficients on blade B4, B5 and B6 are 0.448, 0.446 and 0.434 respectively. The most stable blade is B9 with Ξ = 1.02 and the least stable blade is B1 with Ξ = 0.28.

The unsteady aerodynamic moment oscillations on blade B5 under the 3 frequencies under investigation are plotted together and compared in Fig. 8.31 . The damping coeffi-

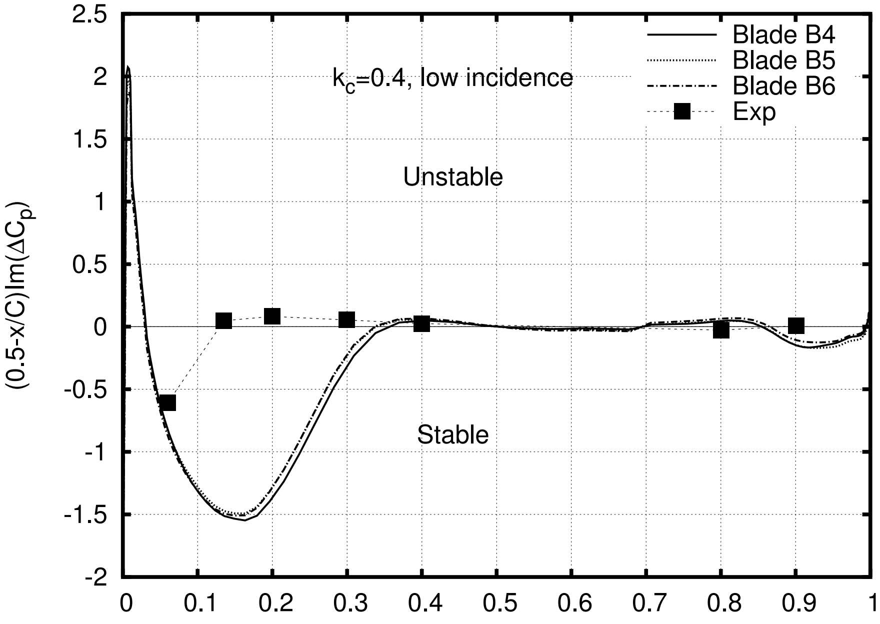


Figure 8.29: Local stability analysis in full scale cascade calculation at low incidence , $k_c$ =0.4

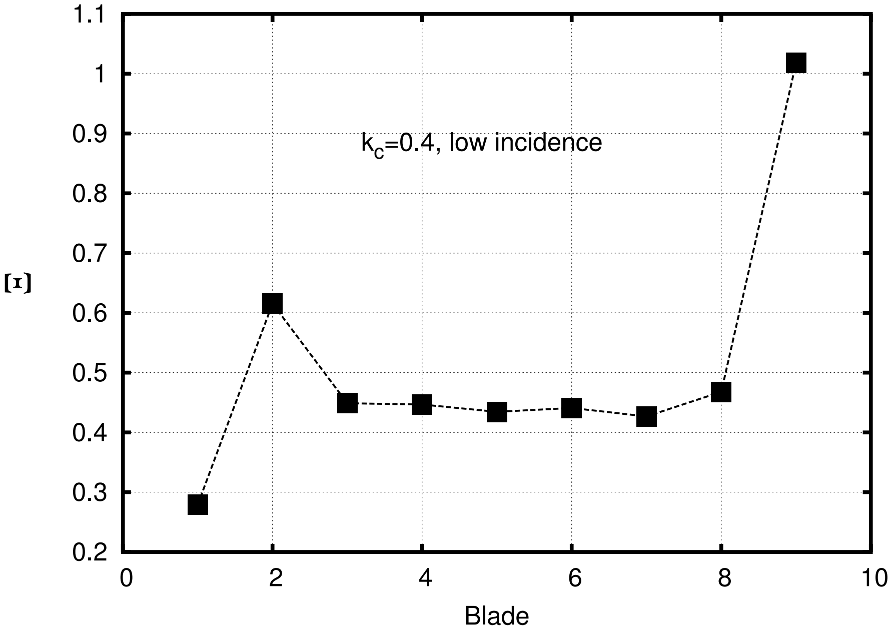


Figure 8.30: Damping coefficient distribution in full scale cascade calculation at low incidence, $k_c$ =0.4

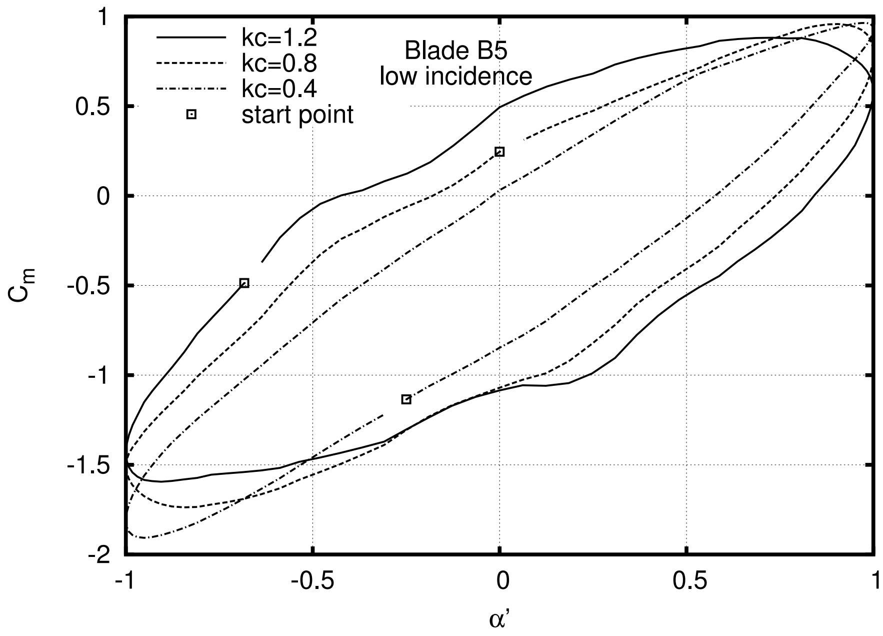


Figure 8.31: Unsteady aerodynamic moment oscillation comparison in full scale cascade calculation at low incidence

cient increasing with the increased frequency is indicated by the increased area enclosed by the unsteady moment oscillation curves. Fig. 8.32 shows the damping coefficients on the 9 blades all increase with the increased frequency. The damping coefficient is roughly uniform on the central blades, mainly from B3 through B7, under the 3 frequencies. The local stability analysis is summarized for all the 3 frequencies in Fig. 8.33 . The computation results indicate higher stability near the leading edge for higher frequency vibration, which is consistent with the experiment measurement. Even though the destabilization region on the front part of the blade and the stability magnitude on the aft part of the blade predicted by the numerical computation are smaller than those in the experiment, the trend is predicted well. Both the destabilization and stabilization increase with the increased frequency.

The static pressure oscillation histories on two points on blade B5 and B6 are shown in Fig. 8.34 . These two points are located at $x/C$=0.15 on the suction surfaces of B5 and B6. Very similar to the simplified two-blade cascade pressure oscillation history in Fig. 8.8, the pressure oscillation in full scale computation reaches good periodicity very quickly in

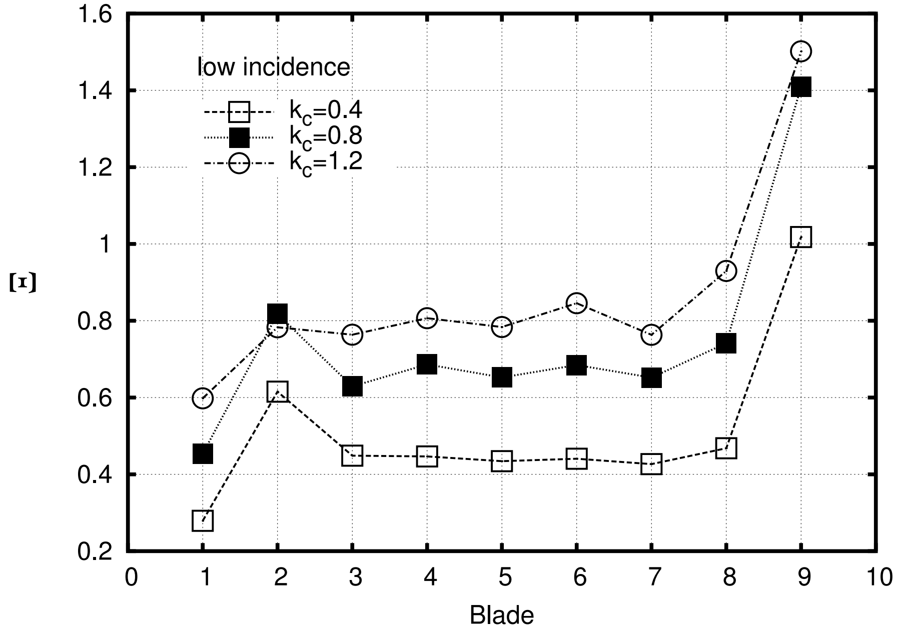


Figure 8.32: Damping coefficient comparison in full scale cascade calculation at low incidence

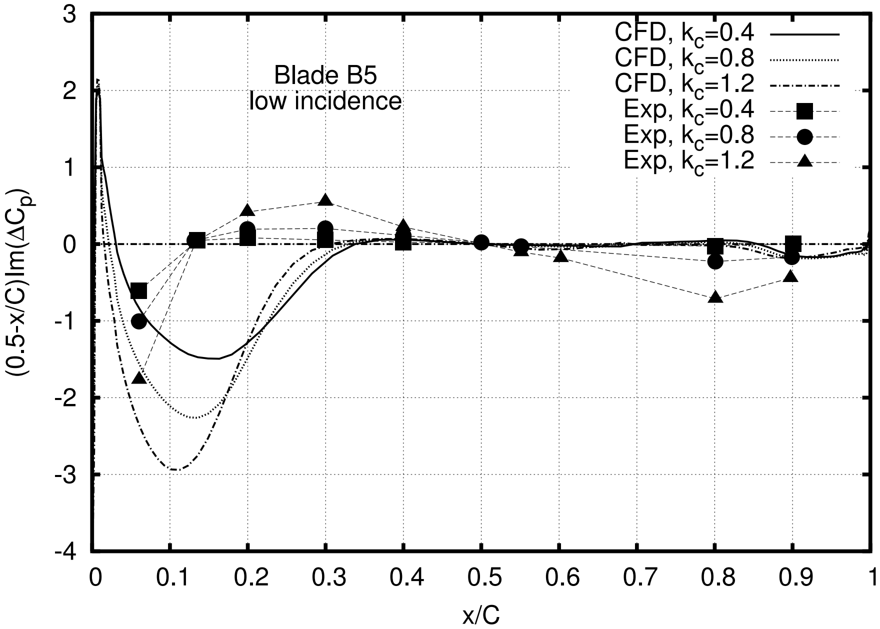


Figure 8.33: Local stability analysis comparison in full scale cascade calculation at low incidence

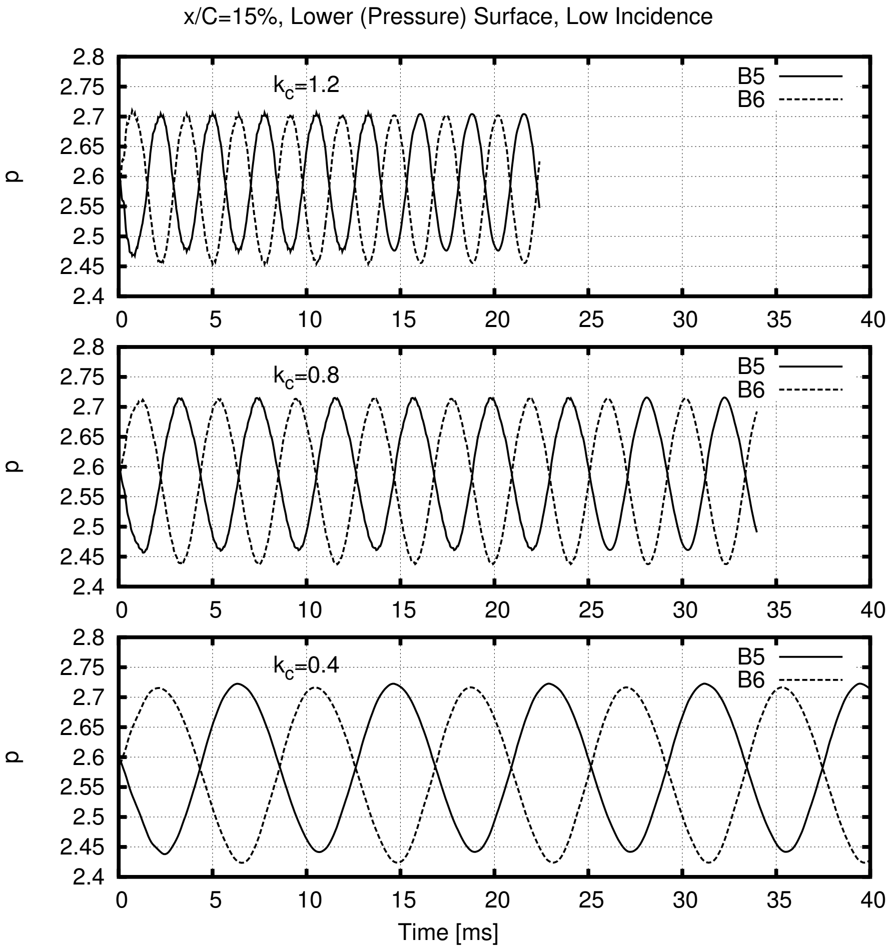


Figure 8.34: Pressure oscillation under different frequencies in full scale cascade calculation at low incidence

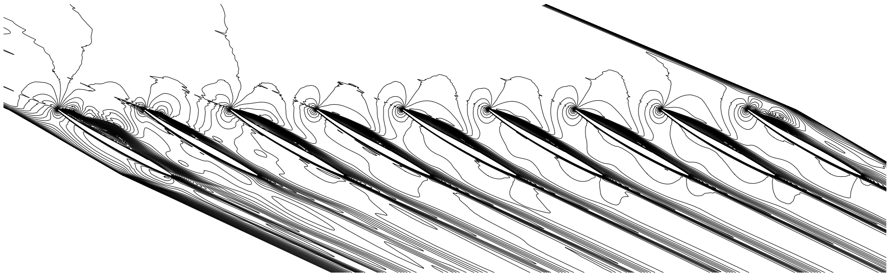


Figure 8.35: Steady state Mach contours in full scale cascade at high incidence

1-2 blade motion cycles. However, the pressure oscillation magnitude is larger on blade B6 than that on blade B5. The pressure oscillation magnitude slightly decreases with the increased frequency.

### 8.6.2 High incidence

At high incidence cases, the nominal incidence in the experiment report is 10°. In the current numerical investigation, just like the low incidence cases, a single passage case with periodic boundary condition in pitchwise direction is first computed under steady state to find the flow angles and the back pressure. After several iterations, a flow angle of 67.2° is finally selected at the inlet. The corresponding outlet flow angle is 65°. These two angles are applied at the upstream and downstream parts of the end walls to make the inner flow go through the cascade smoothly and keep the flow angles in full scale cascade close to the single passage case.

The steady state Mach contours under high incidence are plotted in Fig. 8.35 . The flow separates on the upper (suction) surfaces of the 9 blades. The scale of the separation varies among the blades. The maximum separation size is shown on blade B1. The flow pattern looks periodic in the inner passages especially from blade B3 through B7. However, the flow pattern periodicity is not as good as the low incidence case. The steady state pressure coefficient distribution on 3 center blades, B4, B5 and B6 are plotted in Fig. 8.36 . Compared with the periodic computation results, the full scale computation predicts lower

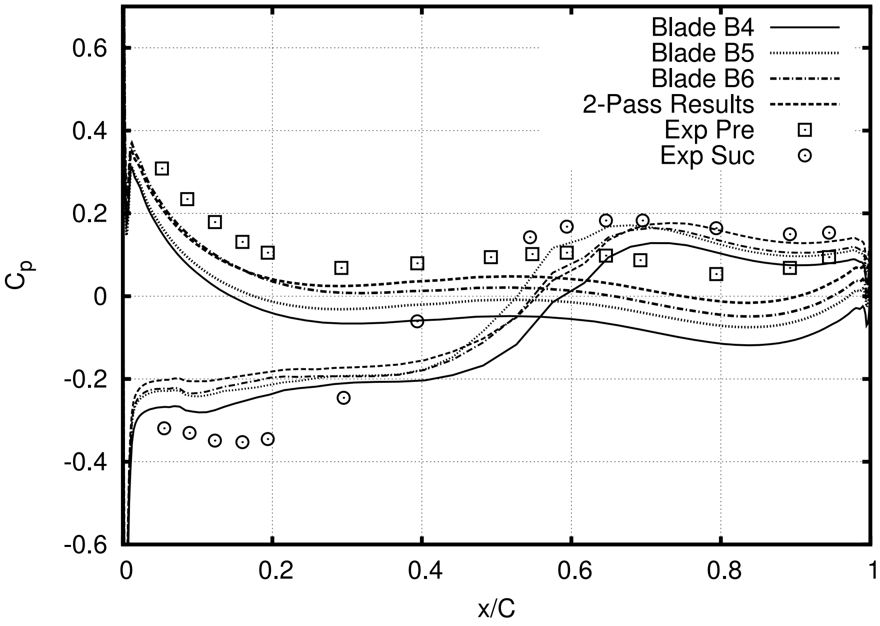


Figure 8.36: Steady state pressure coefficient in full scale cascade at high incidence

pressure on all the 3 blades on both pressure surface and suction surface. The results on blade B6 are closest to the periodic results. The flow pattern is more influenced by the existence of the end walls in the high incidence case than that in the foregoing low incidence case. The separation increases the interaction between the inner flow and the end walls, which makes the inner flow more sensitive to the end walls. The flow pattern periodicity is expected to be improved by adjusting the end wall configuration more carefully.

The full scale cascade at high incidence is then simulated in unsteady cases at 3 frequencies. The static pressure oscillation histories at 15% $x/C$ on the lower (pressure) surface on two central blades are plotted in Fig. 8.37 . Compared with Fig. 8.34, the pressure oscillation temporal periodicity becomes poor when the flow incidence is increased. This is caused by the large separation on the suction surface. The flow becomes more unsteady because of the separation. When the frequency increases, the temporal periodicity becomes even worse. The pressure oscillation amplitude decreases with the increasing frequency. The 180°IBPA is more clearly shown by the low frequency plot with $k_c$ =0.4. These 3 cases start from the same steady state flow field. The pressure oscillation approaches peri-

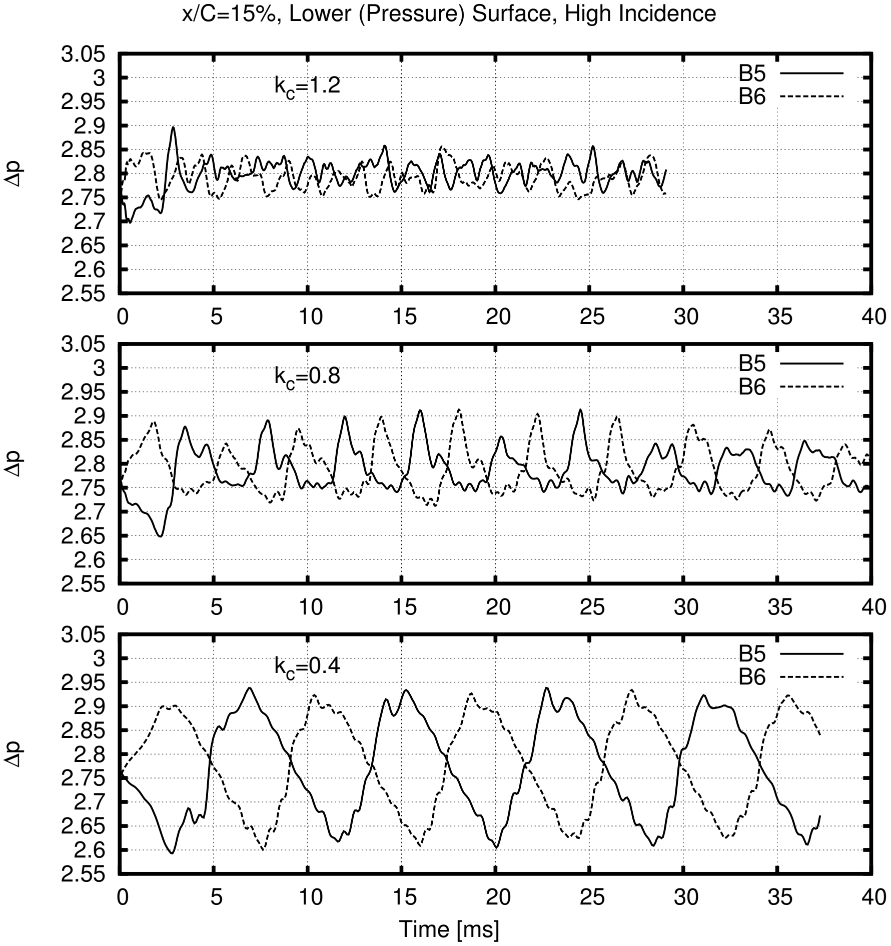


Figure 8.37: Pressure oscillation under different frequency in full scale cascade at high incidence

odicity more quickly in the low frequency case ($k_c$=0.4).

Just like the periodic variation of the separation shown in Fig. 8.24 at low incidence, a separation bubble in shown on the suction surface on each blade and varies periodically with the blade motion. The Mach contours under frequency $k_c = 0.8$ around B5 and B6 in a single motion cycle are plotted in Fig. 8.38 . The separation bubble is generally larger in the high incidence case than that in the low incidence case. Even though large separation bubble exists in the blade suction surface at LE region, the bubble may disappear at certain blade position because of the unsteady flow incidence. As shown in Fig. 8.38, the two blades start from the same mean positions, but move toward opposite direction. The bubble is clearly shown on the suction surface of B5, but almost invisible on B6. When blade B5 rotates nose down (anti-clockwise) to $\alpha'$=-0.95 at $t$ = 0.2T, the blade is close to its minimum deflection position, the separation bubble is pushed toward downstream and the bubble center is closer to the mid-chord location. The size of the bubble does not change much. When blade rotates further and passes its minimum location to $\alpha'$ =-0.59 at $t$ =0.4 T, the separation bubble disappears from the blade surface. The low velocity flow dominates on the blade suction surface region. At $t$ = 0.6T, blade B5 has passed its mean position and is moving toward its maximum deflection position. The low velocity region on the suction surface decreases in size even further. A new separation bubble is generated at the B5 LE when the blade passes its maximum deflection position and begins to move back to its mean position at $t$ = 0.8T. The size of the bubble increases when the blade is back to its mean location. Generally speaking, the separation bubble is generated when the blade passes its maximum location and moves toward its mean location. The bubble size increases with the decreased deflection (or incidence) angle. The bubble is then shed downstream when the blade passes its mean location and moves toward its minimum position. Meanwhile, the bubble loses its strength and finally disappears from the suction surface. The low velocity region on the suction surface decreases when the blade rotates nose up from its minimum deflection position. A similar phenomenon repeats on the neighboring blade B6, but with a phase angle difference of 180°. The periodic variation of the separation bubble brings even

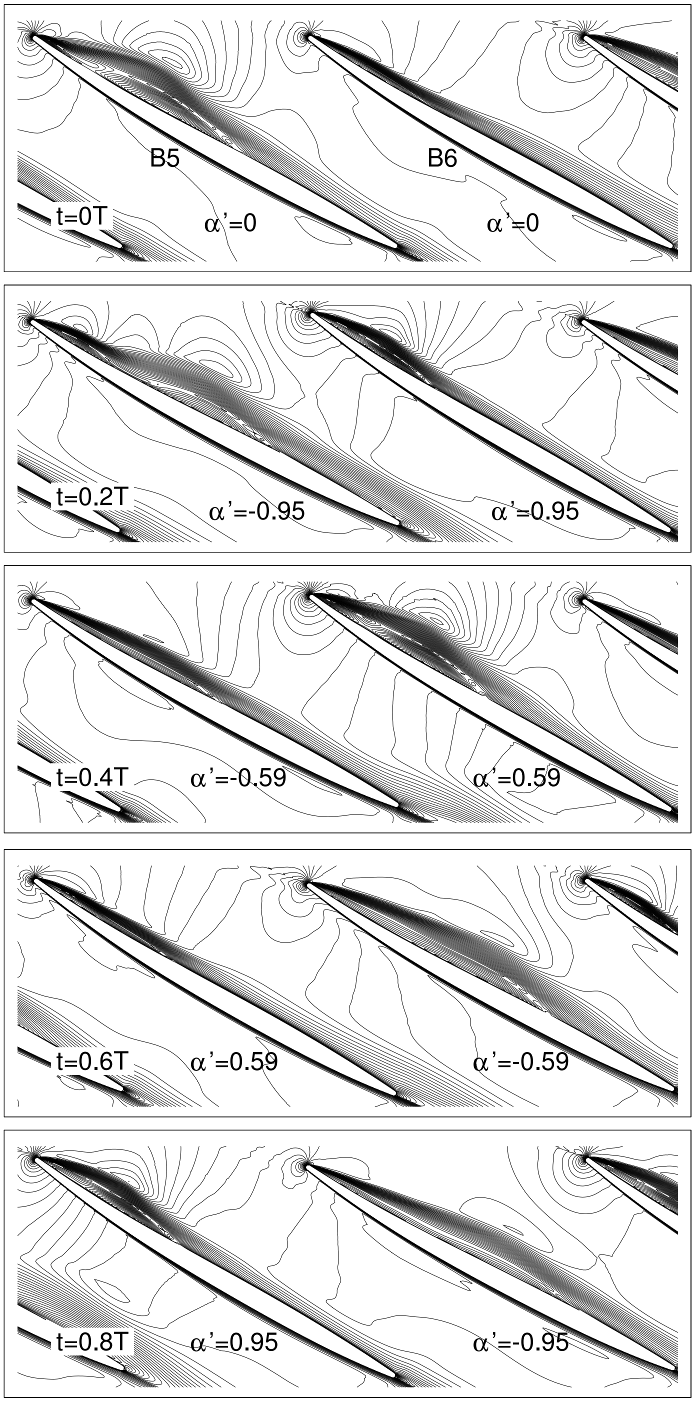


Figure 8.38: Unsteady Mach contours in full scale cascade at high incidence, $k_c = 0.8$

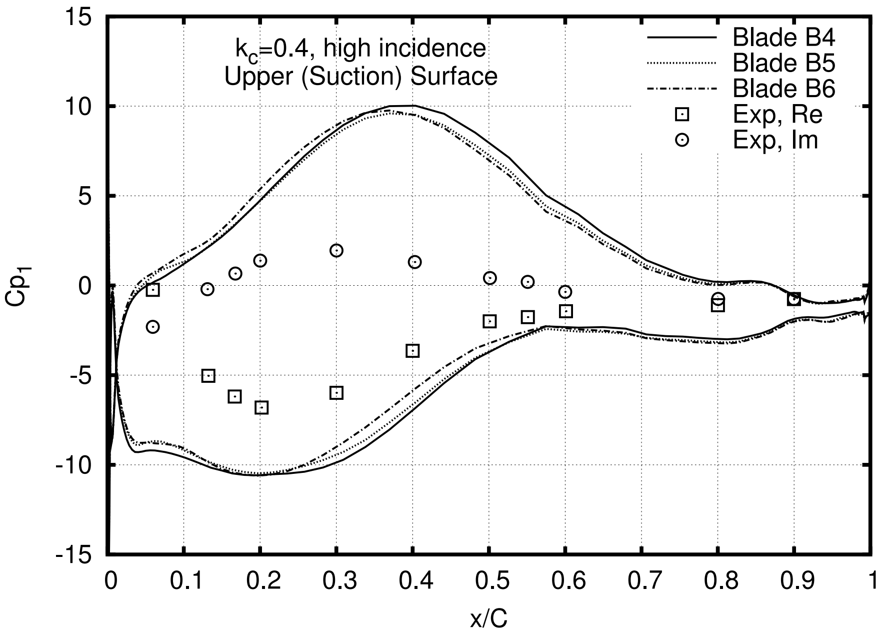


Figure 8.39: Upper surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 0.4$

higher influence on the unsteady flow pattern in the cascade. The unsteady performance will differ from that in the low incidence cases.

The unsteady stability analysis is conducted for the high incidence cases under the 3 frequencies in Figs. 8.39 through 8.48. Because of the periodicity of the calculation results is not as good as that in the low incidence cases, the numerical unsteady pressure in the last 4 cycles are averaged for the Fourier transform analysis.

In the low frequency case of $k_c = 0.4$, the stability analysis is shown in Figs. 8.39 through 8.41 . The unsteady pressure coefficient is predicted very well in trend on both the upper surface and the lower surface. However the magnitude does not compare very well. On the upper surface, as shown in Fig. 8.39, the imaginary parts compares well with the experiment results on LE and TE region, but are over predicted at mid-chord regions. The real part is predicted well on the aft part of the blade, but under predicted in the front part. The numerical results for the 3 center blades compare well with each other, which indicates the good spatial periodicity in obtained. On the lower surface, as shown in Fig. 8.40, the

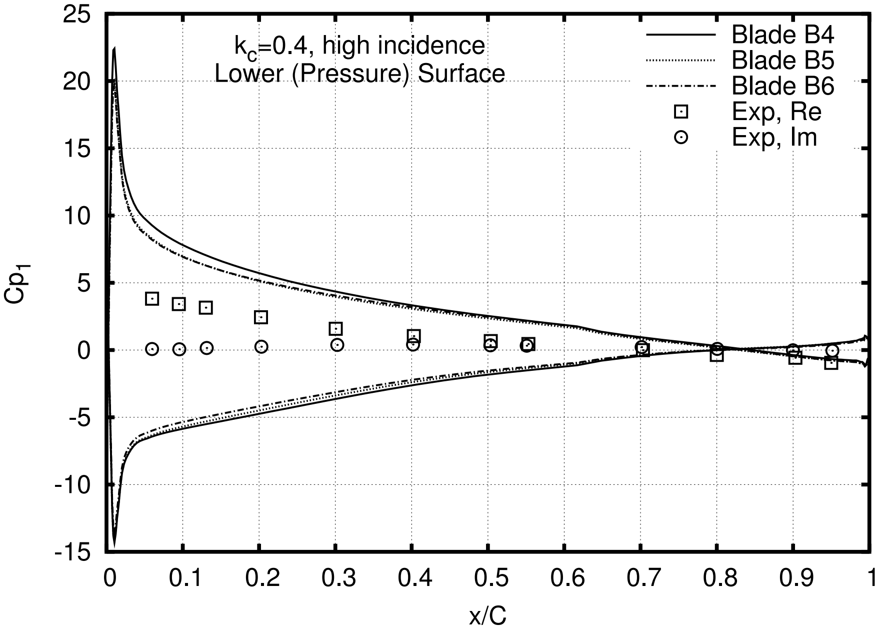


Figure 8.40: Lower surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 0.4$

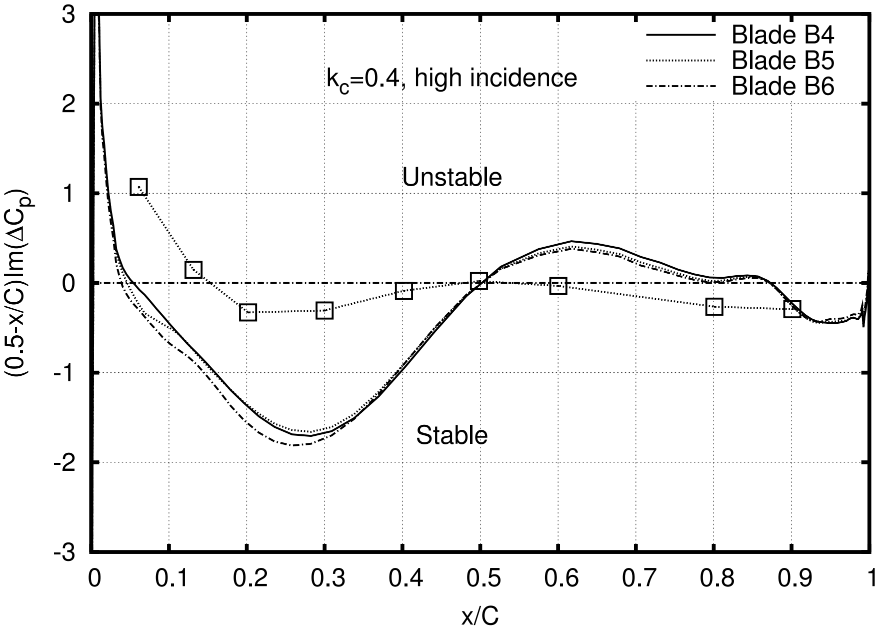


Figure 8.41: Local stability analysis in full scale cascade at high incidence, $k_c = 0.4$

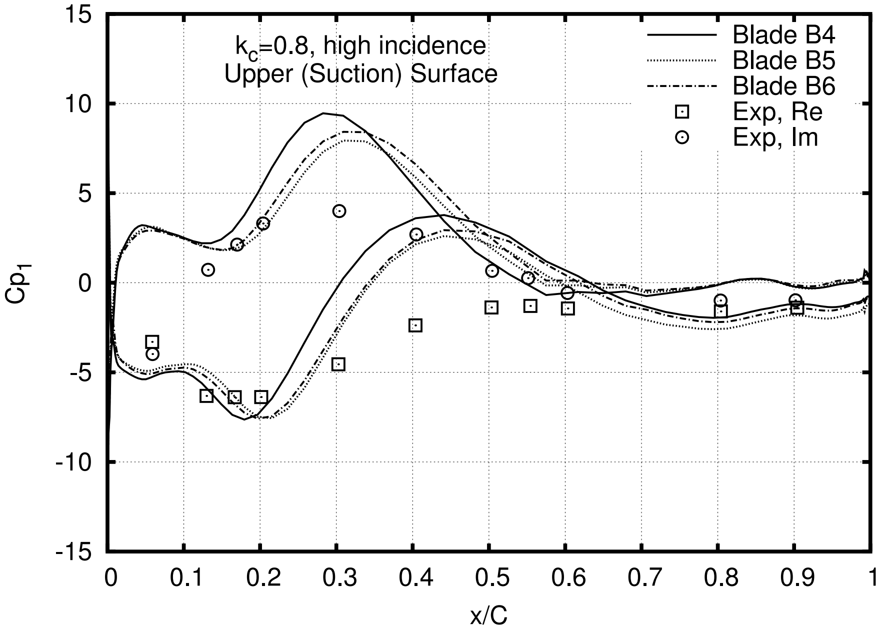


Figure 8.42: Upper surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 0.8$

trend of the unsteady pressure coefficient is predicted very well for the 3 center blades on both real part and imaginary part. The real part and the imaginary part compare very well with the experiment on the aft part of the blades. On the front part, the imaginary part is under predicted, the real part is over predicted. In Fig. 8.41, the trend of the local stability is predicted very well on the front part of the blade. The magnitude of the stability is predicted higher than the experiment measurement. The stable region in the numerical analysis starts from $x/C$ =0.05, earlier than $x/C = 0.15$ in the experiment measurement. The LE instability is well predicted. On the aft part, the numerical simulation predicts a little instability from $x/C = 0.5$ to $x/C = 0.85$, which is different from the experiment. At TE region, both the numerical results and the experiment measurement show a stable region from $x/C$ =0.85 to $x/C = 1$. The local stability analysis results are very close among the 3 center blades.

In the case of $k_c = 0.8$, the numerical analysis is plotted in Figs. 8.42 through 8.44 . As shown in Fig. 8.42, the upper surface pressure coefficients are predicted better than the case

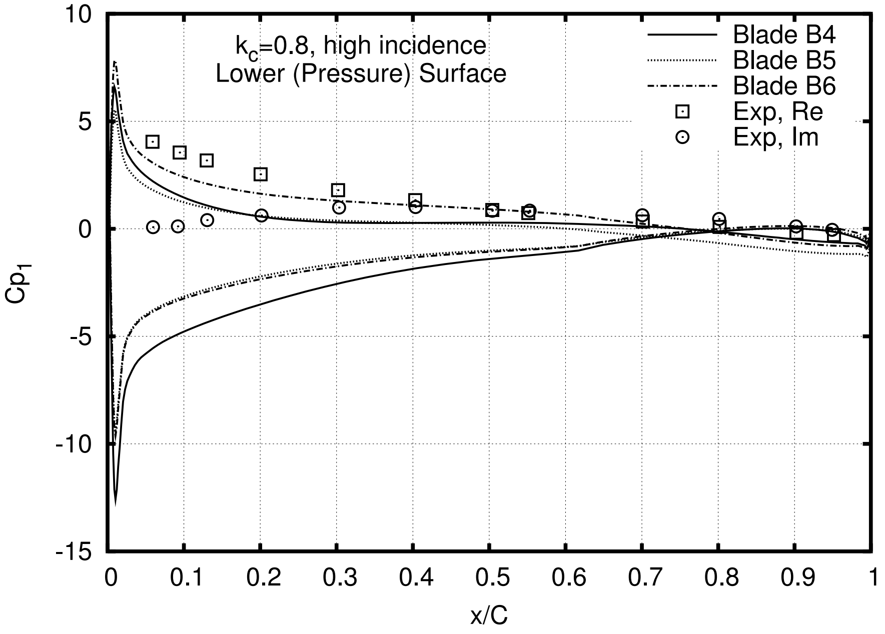


Figure 8.43: Lower surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 0.8$

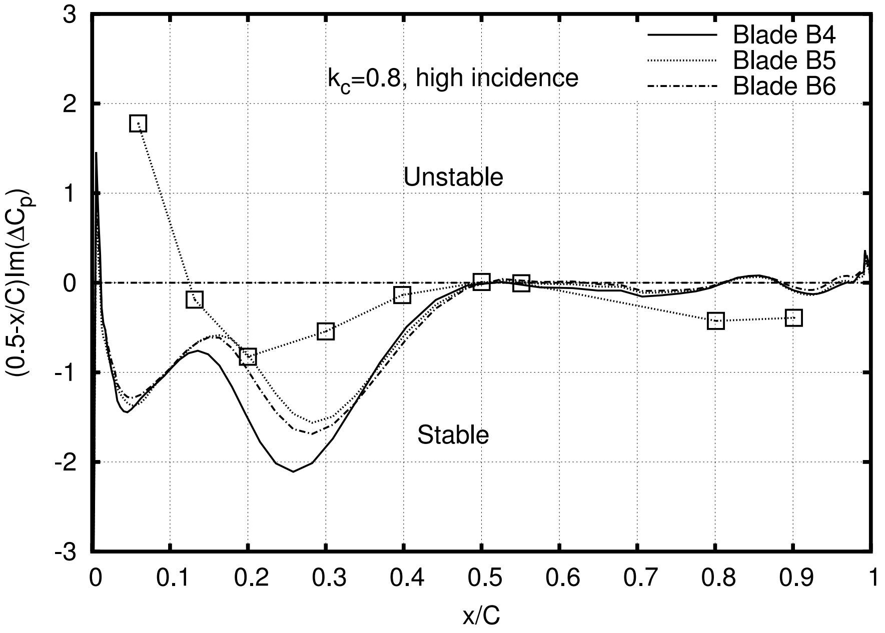


Figure 8.44: Local stability analysis in full scale cascade at high incidence, $k_c = 0.8$

of $k_c = 0.4$. The numerical results compare well with the experiment in trend and in the magnitude in LE region and the aft part of the blade, but are over predicted in the mid-chord part. The trend of the predicted lower surface unsteady pressure coefficient compare well with the experiment measurement but the magnitudes are under predicted for both the real part and the imaginary part of the front part of the blade (Fig. 8.43). The numerical results for the 3 center blades do not compare with each other as well as that in the low frequency case of $k_c = 0.4$. The local stability analysis shows better agreement with the experiment measurement in the case of $k_c = 0.8$ than the low frequency case ($k_c = 0.4$) in Fig. 8.44. The numerical computation shows similar stable region after the mid-chord region even though the size of the stable region and the magnitude of the local stability are smaller than experiment measurement. On the front part of the blade, the stability is over predicted in magnitude, but the trend compares well with the experiment. Similar to the analysis in $k_c = 0.4$, the size of the stable region is predicted larger than the experiment. The LE instability is also predicted. However, the difference of the local stability magnitude becomes more obvious among the 3 center blades, especially in the front part from $x/C = 0.15$ to $x/C = 0.35$. On the aft part, the local stability is close for the 3 center blades.

The results of the high frequency case $k_c = 1.2$ are plotted in Figs. 8.45 through 8.47 . The discrepancy in the results across the blade increases with the increased frequency. As shown in Fig. 8.45, the unsteady pressure coefficient differs obviously among the 3 blades on the front part of the blade upper surface. The best agreement with the experiment is achieved on blade B6, except the imaginary part is under predicted on the front part and the real part is over predicted in the LE region. On the lower surface, as shown in Fig. 8.46, the trend of the unsteady pressure coefficient agrees with the experiment measurement, but the the magnitude is under predicted for all the 3 blades. Similar to the results on upper surface, the best agreement with the experiment is obtained on blade B6. The discrepancy exists also in the local stability analysis in Fig. 8.47. The overall agreement with the experiment is improved compared with the low frequencies, $k_c = 0.4$ and $k_c = 0.8$. On the aft part, the numerical simulation predict stability with similar magnitude as the experiment

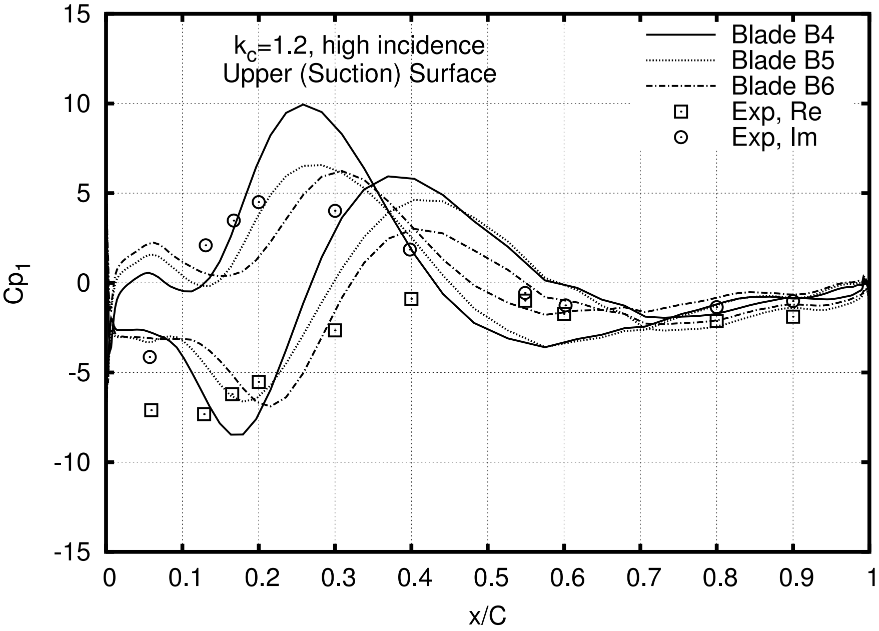


Figure 8.45: Upper surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 1.2$

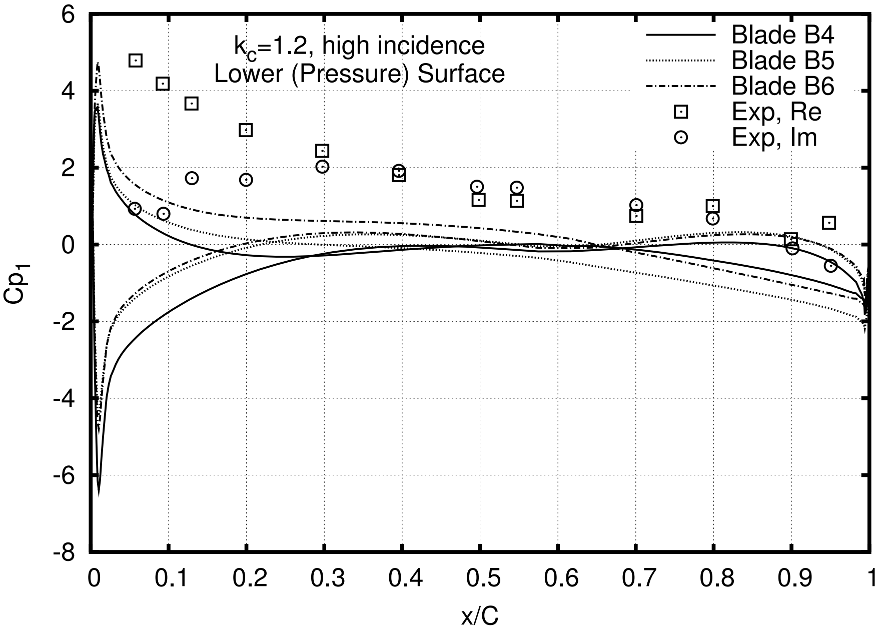


Figure 8.46: Lower surface unsteady pressure coefficient in full scale cascade at high incidence, $k_c = 1.2$

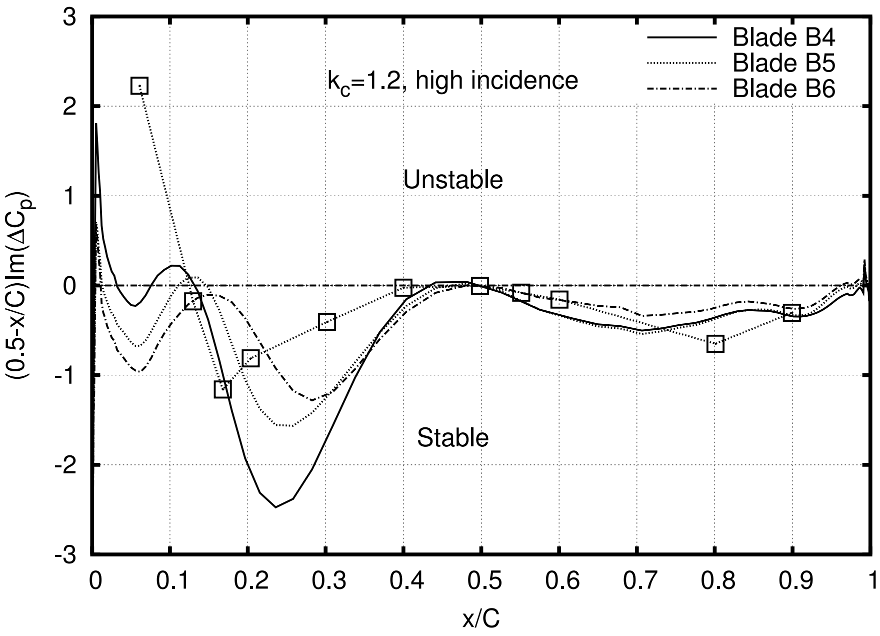


Figure 8.47: Local stability analysis in full scale cascade at high incidence, $k_c = 1.2$

measurement. The best agreement is shown on blade B6, which show smaller stability magnitude than the other two blades. On the front part, even though the numerical results varies significantly among the 3 center blades, the trend of the numerical prediction compares well with the experiment measurement on the 3 blades. The numerical computation shows stability in the same region as the experiment from $x/C = 0.15$ to $x/C = 0.5$. The highest local stability is found at $x/C = 0.24$, 0.25 and 0.28 on blade B4, B5 and B6 respectively compared with maximum location in the experiment at $x/C = 0.17$. Similar to the experiment, the local stability decreases from its maximum location to the leading edge and approaches instability at $x/C = 0.13$ for blade B4 and $x/C = 0.15$ for blade B5. Blade B6 stays stable in the same region and reaches its lowest stability at $x/C = 0.15$. Even though the leading instability is obtained in the numerical simulation on the 3 center blades, the blades are all shown stable in the region around $x/C = 0.05$, which does not exist in the experiment measurement. The size and the maximum magnitude of the leading edge stable region is smallest on blade B4.

The overall stability variation with the increased frequency is shown in Fig. 8.48 . Just

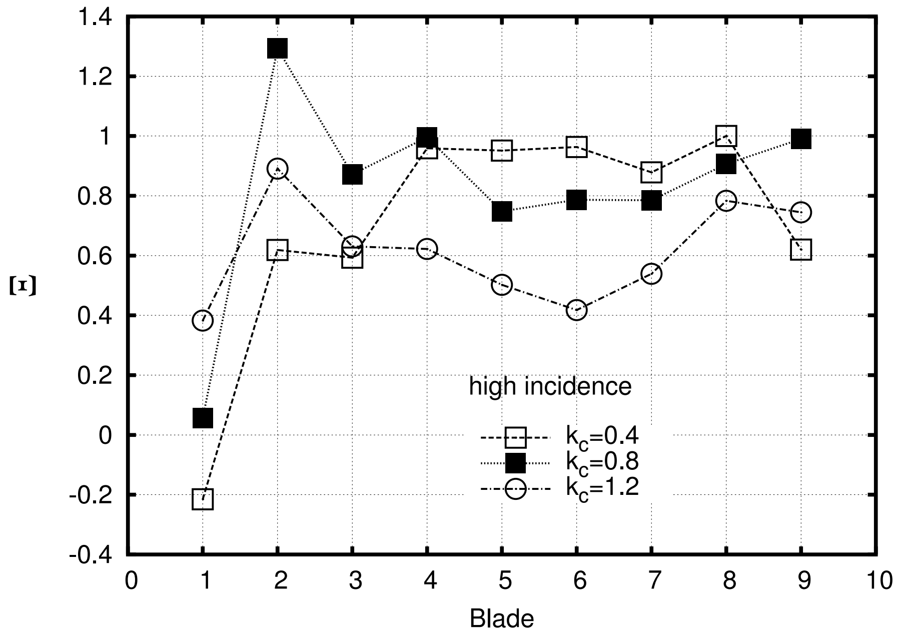


Figure 8.48: Damping coefficient in full scale cascade at high incidence

like the low incidence cases, the least stable blade under the 3 frequencies is blade B1. However, blade B9 is not the most stable blade for all the 3 frequencies. The most stable blade is B8 when $k_c = 0.4$ and B2 when $k_c = 0.8$ and 1.2. The damping coefficient distributions across the center blades are not as uniform as that in the low incidence cases. The best uniform distribution is obtained on blade B4, B5 and B6 under $k_c = 0.4$.. The overall stability indicated by the magnitude of the damping coefficient decreases with the increased frequency. The similar result is reported in [63].

# Chapter 9

# Conclusions

A fully implicit three-dimensional time accurate solver is developed to solve the non-linear Favre-averaged Navier-Stokes equations with the Baldwin-Lomax turbulence model. The governing equations are discretized with the finite volume method. The inviscid flux across the control volume surface is calculated with upwind schemes. A new low diffusion E-CUSP scheme, Zha CUSP scheme, which is consistent with the characteristic direction of the disturbance propagation, is developed to improve the efficiency and accuracy of the inviscid flux computation. The inviscid flux differencing achieves the third order accuracy by the MUSCL differencing approach. The viscous flux achieves the second order with central differencing. The second order time marching is achieved by the dual-time stepping scheme. The Gauss-Seidel line iteration is applied on the inner pseudo time step to implicitly solve the linear governing equations. The convergence is accelerated using the technique of local time stepping. The solver is made parallel using the MPI protocol.

The solver is first validated for its turbulence model, parallel computation speedup scalability and unsteady calculation capability. The Zha CUSP scheme and it modified version are compared with other popular upwind schemes in 2D and 3D test cases. The solver is then used to study the steady and unsteady characteristics of a 3D cascade separation flow. Finally the aeroelastic performance of a flutter cascade with a constant IBPA is studied with parallel computation.

In this dissertation, the Zha CUSP schemes are the first time to be applied in moving grid systems and 2D and 3D calculations. The implicit Gauss-Seidel iteration with dual time stepping is the first time to be used for moving grid systems. The NASA flutter cascade is the first time to be calculated in full scale.

## 9.1 Validation Calculation

The Baldwin-Lomax turbulence model in the Navier-Stokes solver works very well in the viscous boundary layer and the shock wave-boundary layer interaction. The velocity distribution in the subsonic flat plate boundary layer flow case compares very well with the law of the wall. The shock wave in the transonic inlet-diffuser is clearly captured and the pressure distribution on the wall agrees well with the experiment measurement. In the flat plate boundary layer flow parallel computation, a very good parallel computation speedup scalability is achieved by distributing the computation jobs evenly on multiple processors. The solver performs very well in unsteady simulation. In the unsteady transonic inlet-diffuser case, the frequency of the self-excited shock wave oscillation is fairly well predicted. In the forced oscillating airfoil calculation, the predicted histories of the unsteady coefficients of moment and lift compare well with the experiment measurement.

## 9.2 Zha CUSP Schemes

The performance of the Zha CUSP schemes is compared with the Roe scheme, AUSM$^+$ scheme, van Leer scheme, and van Leer-Hänel scheme.

The original Zha CUSP scheme is tested in 2D cases. In the supersonic adiabatic laminar flat plate boundary layer, the Zha CUSP scheme accurately resolve the boundary layer velocity and temperature profiles using the first order differencing. The solution is as accurate as that of the Roe scheme and the AUSM$^+$ scheme and hence demonstrates the low diffusion of the new scheme. In the transonic converging-diverging nozzle, oblique shock waves and reflections are crisply captured even though the shock waves do not align with the mesh lines. The predicted wall surface isentropic Mach number distribution agrees well with the experiment. In the transonic inlet-diffuser with shock wave-turbulent boundary layer interaction, the Zha CUSP scheme and the Roe scheme predict the surface pressure distributions agreeing well with the experiment for the case of a weak shock. For the strong shock case, both the Zha CUSP scheme and the Roe scheme over predict the strength of the shock wave. However, the pressure distribution predicted by the Zha CUSP scheme is closer to the experiment. The AUSM$^+$ solution has large pressure oscillations.

The Zha CUSP scheme is modified to the Zha CUSP2 scheme and applied in 3D cases. For the transonic nozzle with circular-to-rectangular cross section and the subsonic compressor cascade, the wall static pressure distributions computed by the Zha CUSP2 scheme are in good agreement with the experiments. The CPU time to calculate the flux using the Zha CUSP2 scheme is about 40% less than that used by the Roe scheme. For the transonic channel case, the shock wave structure given by the Zha CUSP2 scheme agrees well with the experiment. The result of the Zha CUSP2 scheme agrees better with the experiment than the one predicted by the Roe scheme, which gives flow separation that does not exist in the experiment. The Zha CUSP2 scheme also predicts the peak Mach number closer to the experiment than that of the Roe scheme.

The Zha CUSP schemes are shown to be accurate, efficient and robust for 2D and 3D flows.

## 9.3 Stationary Cascade Separation Computation

The steady and unsteady characteristics of the separation phenomenon in a transonic flutter cascade under high incidence is numerically studied in 3D with a incidence angle of 10° and 3 inlet Mach numbers, 0.5, 0.8 and 1.18. The van-Leer scheme is applied.

In steady state results, high incidence triggers the flow separation on the blade suction surface. At subsonic inlet Mach number 0.5 and 0.8, a large separation region is formed immediately from the leading edge when the incidence is increased from 0° to 10°. The inlet Mach number determines the characteristics of the separation region. When the inlet flow is subsonic, the size of the separation region increases with the inlet Mach number. This is evidenced as the separation region of Mach 0.8 is greater than that of Mach 0.5. The predicted separation flow pattern and the separation bubble length agree well with the experiment. The overall predicted surface pressure distribution agrees reasonably well with the experiment, except in the suction surface leading edge region where the computation predicts the pressure rise more steeply than that of the experiment. When the inlet flow becomes supersonic, shock waves appear in the flow path close to the cascade leading edge. The shock wave-turbulent boundary layer interaction causes the flow separation, which is pushed more downstream with smaller size compared with those in the subsonic cases.

In the unsteady simulation results, the high incidence cascade separation flow shows a sinusoidal pattern on the oscillation of the surface pressure and the separation bubble size. A frequency spectrum peak is obtained at 770Hz for the case of $M$=0.5 and 1400Hz for the case of $M$=0.8. The leading edge vortex shedding is the mechanism behind the unsteady characteristics of the subsonic high incidence separation flow. New vortexes are continuously generated at the suction surface leading edge. The new vortex grows and pushes the old vortexes downstream. The interaction between the vortexes results in the periodical oscillation of the separation bubble size and the surface pressure. The vortex generation, pressure variation and separation length oscillation have the same frequency characteristics with a phase difference. The characteristics of the separation flow is determined by the inlet Mach number. When the inlet flow goes from lower subsonic to higher subsonic, the size and the oscillation intensity of the separation bubble are enhanced. The flowfield oscillation peak frequency increases. When the inflow goes further to supersonic, the flow is attached on the leading edge. A small size separation bubble due to the interaction of the shock wave and the turbulent boundary layer is located right after the shock wave.

## 9.4 Oscillating Cascade Unsteady Simulation

The oscillating cascade is calculated with parallel computation in 2D with 2 incidences, 0° and 10° and an inlet Mach number 0.5 under 3 oscillation frequencies, 0.4, 0.8 and 1.2 and a IBPA of 180°.

The unsteady results are more sensitive to mesh configuration. The computed flowfield is affected by the existence of the end walls. The steady state blade surface pressure varies with the blade position in the cascade. The periodicity of the flow pattern is improved by aligning the end walls to flow direction. The end wall influence attenuates quickly from boundary passages to center passages. Good periodicity is achieved in the inner passages.

All blades in the full scale cascade are stable, which is indicated by a positive damping coefficient. The damping coefficient is more uniformly distributed on center blades. The damping coefficient increases with the increased frequency in low incidence cases, but decreases in high frequency cases. The spatial periodicity deteriorates when the incidence increases. The large separation on the suction surface brings higher unsteadiness to the flow. Its interaction with the blade surface and the end walls affect the flow periodicity. The temporal periodicity is highly affected by the increased frequency in high incidence cases. The unsteady pressure coefficient and the local stability are predicted well in trend compared with the experiment. But the magnitudes differ from the experiment. The leading edge instability is captured in high incidence cases. When the frequency is increased, the local stability compares better with experiment.

# Chapter 10

# Future Work

The fully implicit 3D parallel Navier-Stokes solver developed in the current provides a powerful and reliable tool for the fluid-structural interaction research in turbomachinery. The new Zha CUSP schemes are proved robust, efficient and accurate in the numerical simulation. The following issues are recommended by the author to further improve the performance of the Navier-Stokes solver for the future work.

## Turbulence models

The Baldwin-Lomax model is applied for its simplicity. However its poor performance in separation flows affect the solver's capability to handle more complicated flows in turbomachinery. The one-equation Baldwin-Barth [57] model, Spalart-Allmaras [58] model and the Wilcox's $\kappa - \omega$ model [59] have demonstrated good performance in turbomachinery study. They are good options to be added into the solver.

## Multi-grid and residual smoothing

The multi-grid and the residual smoothing are another two popular and effective convergence acceleration methods used in turbomachinery. The current solver can be made more efficient if they are adopted, which will improve the solver's ability in handing more complicated 3D turbomachinery problems.

## 3D fluid-structural interaction simulation

The unsteady full scale multi-passage cascade is calculated using parallel computation in 2D only in the current work to save computing time. This is reasonable for the special flutter cascade studied, which is designed intentionally for 2D flows. However, the flows in turbomachinery have very strong 3D characteristics. The full annular stages in turbomachinery is even different from the linear cascade. A full 3D calculation is required to carry out more realistic turbomachinery research. This is feasible when more time saving is achieved by more effective convergence acceleration techniques, the effective multi-grid and residual smoothing, are adopted in the solver.

## Fluid-structural interaction with self-excited oscillating blades

The self-excited shock wave oscillation in stationary inlet-diffuser and the fluid-structural interaction around the forced oscillating blades are calculated in the current work. However, the fluid-structural interaction in reality is more complicated. To be more general, the blade motion should be resulted from the fluid-structural interaction, not given in advance. The current studies are more focused on the one-way fluid response to the oscillating structure. The fluid-structural interaction with the flow induced vibrating blades should be added to simulate more general turbomachinery problems.

## Multi-block computation domain

Efforts have been made in the current solver to handle the complicated geometries. The full scale cascade in the linear cascade is able to be split into individual passages and computed by multiple processors simultaneously. The mesh orthogonality requirement of the Baldwin-Lomax model is assured by the additional boundary layer mesh attached on the blade surface region. However, the geometries in reality may be much more complicated. Applying the multi-block topology in the computation domain can improve the solver's adaptability for turbomachinery research. A series of sub-blocks can be devised to satisfy the special requirements at different locations in the computation domain.

# Appendix A

# MPI Implementation

The MPI protocol provides a widely used standard for writing MPI programs. The MPI specifications have been defined in C and Fortran routines. A parallel program use the MPI protocol by calling the MPI routines in the code. The LAM/MPI version 6.5.9 is obtained free from `http://www.lam-mpi.org/` and is adopted in the current code.

## A.1 MPI Basics

MPI uses objects called communicators and groups to define which collection of processes may communicate with each other. The data is exchanged between processes within the same group associated with the same communicator. Each member process is indexed with a unique integer rank, which is used to specify the source and target process in data exchange. In the current code, the `MPI_Comm_world` is the only communicator used for all MPI processes.

A standard MPI programming flowchart is shown in Fig. A.1 . The MPI environment is first initialized, then an optional construction of the derived data type is conducted. Once the environment is set up, the MPI communication routines are used to carried out the message exchanges between neighboring subdomains during the integration. After all computation is done, the MPI environment is finalized.

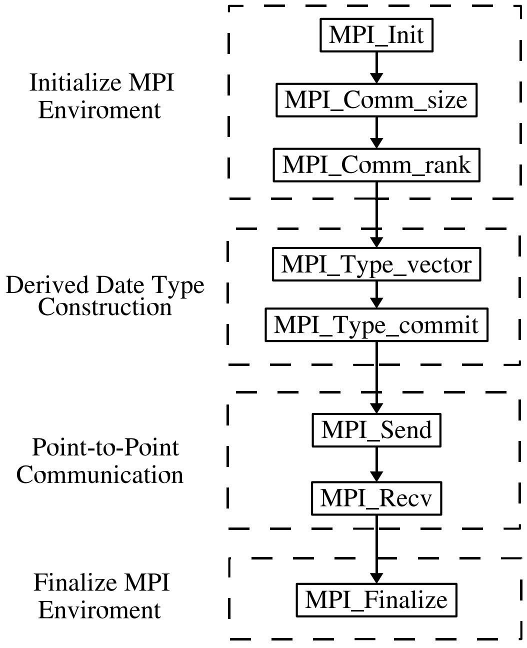


Figure A.1: MPI programming standard flowchart

The routines used for the initialization and finalization of MPI environment are the MPI environment management routines. `MPI_init` is used to initialize the MPI execution environment. It must be called and be called only once in every MPI program and must be called before any other MPI functions. `MPI_Comm_size` is used to determine the number of processes (*np*) in the communication group associated with the communicator. `MPI_Comm_rank` is used to determine the rank or the index of the calling process in the communicator. After calling this routine, each process will be assigned a unique integer rank between 0 and $np-1$. `MPI_Finalize` is used when all MPI communication is done to terminate the MPI execution environment.

The construction of derived data type is optional. The fundamental MPI communication is carried out based on the primitive data types, for example `MPI_integer`, `MPI_double_precision`. When the data to be exchanged are located non-contiguously in memory, each element will have to be exchanged individually between two neighboring processes, or the data is first copied to an contiguous memory block and then the new memory block is exchanged as a unit. A better way to handle the non-contiguous data is to construct an appropriate derived data type to include all the data elements to be exchanged at one time and transfer them as a single block using one communication call. This will simplify the programming and achieve higher efficiency in inter-process data transferring. The derived data type used in the current code are constructed using the following MPI routines. MPI_Type_contiguous makes contiguous copies of a primitive data type. MPI_Type_vector and MPI_Type_hvector are similar to MPI_Type_contiguous, but allows regular gaps (strides) in the displacement. MPI_Type_vector and MPI_Type_hvector specify the gap in the unit of the base data type and in the unit of bytes respectively. MPI_Type_extent returns the size in bytes of the specified data type. All derived data types are constructed only once and should be registered to the system using MPI_Commit before they are used by other MPI routines. After the registration, a unique handle will be assigned to each derived data type.

MPI point-to-point communication operations typically involve message passing between two, and only two processes. In a communication pair, one task is performing a send operation and the other task is performing a matching receive operation. The communication can be carried out in two one-way routines (MPI_Send and MPI_Recv) or in a two-way routine MPI_Sendrecv. The one-way routine sends or receives a message and block until the requested data is available in the application buffer in the sending or receiving process. The two-way routine sends a message and posts a receive before blocking. It blocks until the sending application buffer is free for reuse and until the receiving application buffer contains the received message. These communications repeat in every computation iteration.

More detailed descriptions about the MPI routines are available from the documentation at http://www.lam-mpi.org/ or from reference [90].

## A.2 MPI Programming Example

In the current code, the MPI data exchanging is carried out for the flow variables $\rho$, $\rho u$, $\rho v$, $\rho w$ and $\rho e$ (2 layers), turbulent viscosity $\mu_t$ (1 layer), computation residual and its location $(i, j, k)$ . The maximum residuals on all subdomains are transfered to the process with rank 0 to determine the maximum residual for the whole domain. When the residual criteria is reached, a stop signal is sent from process rank 0 to all other processes to terminate their computations. If a uniform time step is used, the minimum time steps $\Delta t$ on all sub-domains are also exchanged to determine the the time step appropriate for the whole domain. If the moving grid system is enabled, the boundary coordinates $x$, $y$ and $z$ (1 layer) and the grid moving velocity ($\xi_t$,$\eta_t$, $\zeta_t$, 1 layer) are also exchanged every time the boundary is updated. The communications for residual and time step are relatively simple. For all other variables, an array of data on the boundary surface (3D) or line (2D) are to be exchanged every time between neighboring processes. These arrays of data are not contiguously located in memory. More complicated derived data types are constructed for each variable to increase the data transferring efficiency. The turbulent viscosity $\mu_t$ is chosen for description in the following MPI programming example.

The current code is programmed in Fortran 90. In each MPI program, the MPI library header file is required to be included at the beginning as,

```
include 'loading/path/mpif.h'
```

The `loading path` is installation dependent. It is "`/usr/local/include/`" for the current code.

### MPI environment initialization

The MPI environment initialization is carried out once at the beginning of the program. The following example are excerpted from the current code.

```
call mpi_init(ierr)
```

```
    call mpi_comm_size(mpi_comm_world, np, ierr)
    call mpi_comm_rank(mpi_comm_world, rank, ierr)
```

Here, ierr is the integer to indicate the return status of the MPI routine, the communicator is the universal mpi_comm_world, np indicates the total process number involved in the current parallel computation, rank is the system assigned rank or identifier for the calling process.

## Derived data type construction

At first, the exact length in bytes of the data type mpi_double_precision is assigned to an integer variable sizeofdouble for later use.

```
    call mpi_type_extent(mpi_double_precision, sizeof-
    double,
           ierr)
```

The 3D array defined for $\mu_t$ is visturb(0:il+1,0:jl+1,0:kl+1).il, jl, kl are the subdomain cell numbers in $\xi$, $\eta$, $\zeta$ direction. The extra phantom cells as deep as 1 layer are added to the subdomain on each direction. The data is contiguous following the directory of $\xi$ (0→il+1), then $\eta$ (0→jl+1) and $\zeta$ (0→kl+1). The data elements for exchange with the neighboring process is a 2D array. For example, on the right side boundary, the data array visturb(il,1:jl,1:kl) will be sent to the right side neighboring process to update its left side phantom boundary layer variables in visturb(0,1:jl,1:kl). The data in these 2D arrays are not contiguous in memory. A turbxi data type is created for this type of data exchange as the following.

```
    call mpi_type_vector(jl, 1, irange,
           mpi_double_precision, line, ierr)
    call mpi_type_hvector(kl, 1,
           irange*jrange*sizeofdouble, line, tur-
    bxi, ierr)
    call mpi_type_commit(turbxi, ierr)
```

The first routine call constructs a temporary line data type to include a line of data of type mpi_double_precision in $\eta$ direction (1→jl). There are jl blocks in a line. In each block, there is 1 base data type mpi_double_precision. The displacement between two blocks is irange (number of base data types). Here irange is equal to il+2, which refers to 0:il+1. The second routine call constructs the turbxi data type based on the temporary line data type. Similar to the first routine call. There are kl blocks in a turbxi. In each block, there is 1 base data type line. The displacement between two blocks is irange*jrange*sizeofdouble (in unit of bytes). jrange is equal to jl+2, which refers to 0:jl+1. The new turbxi data type is registered to the system in the third routine call.

There are other two data types, turbeta and turbzeta constructed similarly for turbulent viscosity $\mu_t$ communication in $\eta$ and $\zeta$ directions respectively.

## Point-to-point communication

During computation iterations, the turbulent viscosities are exchanged between processes every time when their values are updated locally. Following the previous turbxi data type example, the turbulent viscosity exchange is carried in the code as the following.

```
       call mpi_sendrecv(visturb(il,1,1), 1, turbxi, ie,
              tag_turb, visturb(il+1,1,1), 1, turbxi, ie,
              tag_turb,mpi_comm_world, status, ierr)
```

Here ie is rank of its right neighbor process. tag_turb is an integer tag, which uniquely marks the current message when combined with the source or target process rank. This routine call exchanges visturb between its right surface and the left surface of its right neighboring process. Starting from the memory location of visturb(il,1,1), a data type turbxi is sent to process ie. In the mean time, another phantom layer data type starting from visturb(il+1,1,1) is receiving data from process ie. The routine call returns its status to an integer variable status. The similar routine calls are made on the other 5 surface of the current sub-domain to finish all necessary boundary data exchange.

### MPI environment finalization

When the computation iteration is finished, the MPI execution environment is finally terminated by the following MPI routine call.

```
call mpi_finalize(ierr)
```

# Appendix B

# $k-\omega$ Turbulence Model Implementation

Even though the two-equation $k-\omega$ turbulence model of Wilcox [59] is not used in the current work for turbulence modeling, efforts have been made to add this function as an option to the solver for future application.

This two-equation model includes one equation for the turbulent kinetic energy $k$ and a second equation for the specific turbulent dissipation rate (or turbulent frequency) $\omega$. Different from the zero-equation Baldwin-Lomax model, the turbulent viscosity is determined by the turbulent kinetic energy $k$ and the turbulent dissipation rate $\omega$. This removes the implementation difficulties of the Baldwin-Lomax model in 3D and complex geometry conditions.

The two equations for $k$ and $\omega$ are expressed in the Favre-averaged form as,

$$\frac{\partial(\bar{\rho}\tilde{k})}{\partial t}+\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{k}\right)=\frac{\partial}{\partial x_j}\left[(\mu+\sigma^*\mu_t)\frac{\partial\tilde{k}}{\partial x_j}\right]+P-\beta^*\bar{\rho}\tilde{\omega}\tilde{k} \tag{B.1}$$

$$\frac{\partial(\bar{\rho}\tilde{\omega})}{\partial t}+\frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\omega}\right)=\frac{\partial}{\partial x_j}\left[(\mu+\sigma\mu_t)\frac{\partial\tilde{\omega}}{\partial x_j}\right]+\alpha\frac{\tilde{\omega}}{\tilde{k}}P-\beta\bar{\rho}\tilde{\omega}^2 \tag{B.2}$$

where the generation of turbulent energy $P$ is,

$$P=\left[2\mu_t\left(S_{ij}-\frac{1}{3}\frac{\partial\tilde{u}_k}{\partial x_k}\delta_{ij}\right)-\frac{2}{3}\rho k\delta_{ij}\right]\frac{\partial\tilde{u}_i}{\partial x_j} \tag{B.3}$$

The constants used for the two equations are,

$$\alpha = \frac{5}{9}, \beta = \frac{3}{40}, \beta^* = \frac{9}{100}, \sigma = \frac{1}{2}, \sigma^* = \frac{1}{2}$$

The turbulent viscosity is therefore calculated as,

$$\mu_t = \rho \frac{k}{\omega} \tag{B.4}$$

The non-dimensional $k^*$ and $\omega^*$ are defined as,

$$k^* = \frac{k}{U_\infty^2} \tag{B.5}$$

$$\omega^* = \frac{\omega}{U_\infty / L} \tag{B.6}$$

The normalized equations for $k^*$ and $\omega^*$ are,

$$\frac{\partial(\rho^* k^*)}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho u_j^* k^*\right) = \frac{1}{\mathrm{Re}_L}\frac{\partial}{\partial x_j^*}\left[(\mu^* + \mathrm{Re}_L \sigma^* \mu_t^*)\frac{\partial k^*}{\partial x_j^*}\right] + P_k - \beta^* \rho^* \omega^* k^* \tag{B.7}$$

$$\frac{\partial(\rho^* \omega^*)}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho u_j^* \omega^*\right) = \frac{1}{\mathrm{Re}_L}\frac{\partial}{\partial x_j^*}\left[(\mu^* + \mathrm{Re}_L \sigma \mu_t^*)\frac{\partial \omega^*}{\partial x_j^*}\right] + \alpha \frac{\omega^*}{k^*} P_k - \beta \rho^* \omega^{*2} \tag{B.8}$$

Eqs. (B.7) and (B.8) are then added to the governing equations (2.34), (2.35) and (2.36). The total stress in the momentum equation (2.35) and energy equation (2.36) is updated to include the additional turbulent kinetic energy $k$,

$$\tau_{ik}^* = (\mu^* + \mu_t^* \mathrm{Re}_L)\left[\left(\frac{\partial u_i^*}{\partial x_k^*} + \frac{\partial u_k^*}{\partial x_i^*}\right) - \frac{2}{3}\delta_{ik}\frac{\partial u_j^*}{\partial x_j^*}\right] - \frac{2}{3}\mathrm{Re}_L \rho^* k^* \delta_{ij} \tag{B.9}$$

The governing equations with the $k - \omega$ turbulence model are finally written in an integral vector form similar to Eq. (2.69). Drop the superscript asterisk for clarity. The conservative variable vector **U**, inviscid flux **E**, **F**, **G**, viscous flux **R**, **S**, **T** and the source term **D** are expressed as the following,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \\ \rho k \\ \rho \omega \end{bmatrix} \tag{B.10}$$

$$\mathbf{E} = \begin{bmatrix} \rho U \\ \rho u U + l_x p \\ \rho v U + l_y p \\ \rho w U + l_z p \\ (\rho e + p) U \\ \rho k U \\ \rho \omega U \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho V \\ \rho u V + m_x p \\ \rho v V + m_y p \\ \rho w V + m_z p \\ (\rho e + p) V \\ \rho k V \\ \rho \omega V \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho W \\ \rho u W + n_x p \\ \rho v W + n_y p \\ \rho w W + n_z p \\ (\rho e + p) W \\ \rho k W \\ \rho \omega W \end{bmatrix} \tag{B.11}$$

$$\mathbf{R} = \begin{bmatrix} 0 \\ l_k \tau_{xk} \\ l_k \tau_{yk} \\ l_k \tau_{zk} \\ l_k \beta_k \\ l_k k_k \\ l_k \omega_k \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ m_k \tau_{xk} \\ m_k \tau_{yk} \\ m_k \tau_{zk} \\ m_k \beta_k \\ m_k k_k \\ m_k \omega_k \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 0 \\ n_k \tau_{xk} \\ n_k \tau_{yk} \\ n_k \tau_{zk} \\ n_k \beta_k \\ n_k k_k \\ n_k \omega_k \end{bmatrix} \tag{B.12}$$

$$\mathbf{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ P - \beta^* \rho \omega k \\ \alpha \frac{\omega}{k} P - \beta \rho \omega^2 \end{bmatrix} \tag{B.13}$$

The stress components are expressed as the following.

$$\tau_{xx} = \frac{4}{3}\alpha_\tau \left( u_\xi l_x + u_\eta m_x + u_\zeta n_x \right) - \frac{2}{3}\rho k \\ - \frac{2}{3}\alpha_\tau \left( v_\xi l_y + v_\eta m_y + v_\zeta n_y + w_\xi l_z + w_\eta m_z + w_\zeta n_z \right) \tag{B.14}$$

$$\tau_{yy} = \frac{4}{3}\alpha_\tau \left( v_\xi l_y + v_\eta m_y + v_\zeta n_y \right) - \frac{2}{3}\rho k \\ - \frac{2}{3}\alpha_\tau \left( u_\xi l_x + u_\eta m_x + u_\zeta n_x + w_\xi l_z + w_\eta m_z + w_\zeta n_z \right) \tag{B.15}$$

$$\tau_{zz} = \frac{4}{3}\alpha_\tau \left( w_\xi l_z + w_\eta m_z + w_\zeta n_z \right) - \frac{2}{3}\rho k \\ - \frac{2}{3}\alpha_\tau \left( u_\xi l_x + u_\eta m_x + u_\zeta n_x + v_\xi l_y + v_\eta m_y + v_\zeta n_y \right) \tag{B.16}$$

$$\tau_{xy} = \alpha_\tau \left[ u_\xi l_y + u_\eta m_y + u_\zeta z_y + v_\xi l_x + v_\eta m_x + v_\zeta z_x \right] \tag{B.17}$$

$$\tau_{xz} = \alpha_\tau \left[ u_\xi l_z + u_\eta m_z + u_\zeta z_z + w_\xi l_x + w_\eta m_x + w_\zeta z_x \right] \tag{B.18}$$

$$\tau_{yz} = \alpha_\tau \left[ v_\xi l_z + v_\eta m_z + v_\zeta z_z + w_\xi l_y + w_\eta m_y + w_\zeta z_y \right] \tag{B.19}$$

where

$$\alpha_\tau = \frac{J}{\mathrm{Re}_L} \left( \mu + \mu_t \mathrm{Re}_L \right) = \frac{1}{\mathrm{Re}_L \Delta V} \left( \mu + \mu_t \mathrm{Re}_L \right) \tag{B.20}$$

The coefficients $\beta_x$, $\beta_y$ and $\beta_z$ are expressed as,

$$\beta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \tag{B.21}$$

$$\beta_y = u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y \tag{B.22}$$

$$\beta_z = u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \tag{B.23}$$

$$q_x = -\alpha_q \left( T_\xi l_x + T_\eta m_x + T_\zeta n_x \right) \tag{B.24}$$

$$q_y = -\alpha_q \left( T_\xi l_y + T_\eta m_y + T_\zeta n_y \right) \tag{B.25}$$

$$q_z = -\alpha_q \left( T_\xi l_z + T_\eta m_z + T_\zeta n_z \right) \tag{B.26}$$

where

$$\alpha_q = \frac{1}{(\gamma - 1) M_\infty^2} \frac{J}{\mathrm{Re}_L} \left( \frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t} \mathrm{Re}_L \right) = \frac{1}{(\gamma - 1) M_\infty^2} \frac{1}{\Delta V \, \mathrm{Re}_L} \left( \frac{\mu}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t} \mathrm{Re}_L \right) \tag{B.27}$$

The spatial derivatives of the turbulent kinetic energy $k$ are expressed as,

$$k_x = \alpha_k \left( k_\xi l_x + k_\eta m_x + k_\zeta n_x \right) \tag{B.28}$$

$$k_y = \alpha_k \left( k_\xi l_y + k_\eta m_y + k_\zeta n_y \right) \tag{B.29}$$

$$k_z = \alpha_k \left( k_\xi l_z + k_\eta m_z + k_\zeta n_z \right) \tag{B.30}$$

where

$$\alpha_k = \frac{J}{\mathrm{Re}_L} \left( \mu + \sigma^* \mu_t \mathrm{Re}_L \right) = \frac{1}{\Delta V \, \mathrm{Re}_L} \left( \mu + \sigma^* \mu_t \mathrm{Re}_L \right) \tag{B.31}$$

The spatial derivatives of the turbulent dissipation rate $\omega$ are expressed as,

$$\omega_x = \alpha_\omega \left( \omega_\xi l_x + \omega_\eta m_x + \omega_\zeta n_x \right) \tag{B.32}$$

$$\omega_y = \alpha_\omega \left( \omega_\xi l_y + \omega_\eta m_y + \omega_\zeta n_y \right) \tag{B.33}$$

$$\omega_z = \alpha_\omega \left( \omega_\xi l_z + \omega_\eta m_z + \omega_\zeta n_z \right) \tag{B.34}$$

where

$$\alpha_\omega = \frac{J}{\mathrm{Re}_L} \left( \mu + \sigma \mu_t \mathrm{Re}_L \right) = \frac{1}{\Delta V \, \mathrm{Re}_L} \left( \mu + \sigma \mu_t \mathrm{Re}_L \right) \tag{B.35}$$

The turbulent energy generation term $P$ is expressed as,

$$\begin{aligned} P = \tau_{ij} \frac{\partial u_i}{\partial x_j} = \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} + \tau_{xy} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ + \tau_{xz} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) + \tau_{yz} \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \end{aligned} \tag{B.36}$$

where

$$\begin{aligned}\frac{\partial u_i}{\partial x_j} &= J\left(\frac{\partial u_i}{\partial \xi}l_{x_j} + \frac{\partial u_i}{\partial \eta}m_{x_j} + \frac{\partial u_i}{\partial \zeta}n_{x_j}\right) \\ &= \frac{1}{\Delta V}\left(\frac{\partial u_i}{\partial \xi}l_{x_j} + \frac{\partial u_i}{\partial \eta}m_{x_j} + \frac{\partial u_i}{\partial \zeta}n_{x_j}\right)\end{aligned} \tag{B.37}$$

The characteristics of the governing equations are similar to those described in chapter 3. Take the inviscid flux $\mathbf{E}$ in $\xi$ direction as example, the Jacobian matrix is expressed as,

$$A = \frac{\partial \mathbf{E}}{\partial \mathbf{U}} =$$
$$\begin{pmatrix} 0 & l_x & l_y & l_z & 0 & 0 & 0 \\ -uU+(\gamma-1)l_xq & U+(2-\gamma)l_xu & l_yu-(\gamma-1)l_xv & l_zu-(\gamma-1)l_xw & (\gamma-1)l_x & 0 & 0 \\ -vU+(\gamma-1)l_yq & l_xv-(\gamma-1)l_yu & U+(2-\gamma)l_yv & l_zv-(\gamma-1)l_yw & (\gamma-1)l_y & 0 & 0 \\ -wU+(\gamma-1)l_zq & l_xw-(\gamma-1)l_zu & l_yw-(\gamma-1)l_zv & U+(2-\gamma)l_zw & (\gamma-1)l_z & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & \gamma U & 0 & 0 \\ -kU & l_xk & l_yk & l_zk & 0 & U & 0 \\ -\omega U & l_x\omega & l_y\omega & l_z\omega & 0 & 0 & U \end{pmatrix} \tag{B.38}$$

where $a_{51}$, $a_{52}$, $a_{53}$, $a_{54}$ and $q$ have same formulation as Eqs. (3.2) through (3.6).

The eigenvalues of matrix $A$ are,

$$\lambda_1 = U + C \tag{B.39}$$

$$\lambda_2 = U - C \tag{B.40}$$

$$\lambda_{3-7} = U \tag{B.41}$$

The eigenvalue matrix is,

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_7 \end{pmatrix} = \begin{pmatrix} U+C & & & & & & \\ & U-C & & & & 0 & \\ & & U & & & & \\ & & & U & & & \\ & & & & U & & \\ & 0 & & & & U & \\ & & & & & & U \end{pmatrix} \tag{B.42}$$

The left eigenvector matrix is,

$$L = \begin{pmatrix} q-\dfrac{c\hat{U}}{\gamma-1} & -u+\dfrac{c\hat{l}_x}{\gamma-1} & -v+\dfrac{c\hat{l}_y}{\gamma-1} & -w+\dfrac{c\hat{l}_z}{\gamma-1} & 1 & 0 & 0 \\ q+\dfrac{c\hat{U}}{\gamma-1} & -u-\dfrac{c\hat{l}_x}{\gamma-1} & -v-\dfrac{c\hat{l}_y}{\gamma-1} & -w-\dfrac{c\hat{l}_z}{\gamma-1} & 1 & 0 & 0 \\ -\hat{V} & \hat{m}_x & \hat{m}_y & \hat{m}_z & 0 & 0 & 0 \\ -\hat{W} & \hat{n}_x & \hat{n}_y & \hat{n}_z & 0 & 0 & 0 \\ q-h & -u & -v & -w & 1 & 0 & 0 \\ -k & 0 & 0 & 0 & 0 & 1 & 0 \\ -\omega & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.43}$$

The right eigenvector matrix is,

$$R = \begin{pmatrix} \frac{1}{2h} & \frac{1}{2h} & 0 & 0 & -\frac{1}{h} & 0 & 0 \\ \frac{u+c\hat{l}_x}{2h} & \frac{u-c\hat{l}_x}{2h} & \hat{m}_x & \hat{n}_x & -\frac{u}{h} & 0 & 0 \\ \frac{v+c\hat{l}_y}{2h} & \frac{v-c\hat{l}_y}{2h} & \hat{m}_y & \hat{n}_y & -\frac{v}{h} & 0 & 0 \\ \frac{w+c\hat{l}_z}{2h} & \frac{w-c\hat{l}_z}{2h} & \hat{m}_z & \hat{n}_z & -\frac{w}{h} & 0 & 0 \\ \frac{c\hat{U}+H}{2h} & \frac{-c\hat{U}+H}{2h} & \hat{V} & \hat{W} & -\frac{q}{h} & 0 & 0 \\ \frac{k}{2h} & \frac{k}{2h} & 0 & 0 & -\frac{k}{h} & 1 & 0 \\ \frac{\omega}{2h} & \frac{\omega}{2h} & 0 & 0 & -\frac{\omega}{h} & 0 & 1 \end{pmatrix} \tag{B.44}$$

The coefficients in the above equations are defined in Chapter 3.

# Appendix C

# RANS3D Solver

## C.1 Flowchart

The RANS3D solver flowchart is shown in Fig. C.1. The MPI implementation is also marked along the main program. The main program is made up with 2 layers of iterations. The outer layer iteration works on the physical time, and the total step number is "nstep". The inner iteration works on the pseudo time and is only needed by the dual time stepping method. The solver is programmed using dynamic memory allocation. When the program starts, the dimension parameters are read and the variables arrays will be allocated with appropriate size. The parameters to control the program flow are the following.

**moving** This is the switch parameter for moving grid system. The moving grid system is turned off when moving=0. moving=1 is for forced moving and moving=2 is for induced moving.

**unidt** This is the switch parameter for local time step. When unidt=1, a uniform time step is used for the whole computation domain. When unidt=0, the local time step is used.

**dual_t** This controls the dual time stepping method. When dual_t=0, the dual time stepping is turned off for steady state calculation. The inner iteration total step number "tsteps" is set to 1. The dual time stepping is turned on when dual_t=1 for unsteady calculation. The option dual_t=2 is for unsteady calculation without dual time stepping. The physical time is precisely marched in unsteady calculations. Therefore, the local time stepping is turned off when dual_t=2.

**strtp** This parameter controls the structural type in the fluid-interaction calculation. The currently available options are 1 for cylinder, 2 for single airfoil and 3 for oscillating cascade.

## C.2 Code List

The subroutines and functions used in the latest RANS3D solver version 4 are listed as the following.

**BLOCK** SUBROUTINE block.f
Solves block tridiagonal system.
**call:** DGETRF, DGETRI

**BOUNDARY** SUBROUTINE boundary.f
Boundary condition for viscous terms.
**call:** SURFACE

**BOUND_INVISCID** SUBROUTINE bound_inviscid.f
Boundary condition for inviscid terms.

**CDLTJ** SUBROUTINE cdltj.f
Calculate the drag, lift and torque coefficients during run time.

**COMPUTE_QT** SUBROUTINE compute_qt.f
Compute moving grid velocity at each computation cell center.
**call:** SPVEL

**COMPUTE_SOURCE** SUBROUTINE compute_source.f
Compute k-omega turbulence model source term.
**call:** METRIC_ALL

**CROSS_PRODUCT** SUBROUTINE yplus.f
Calculate the the cross product of two vectors.

**DERIVATIVE** SUBROUTINE derivative.f
Take derivative using 2nd order polynomial method.

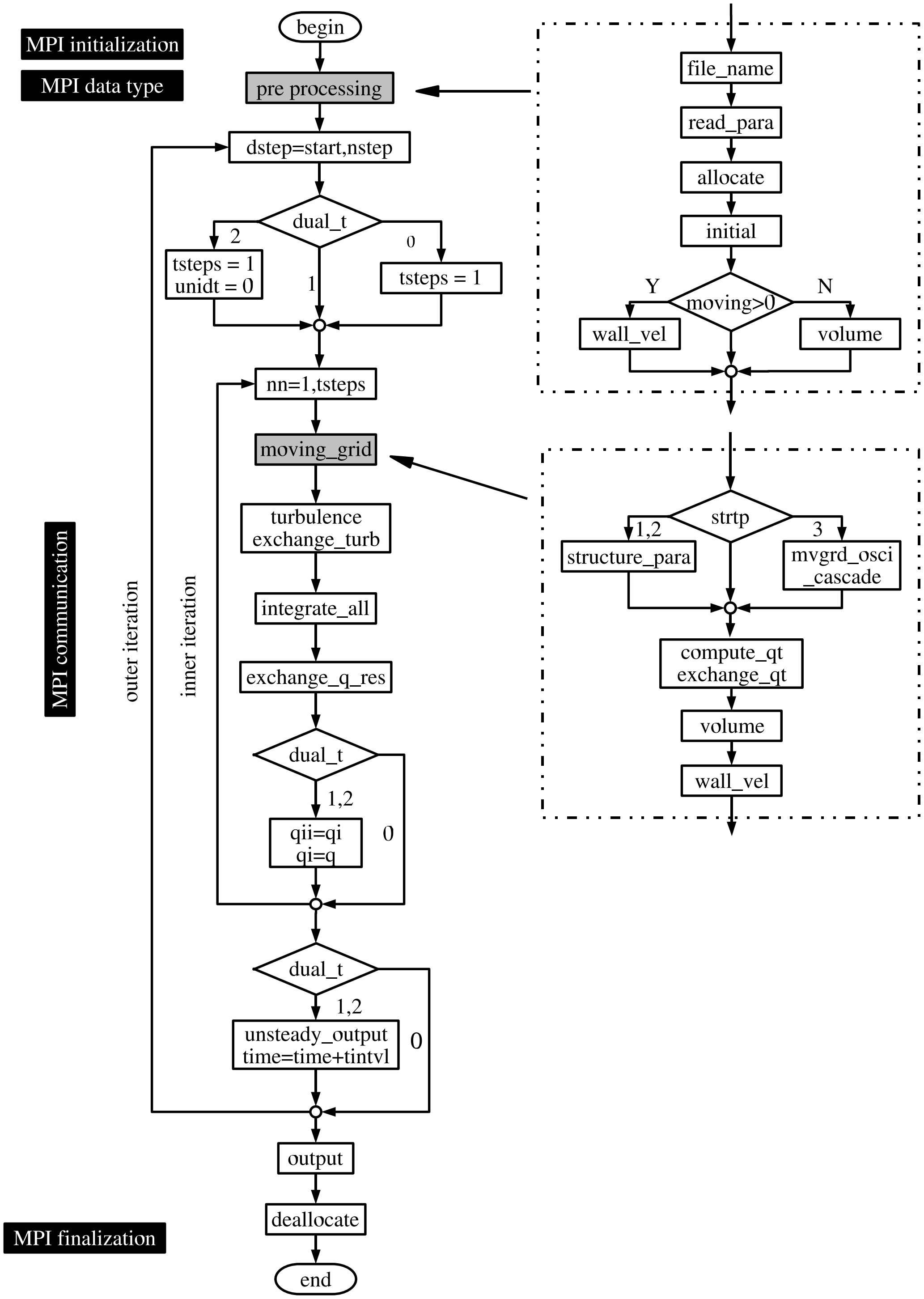


Figure C.1: RANS3D solver flowchart

**DGEMM** SUBROUTINE block.f
**call:** XERBLA

**DGEMV** SUBROUTINE block.f
**call:** XERBLA

**DGER** SUBROUTINE block.f
**call:** XERBLA

**DGETF2** SUBROUTINE block.f
**call:** DGER, XERBLA

**DGETRF** SUBROUTINE block.f
**call:** DGEMM, DGETF2, DLASWP, DTRSM, XERBLA

**DGETRI** SUBROUTINE block.f
**call:** DTRSM, DTRTRI, XERBLA

**DISTANCE** SUBROUTINE distance.f
Compute the distance from inner cell to wall, used for turbulence viscosity.

**DISTOPLANE** FUNCTION yplus.f
Compute the distance from an inner grid point to boundary surface.
**call:** CROSS_PRODUCT

**DLASWP** SUBROUTINE block.f

**DSCAL** SUBROUTINE block.f
Scales a vector by a constant. Uses unrolled loops for increment equal to one.

**DSWAP** SUBROUTINE block.f
Interchanges two vectors. Uses unrolled loops for increments equal one.

**DTRMM** SUBROUTINE block.f
**call:** XERBLA

**DTRMV** SUBROUTINE block.f
**call:** XERBLA

**DTRSM** SUBROUTINE block.f
**call:** XERBLA

**DTRTI2** SUBROUTINE block.f
**call:** DSCAL, DTRMV, XERBLA

**DTRTRI** SUBROUTINE block.f
**call:** DTRMM, DTRSM, DTRTI2, XERBLA

**EXCHANGE_DT** SUBROUTINE mpi_exchange.f
Exchange locally determined time step information in parallel computation.
**call:** MPI_RECV, MPI_SEND

**EXCHANGE_QT** SUBROUTINE mpi_exchange.f
Communication for moving grid information.
**call:** MPI_RECV, MPI_SEND, MPI_SENDRECV

**EXCHANGE_Q_RES** SUBROUTINE mpi_exchange.f
Communication for primitive variables and residual information.
**call:** MPI_RECV, MPI_SEND, MPI_SENDRECV

**EXCHANGE_VISTURB** SUBROUTINE mpi_exchange.f
Communication for turbulence viscosity.
**call:** MPI_SENDRECV

**GAUSS_SEIDEL** SUBROUTINE strmodel.f
This subroutine is used to solve linear system by Gauss Seidel iteration.

**GDSN** SUBROUTINE gdsn.f
Generate dsn() & sn() in the way, the clustering is at the bottom.

**INITIAL** SUBROUTINE initial.f
Read rstart to initialize flowfield.

**INTEGRATE_ALL** SUBROUTINE integrate_all.f
Integrates equations for one time step from time level "n" to time level "n+1".
**call:** EXCHANGE_DT, INVERT, NEGATIVE, OMESH_PERI, RHSIDE, TIMESTEP, UPDATERK, URSN

**INV3** SUBROUTINE accessory.f
Subroutine for rank-3 matrix inverse.

**INVERT** SUBROUTINE invert.f
Defines and solves the linear system for artificial-factorization.
**call:** BLOCK, LHS_MATRIX

**INVSID_FLUX** SUBROUTINE invsid_flux.f
Calculate the inviscid flux.
**call:** ROE_MATRIX, VANLEER_HANEL_FLUX, VAN_LEER_FLUX

**KW_WALL** SUBROUTINE kw_wall.f
Wall treatment calculation and its influence on RHS matrix. More treatment about lhs matrix is applied in matrix_bnd.
**call:** METRIC_ALL, WALL_FUNC

**LHS_MATRIX** SUBROUTINE lhs_matrix.f
Calculate the LHS matrices.
**call:** BOUND_INVISCID, MATRIX_BND, METRIC, RECONSTRUCT, ROE_MATRIX, VAN_LEER_MATRIX, ZHA_MATRIX

**LUDECO** SUBROUTINE pbtrip.f

Calculate L-U decomposition of a given matrix A and store result in A (no pivoting strategy is employed).

**LUSOLV** SUBROUTINE pbtrip.f

Solve linear algebraic system of equations A*C=B and store results in vector C. Matrix A is input in L-U decomposition form (no pivoting strategy has been employed to compute the L-U decomposition of the matrix a).

**MAIN** MAIN PROGRAM main.f

Solver main program.
**call:** EXCHANGE_Q_RES, EXCHANGE_VISTURB, INITIAL, INTEGRATE_ALL, MOVING_GRID, MPI_DATA_TYPE, MPI_FINAL, MPI_INITIAL, OUTPUT, READ_PARA, RECORD_P, TURBULENCE, VOLUME, WALL_VEL

**MATRIX_BND** SUBROUTINE matrix_bnd.f

Special boundary treatment for LHS matrices.

**METRIC** SUBROUTINE metric.f

Compute metrics on xi, eta or zeta faces.

**METRIC_ALL** SUBROUTINE metric_all.f

Compute metrics for all cell centers in computation domain.

**MOVING_GRID** SUBROUTINE moving_grid.f

Update grid related info.
**call:** CDLTJ, COMPUTE_QT, EXCHANGE_QT, MVGRID, MVGRID_OSC_CASCADE, STRUCTURE_PARA, VOLUME, WALL_VEL

**MPI_DATA_TYPE** SUBROUTINE mpi_exchange.f

Create derived data types for MPI communication.
**call:** MPI_TYPE_COMMIT, MPI_TYPE_CONTIGUOUS, MPI_TYPE_EXTENT, MPI_TYPE_HVECTOR, MPI_TYPE_VECTOR

**MPI_FINAL** SUBROUTINE mpi_exchange.f

Terminate MPI environment.
**call:** MPI_FINALIZE

**MPI_INITIAL** SUBROUTINE mpi_exchange.f

**call:** MPI_COMM_RANK, MPI_COMM_SIZE, MPI_INIT

**MULPUT** SUBROUTINE pbtrip.f

Multiply a vector B by a matrix A. Subtract result from another vector C and store result in C. Vector C is overwritten.

**MVGRID** SUBROUTINE mvgrid.f

Update the grid points at each time step.
**call:** GDSN

**MVGRID_OSC_CASCADE** SUBROUTINE mvgrid_osc_cascade.f
Update oscillating blade mesh information.

**NEGATIVE** SUBROUTINE negative.f
Checks for negative density, pressure and temperature.

**OMESH_BND** SUBROUTINE omesh_bnd.f
Find match point for o-mesh periodic boundary condition.

**OMESH_PERI** SUBROUTINE omesh_peri.f
O-type periodic BC, q value is assigned to the ghost cells from the matched points.
**call:** OMESH_BND

**OUTPUT** SUBROUTINE output.f
Write boundary conditions, mesh and flowfield variables into a file.

**PBTRIP** SUBROUTINE pbtrip.f
Solve periodic block tridiagonal system of equations without pivoting strategy. Each block matrix may be of dimension n with n any number greater than 1.
**call:** LUDECO, LUSOLV, MULPUT

**PRINT_UNSTEADY** SUBROUTINE print_unsteady.f
Record necessary instantaneous unsteady variables for postprocessing.

**READ_PARA** SUBROUTINE read_para.f
Read parameters in datain.

**RECONSTRUCT** SUBROUTINE reconstruct.f
Linear reconstruction to the cell faces using conservative variables.

**RECORD_P** SUBROUTINE record_p.f
Record pressure history for oscillating cascade.

**RHSIDE** SUBROUTINE rhside.f
Computes right-hand-side of the linear system and stores in array rhs.
**call:** BOUNDARY, BOUND_INVISCID, INVSID_FLUX, KW_WALL, METRIC, RECONSTRUCT, SOURCE

**ROE_MATRIX** SUBROUTINE roe_matrix.f
Compute Roe matrix.

**SAVE_MESH** SUBROUTINE save_mesh.f
Save mesh information of the previous time step. Compute boundary velocity.

**SOURCE** SUBROUTINE source.f
Calculate the source term due to the geometric conservation law.

**SPLINE** SUBROUTINE spline.f
Interpolation using spline method.

**SPVEL** SUBROUTINE spvel.f
Calculate the grid velocity due to the motion of moving grid.

**STRMODEL** SUBROUTINE strmodel.f
Solve the governing equations for two-degree-of-freedom model.
**call:** GAUSS_SEIDEL

**STRUCTURE_PARA** SUBROUTINE structure_para.f
Compute structure parameters.
**call:** STRMODEL

**SURFACE** SUBROUTINE surface.f
Computes velocity components and static temperature at fictitious cells.

**SYSTIME** FUNCTION accessory.f
Obtain system time and output to easy-to-read format.
**call:** DATE_AND_TIME

**TIMESTEP** SUBROUTINE timestep.f
Calculate local time step on each cell.
**call:** METRIC

**TURBULENCE** SUBROUTINE turbulence.f
Compute turbulence viscosity.
**call:** DISTANCE, METRIC_ALL, TURB_LINE

**TURBULENCEN** SUBROUTINE turbulencen.f
Compute turbulence viscosity in the second way.
**call:** VORTM

**TURB_LINE** SUBROUTINE turb_line.f
Compute B_L turbulent viscosity along a line normal to wall.

**UNSTEADY** SUBROUTINE unsteady.f
Record custom unsteady parameters.
**call:** DERIVATIVE, SPLINE

**UPDATERK** SUBROUTINE updaterk.f
MPI communication for RK method.
**call:** MPI_SENDRECV

**URSN** SUBROUTINE ursn.f
Use Upwind Relaxation Sweeping(URS) with Gauss-Seidel to invert the implicit LHS.
**call:** BLOCK, LHS_MATRIX, PBTRIP

**VANLEER_HANEL_FLUX** SUBROUTINE vanleer_hanel_flux.f
Calculate van Leer-Hänel flux vector splitting at subsonic.

**VAN_LEER_FLUX** SUBROUTINE van_leer_flux.f
Calculate van Leer flux vector splitting at subsonic.

**VAN_LEER_MATRIX** SUBROUTINE van_leer_matrix.f
Calculate LHS matrix for van Leer flux vector splitting at subsonic.

**VOLUME** SUBROUTINE volume.f
Compute volume.

**VORTM** SUBROUTINE vortm.f
Calculate the vorticity at the center of each cell.

**WALL_FUNC** SUBROUTINE wall_func.f
Wall treatment calculation and its influence on rhs matrix.

**WALL_VEL** SUBROUTINE wall_vel.f
Compute wall boundary velocity for moving grid.

**XERBLA** SUBROUTINE block.f

**YPLUS** SUBROUTINE yplus.f
Calculate yplus of the first cell center in wall region.

**ZHA_MATRIX** SUBROUTINE zha_matrix.f
Compute Zha matrix, dhat_l and dhat_r.

# Appendix D

# Publications

## Journal Papers

1. Hu, Z. and Zha, G., "Calculations of 3D Compressible Flows Using an Efficient Low Diffusion Upwind Scheme," *International Journal for Numerical Methods in Fluids*, Vol. 107, 2005, pp. 253–269.

2. Zha, G. and Hu, Z., "Calculation of Transonic Internal Flows Using an Efficient High-Resolution Upwind Scheme," *AIAA Journal*, Vol. 42, No. 2, 2004, pp. 205–214, also AIAA Paper 2004-1097.

## Conference Papers

1. Hu, Z., Zha, G., and Lepicovsky, J., "Numerical Study on Flow Separation of a Transonic Cascade," AIAA 42nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 5-8 2004, AIAA Paper 2004–0199.

2. Hu, Z., and Zha., Z., "Simulation of 3D Flows of Propulsion Systems Using an Efficient Low Diffusion E-CUSP Upwind Scheme," 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Fort Lauderdale, Florida, July 2004, AIAA Paper 2004–4082.

3. Hu, Z., Zha, G., and Lepicovsky, J., "Numerical Study of a Cascade Unsteady Separation Flow," ASME Turbo Expo 2004, Power for Land, Sea and Air, Vienna, Austria, June 14–17 2004, ASME Paper GT2004-53195.

4. Chen, X., Zha, G., and Hu, Z., "Numerical Simulation of Flow Induced Vibration Based on Fully Coupled Fluid-Structural Interactions," 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, Oregon, June 28–July 1 2004, AIAA Paper 2004-2240.

5. Chen, X., Zha, G., Hu, Z., and Yang, M., "Flutter Prediction Based on Fully Coupled Fluid-Structural Interactions," 9th National Turbine Engine High Cycle Fatigue Conference, March 16–19 2004.

# References

[1] Weber, S., Benetschik, H., Peitsch, D., and Gallus, H. E., "A Numerical Approach to Unstalled and Stalled Flutter Phenomena in Turbomachinery Cascades," ASME Paper 97-GT-102, 6 1997, The International Gas Turbine & Aeroengine Congress & Exhibition.

[2] Lepicovsky, J., McFarland, E. R., Chima, R. V., Capece, V. R., and Hayden, J., "Intermittent Flow Regimes in a Transonic Fan Airfoil Cascade," NASA/TM-2002-211375, 2 2002.

[3] Sadeghi, M. and Liu, F., "Computation of Mistuning Effects on Cascade Flutter," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 22–28.

[4] Lepicovsky, J., Chima, R. V., Jett, T. A., Bencic, T. J., and Weiland, K. E., "Investigation of Flow Separation in a Transonic-fan Linear Cascade Using Visualization Methods," *NASA/TM-2000-210521*, Dec 2000.

[5] Sanders, A. J., Hassan, K. K., and Rabe, D. C., "Experimental and Numerical Study of Stall Flutter in a Transonic Low-Aspect Ratio Fan Blisk," *Journal of Turbomachinery*, Vol. 126, 2004, pp. 166–174.

[6] Carta, F. O., "Coupled Blade-Disc-Shroud Flutter Instabilities in Turbojet Engine Rotors," *Journal of Engineering for Power*, Vol. 89, No. 3, 1967, pp. 419–426.

[7] Traci, R. M., Albano, E. D., and L., F. J., "Perturbation Method for Transonic Flow about Oscillating Airfoils," *AIAA Journal*, Vol. 14, 12 1976, pp. 1258–1265.

[8] Ballhaus, W. F. and Goorjian, P. M., "Implicit Finite Difference Computations of Unsteady Transonic Flows about Airfoils," *AIAA Journal*, Vol. 15, 1977, pp. 1728–1735.

[9] Farrell, C. and Adamczyk, J., "Full Potential Solution of Transonic Quasi-Three-Dimensional Flow Through a Cascade Using Artificial Compressibility," ASME Paper 81-GT-70, 1981.

[10] Verdon, J. M. and Caspar, J. R., "Development of a Linear Unsteady Aerodynamic Analysis for Finite-Deflection Subsonic Cascades," *AIAA Journal*, Vol. 20, No. 9, 9 1982.

[11] Whitehead, D. S., "A Finite Element Solution of Unsteady Two-Dimensional Flow in Cascades," *International Journal for Numerical Methods in Fluids*, Vol. 10, 1990, pp. 13–34.

[12] Hariharan, S. I. and Ping, Y., "Time Domain Numerical Calculations of Unsteady Vertical Flows about a Flat Plate Airfoil," *Journal of Computational Physics*, Vol. 101, 1992, pp. 419–430.

[13] Ji, S., *Computation of Unsteady Viscous Flows with Aeroelastic Applications*, Ph.D. thesis, University of California at Irvine, 7 1998.

[14] Grüber, B. and Carstens, V., "The Impact of Viscous Effects on the Aerodynamic Damping of Vibrating Transonic Compressor Blades — A Numerical Study," *Journal of Turbomachinery*, Vol. 123, 2001.

[15] Smith, S. N., *Discrete Frequency Sound Generation in Axial Flow Turbomachines*, Ph.D. thesis, Cambridge University, 1971, Engineering Department.

[16] Verdon, J. M. and Caspar, J. R., "A Linearized Unsteady Aerodynamic Analysis for Transonic Cascades," *Journal of Fluid Mechanics*, Vol. 149, 12 1982, pp. 403–429.

[17] Hall, K. C. and Crawley, E. F., "Calculation of Unsteady Flows in Turbomachinery Using the Linearized Euler Equations," *AIAA Journal*, Vol. 27, 1993, pp. 777–787.

[18] Kahl, G. and Klose, A., "Computation of the Linearized Transonic Flow in Oscillating Cascades," ASME Paper 93-GT-269.

[19] Montgomery, M. D. and Verdon, J. M., "A Linearized Unsteady Euler Analysis for Turbomachinery Blade Rows Using an Implicit Wave-Split Scheme," *Proc. 7th Int. Symp. on Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, Elsevier Science, 1995.

[20] Holmes, D. G., Mitchell, B. E., and Lorence, C. A., "Three Dimensional Lineaized Navier-Stokes Calculations for flutter and Forced Response," *Proceedings of the 8th International Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, Stockholm, Sweden, 1997.

[21] Clark, W. S. and Hall, K. C., "A Time-Linearized Navier-Stokes Analysis of Stall Flutter," *Journal of Turbomachinery*, Vol. 122, 2000, pp. 467–476.

[22] Shibata, T. and Kaji, S., "Role of Shock Structures in Transonic Fan Rotor Flutter," *Proceedings of the 8th International Symposium of Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, kluwer, Dordrecht, The Netherlands, 1998, pp. 733–747.

[23] Jourbert, H., "Supersonic Flutter in Axial Flow Compressor," *Proceedings of 3rd Symposium on Unsteady Aerodynamics and Aeroelasticity of Turbomachines and Propellers*, Cambridge, U. K., 1984.

[24] Koya, M. and Kotake, S., "Numerical Analysis of Fully Three-Dimensional Periodic Flows through a Turbine Stage," *Journal of Engineering for Gas Turbines and Power*, Vol. 107, 1985, pp. 945–952.

[25] Gerolymos, G. A., "Numerical Integration of the 3rd Unsteady Euler Equations for Flutter Analysis of Axial Flow Compressors," ASME Paper 88-GT-255, 1988.

[26] He, L., "An Euler Solution for Unsteady Flows around Oscillating Blades," *Journal of Turbomachinery*, Vol. 112, No. 4, 1990, pp. 714–722.

[27] Peitsch, D., Gallus, H. E., and Kau, H. P., "Prediction of Unsteady 2D-Flow in Turbomachinery Bladings," *Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines and Propellers*, edited by H. M. Atassi, Springer, 1993.

[28] Huff, D. L., Swafford, T. W., and Reddy, T. S. R., "Euler Flow Predictions for an Oscillating Cascade Using a High Resolution Wave Split Scheme," ASME Paper 91-GT-198, 1991.

[29] Carstens, V., "Computation of the Unsteady Transonic 2D Cascade Flow by an Euler Algorithm with Interactive Grid Generation," AGARD CP501, Transonic Unsteady Aerodynamics and Aeroelasticity, October 1991, San Diego, USA.

[30] Gerolymos, G. A., "Advances in the Numerical Integration of the 3-D Euler Equations in Vibrating Cascades," ASME Paper 92-GT-170, 1992.

[31] He, L., "New Two-Grid Acceleration Method for Unsteady Navier-Stokes Calculations," *Journal of Propulsion and Power*, Vol. 9, No. 2, March–April 1993, pp. 272–280.

[32] Carstens, V., "Computation of Unsteady Transonic 3D Flow in Oscillating Turbomachinery Bladings by an Euler Algorithm wth Deforming Grids," *Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, edited by Y. Tanida and M. Namba, Elsevier Science, 1995.

[33] Peitsch, D., Gallus, H. E., and Weber, S., "Computation of Unsteady Transonic 3D-Flow in Turbomachine Bladings," *Proc. 7th Int. Symp. on Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, Elsevier Science b.v. Amsterdam, 1995.

[34] Peitsch, D., Gallus, H. E., and Weber, S., "Numerical Investigation of the Unsteady Transonic 3D-Flow in Stator and Rotor Cascades with Oscillating Blades," ASME Paper 96-GT-307, 1996.

[35] Gopalakrishnan, S. and Bozzola, R., "A Numerical Technique for Calculation of Transonic Flows in Turbomachinery," ASME Paper 71-GT-42, 1971.

[36] Veuillot, J. P., "Calculation of Quasi-Three-Dimensional Flow in a Turbomachine Blade Row," ASME Paper 76-GT-56, 1976.

[37] McDonald, P. W., “The Computation of the Transonic Flow Through Two Dimensional Gas Turbine Cascades,” ASME Paper 71-GT-89, 1971.

[38] Denton, J. D. and Singh, U. K., “Time Marching Methods for Turbomachinery Flow Calculation,” VKI Lecture Series on Transonic Flows for Turbomachinery, 1979.

[39] Kopper, F., Milano, R., and Vanco, M., “An Experimental Investigation of Endwall Profiling in a Turbine Vane Cascade,” *AIAA Journal*, Vol. 19, No. 8, 1981.

[40] Sarathy, K. P., “Computation of Three-Dimensional Flow Fields through Rotating Blade Rows and Comparison with Experiment,” ASME Paper 81-GT-121, 1981.

[41] He, L. and Denton, J. D., “Inviscid-Viscous Coupled Solution for Unsteady Flows through Vibrating Blades - Part 1: Description of the Method,” ASME Paper 91-GT-125, 6 1991, International Gas Turbine and Aeroengine Congress and Exposition.

[42] Ekaterinaris, J. A. and Platzer, M. F., “Progress in the Analysis of Blde Flutter,” *UAAT-Symposium*, Fukuoka, Japan, 1994.

[43] Abhari, R. S. and Giles, M., “A Navier-Stokes Analysis of Airfoil in Oscillating Transonic Cascades for the Prediction of Aerodynamic Damping,” *Journal of Turbomachinery*, Vol. 119, 1997, pp. 77–84.

[44] Brenneis, A. and Eberle, A., “Application of an Implicit Relaxation Method Solving the Euler Equations for Time-Accurate Unsteady Problems,” *Journal of Fluids Engineering*, Vol. 112, December 1990, pp. 510–520.

[45] Ji, S. and Liu, F., “Flutter computation of turbomachinery cascades using a parallel unsteady Navier-Stokes code,” *AIAA Journal*, Vol. 37, No. 3, 1999, pp. 320–327.

[46] Rai, M. M., “Unsteady Three-Dimensional Navier-Stokes Simulaition of Turbine Rotor-Stator Interaction,” AIAA Paper 87-2058.

[47] Giles, M. and Haimes, R., “Validation of a Numerical Method for Unsteady Flow Calculation,” *Journal of Turbomachinery*, Vol. 115, 1993, pp. 110–117.

[48] Cho, N. H., Liu, X., Rodi, W., and Schonung, B., “Calculation of Wake-Induced Unsteady Flow in a Turbine Cascade,” *Journal of Turbomachinery*, Vol. 115, 1993, pp. 675–686.

[49] Huff, D. L., “Numerical Simulations of Unsteady, viscous, Transonic Flow over Isolated and Cascaded Airfoils using a Deformed Grid,” AIAA Paper 87-1316.

[50] Arnone, A., Liou, M.-S., and Povinelli, L. A., “Multigrid Calculation of Three-Dimensional Viscous Cascade flows,” AIAA Paper 91-3238, 1991.

[51] Kikuchi, O., Nakahashi, K., and Tamura, A., “Navier-Stokes Computations of Two- and Three-Dimensional Cascade Flowfields,” *Journal of Propulsion and Power*, Vol. 5, No. 3, 1989, pp. 320–326.

[52] Rai, M. M. and Madavan, N. K., "Multi-Airfoil Navier-Stokes Simulations of Turbine Rotor-Stator Interaction," *Journal of Turbomachinery*, Vol. 112, 1990, pp. 377–384.

[53] Scott, J. N. and Hankey, W. L. J., "Navier-Stokes Solutions of Unsteady Flow in Compressor Rotor," *Journal of Turbomachinery*, Vol. 108, 1986, pp. 206–215.

[54] Hwang, C. J. and Liu, J. L., "Analysis of Steady and Unsteady Turbine Cascade Flows by a Locally Implicit Hybrid Algorithm," *Journal of Turbomachinery*, Vol. 115, 10 1993, pp. 699–706.

[55] He, L. and Denton, J. D., "Three-Dimensional Time-Marching Inviscid and Viscous Solutions for Unsteady Flows around Vibrating Blades," *Journal of Turbomachinery*, Vol. 116, 1994, pp. 469–476.

[56] Baldwin, B. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent flows," AIAA Paper 78-257, 1978.

[57] Baldwin, B. S. and Barth, T. J., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," NASA TM 102847, 1990.

[58] Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, 1992.

[59] Wilcox, D. C., "Reassessment of the scale-determining equations for advanced turbulence models," *AIAA Journal*, Vol. 26, 1988, pp. 1299–1310.

[60] Ekaterinaris, J. A. and Platzer, M. F., "Computational Prediction of the Airfoil Dynamic Stall," *Prog. Aerosp. Sci.*, Vol. 33, 1997, pp. 759–846.

[61] Sanz, W. and Platzer, M. F., "On the Navier-Stokes Calculation of Separation Bubble With a New Transition Model," *Journal of Turbomachinery*, Vol. 120, 1998, pp. 36–42.

[62] Weber, S., Jones, K. D., Ekaterinaris, J. A., and Platzer, M. F., "Transonic Flutter Computations for a 2D Supercritical Wing," AIAA Paper 99-0798, 1999.

[63] Weber, S. and Platzer, M. F., "A Navier-Stokes Analysis of the Stall Flutter Characteristics of the Buffum Cascade," *Journal of Turbomachinery*, Vol. 122, 2000.

[64] Eguchi, T. and Wiedermann, A., "Numerical Analysis of Unstalled and Stalled Flutter Using a Navier-Stokes code with Deforming Meshes," *UATT-Symposium*, Fukuoka, Japan, 1994.

[65] Grüber, B. and Carstens, V., "Computation of the Unsteady Transonic Flow in Harmonically Oscillating Turbine Cascades Taking the Viscous Effects," *Journal of Turbomachinery*, Vol. 120, No. 1, 1998, pp. 104–111.

[66] Wu, C. H., "A General Through-Flow-Theory of Fluid Flow with Subsonic or Supersonic Velocity in Turbomachines of Arbitrary Hub and Casing Shapes," Tech. Rep. TN-2302, NACA, 1951.

[67] Erdos, J. I., Alzner, E., and McNally, W., "Numerical Solution of Periodic Transonic Flow through a Fan Stage," *AIAA Journal*, Vol. 15, 1977, pp. 1559–1568.

[68] Silkowski, P. D., Rhie, C. M., Copeland, G. S., Eley, J. A., and Bleeg, J. M., "CFD Investigation of Aeromechanics," ASME Paper GT-2001-0267, 2001.

[69] Srivastava, R., Bakhle, M. A., Keith Jr., T. G., and Stefko, G. L., "Flutter Analysis of a Transonic Fan," ASME Paper GT-2002-30319, 2002.

[70] Alonso, J., Martinelli, L., and Jameson, A., "Multigrid Unsteady Navier-Stokes Calculations with Aeroelastic Applications," AIAA Paper 95-0048, 1995.

[71] Jameson, A., "Transonic Flow Calculations," Tech. Rep. MAE 1651, Princeton University, 1984.

[72] Ni, R.-H., "A Multiple-Grid Scheme for Solving the Euler Equations," AIAA Paper 81-1025, 1981.

[73] Li, C. P., "Numerical Solution of Viscous Reacting Blunt Body Flows of a Multicomponent Mixture," AIAA Paper 73-202, 1973.

[74] Steger, J. L., "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, 1978, pp. 679–686.

[75] Chyu, W. J., Davis, S. S., and Chang, K. S., "Calculation of Unsteady Transonic Flow over an Airfoil," *AIAA Journal*, Vol. 19, 1981, pp. 684–690.

[76] Jameson, A., "Time Dependent Calculations Using Multigrid and Algebraic Model for Separated Turbulent Flows," AIAA Paper 91-1596, 1991.

[77] Liou, M. S., Arnone, A., and Povinelli, L. A., "Multi-grid Time-accurate Integration of Navier-Stokes Equations," AIAA Paper 93-3361-CP, 7 1993.

[78] McBean, I. W., Liu, F., Hourigan, K., and Thompson, M., "Navier-Stokes Simulation of 2-D Unsteady Aerodynamics of a Turbine Cascade," 14th Australasian Fluid Mechanics Conference, 12 2001.

[79] Cinnella, P., Palma, P. D., Pascazio, G., and Napolitano, M., "A Numerical Method for Turbomachinery Aeroelasticity," *Journal of Turbomachinery*, Vol. 126, 2004, pp. 310–316.

[80] Isomura, K. and Giles, M. B., "A Numerical Study of Flutter in a Transonic Fan," *Journal of Turbomachinery*, Vol. 120, 1999, pp. 500–507.

[81] Tuncer, I. H., Weber, S., and Sanz, W., “Investigation of Periodic Boundary Conditions in Multipassage Cascade Flows Using Overset Grids,” *Journal of Turbomachinery*, Vol. 121, 1999, pp. 341–347.

[82] Carstens, V. and Schmitt, S., “Comparison of Theoretical and Experimental Data for an Oscillating Transonic Compressore Cascade,” ASME Paper 99-GT-408, 1999.

[83] Lin, J. S. and Murthy, D. V., “Unsteady Aerodynamic Analysis of an Oscillating Cascade at Large Incidence,” ASME Paper 99-GT-22, 1999.

[84] Gerolymos, G. A. and Vallet, I., “Validation of 3D Euler Methods for Vibrating Cascade Aerodynamics,” *Journal of Turbomachinery*, Vol. 118, 1996, pp. 771–782.

[85] Chuang, H. A. and Verdon, J. M., “A Nonlinear Simulator for Three-Dimensional Flows Through Vibrating Cascades,” *Journal of Turbomachinery*, Vol. 121, 1999, pp. 348–357.

[86] Mcbean, I. W., Liu, F., and Thompson, M. C., “A Three Dimensional Navier-Stokes Code for Aeroelasticity in Turbomachinery,” *International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, 9 2000.

[87] McBean, I., Liu, F., Hourigan, K., and Thompson, M., “Simulations of Aeroelasticity in an Annular Cascade Using a Parallel 3-Dimensional Navier-Stokes Solver,” ASME Paper GT-2002-30366, 2002.

[88] Lane, F., “System Mode Shapes in the Flutter of Compressor Blade Rows,” *Journal of the Aeronautical Sciences*, Vol. 23, 1956, pp. 54–56.

[89] Giles, M., “Stator/rotor Interaction in a Transonic Turbine,” *Journal of Propulsion and Power*, Vol. 6, No. 5, 1990.

[90] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI, Portable Parallel Programming with the Message-Passing Interface*, MIT Press, 1996.

[91] Roe, P., “CFD Algorithms,” NASA Spring Workshop on Fluids, University of Alabama, Birmingham, Alabama, USA, April 22-24, 2003.

[92] Zha, G., *Numerical Solutions of Three-Dimensional Transonic Flows Based on Upwind Schemes*, Ph.D. thesis, University of Montreal, December 1993.

[93] van Leer, B., “Flux Vector Splitting for the 1990’s,” *Proceedings of the Computational Fluid Dynamics Symposium on Aeropropulsion NASA CP-3078*, 1990, pp. 203–214.

[94] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Computatoinal and Physical Processes in Mechanics and Thermal Sciences, Taylor & Francis, Philadephia, PA, 2nd ed., 1997.

[95] Roe, P., “Approximate Riemann Solvers, Parameter vectors, and difference schemes,” *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[96] Van Leer, B., Thomas, J., Roe, P. L., and Newsome, R., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," AIAA paper 87-1104, 1987.

[97] van Leer, B., "Flux-vector splitting for the Euler equations," *Lecture Note in Physics*, Vol. 170, 1982.

[98] Zha, G. and Bilgen, E., "Numerical Solutions of Euler Equations by Using a New Flux Vector Splitting Scheme," *International Journal for Numerical Methods in Fluids*, Vol. 17, 1993, pp. 115–144.

[99] Hänel, D., Schwane, R., and Seider, G., "On the Accuracy of Upwind Schemes for the Solution of the Navier-Stokes Equations," AIAA Paper 87-1105 CP, 1987.

[100] Liou, M. S. and Steffen, C. J., "A New Flux Splitting Scheme," *Journal of Computational Physics*, Vol. 107, 1993, pp. 1–23.

[101] Wada, Y. and Liou, M. S., "An Accurate and Robust Splitting Scheme for Shockand Contact Discontinuities," AIAA Paper 94-0083, 1994.

[102] Liou, M. S., "Progress Towards an Improved CFD Methods: AUSM$^+$," AIAA Paper 95-1701-CP, June, 1995.

[103] Liou, M. S., "A Sequel to AUSM: AUSM$^+$," *Journal of Computational Physics*, Vol. 129, 1996, pp. 364–382.

[104] Liou, M. S., "Ten Years in the Making- AUSM family," AIAA Paper 2001-2521, 2001.

[105] Edwards, J. R., "A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations," *Computer & Fluids*, Vol. 6, 1997, pp. 635–659.

[106] Jameson, A., "Analysis and Design of Numerical Schemes for GasDynamics I: Artificial Diffusion, Upwind Biasing,Limiters and Their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flow," *Journal of Computational Fluid Dynamics*, Vol. 4, 1995, pp. 171–218.

[107] Jameson, A., "Analysis and Design of Numerical Schemes for GasDynamics II: Artificial Diffusion and Discrete Shock Structure," *Journal of Computational Fluid Dynamics*, Vol. 5, 1995, pp. 1–38.

[108] Zha, G., "Numerical Tests of Upwind Scheme Performance for Entropy Condition," *AIAA Journal*, Vol. 37, 1999, pp. 1005–1007.

[109] Zha, G., "Comparative Study of Upwind Scheme Performance for Entropy Condition and Discontinuities," AIAA Paper 99-CP-3348, June 28- July 1, 1999.

[110] Zha, G. and Hu, Z., "Calculation of Transonic Internal Flows Using an Efficient High Resolution Upwind Scheme," *AIAA Journal*, Vol. 42, No. 2, 2004, pp. 205–214.

[111] Chen, X. and Zha, G., "Fully Coupled Fluid-Structural Interactions Using an Efficient High Resolution Upwind Scheme," AIAA Paper 2004-2331, 2004, 34th AIAA Fluid Dynamics Conference.

[112] Zha, G., "A Low Diffusion Efficient E-CUSP Upwind Scheme for Transonic Flows," AIAA Paper 2004-2707, 2004.

[113] White, F., *Viscous Fluid Flow*, McGraw-Hill, New York, 1974.

[114] Favre, A., "Equations des Gaz Turbulents Compressibles," *Journal de Mecanique*, Vol. 4, 1954, pp. 361–390.

[115] Hoffmann, K. A. and Chiang, S. T., *Computational Fluid Dynamics*, Vol. II, Engineering Education System, 4th ed., 2000.

[116] Thomas, P. and Lombard, C., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[117] Baysal, O. and Hoffman, W. B., "Simulation of Three-Dimensional Shear Flow Around a Nozzle-Afterbody at High Speeds," *Journal of Fluids Engineering*, Vol. 114, 1992, pp. 178–185.

[118] Chen, J. P. and Whitfield, D. L., "Navier-Stokes Calculations for the Unsteady Flowfield of Turbomachinery," AIAA Paper 93-0676, 1993.

[119] Chen, X., Zha, G., and Hu, Z., "Numerical Simulation of Flow Induced Vibration Based on Fully Coupled Fluid-Structural Interactions," AIAA Paper 2004-2240, June 2004.

[120] van Leer, B., "Towards the Ultimate Convservative Difference Scheme, V: A Second-Order Sequel to Godunov's Method," *Jouranl of Computational Physics*, Vol. 32, 1979, pp. 101–136.

[121] Anderson, W., Thomas, J., and van Leer, B., "A Comparison of Finite-Volume Flux-Vector Splittings for the Euler Equations," AIAA Paper 85-0122, 1985.

[122] Van Leer, B., "Towards the Ultimate Conservative DifferenceScheme, III," *Journal of Computational Physics*, Vol. 23, 1977, pp. 263–75.

[123] Anderson, W. K., Thomas, J. L., and van Leer, B., "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal*, Vol. 24, No. 9, 1986, pp. 1453–1460.

[124] Morton, S. A., Melville, R. B., and Visbal, M. R., "Accuracy and Coupling Issues of Aeroelastic Navier-Stokes Solutions on Deforming Meshes," AIAA Paper-97-1085, 1997.

[125] Hoffmann, K. A., *Computational Fluid Dynamics*, Vol. III, Engineering Education System, 4th ed., 2000.

[126] Bogar, T., Sajben, M., and Kroutil, J., “Characteristic Frequency and Length Scales in Transonic Diffuser Flow Oscillations,” AIAA Paper 81-1291, 1981.

[127] Davis, S. S., “NACA 64 A010 (NACA Ames Model) Oscillatory Pitching,” Tech. Rep. AGARD Report No. 702, AGARD, August 1982.

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

[129] Mason, M. L. and Putnam, L. E., “The Effect of Throat Contouring on Two-Dimensional Converging-Diverging Nozzles at Static Conditions,” NASA Technical Paper 1704, 1980.

[130] Burley II, J. R., Bangert, L. S., and Carlson, J. R., “Static Investigation of Circular-to-rectangular transition ducts for high-aspect-ratio nonaxisymmetric nozzles,” NASA Technical Paper 2534, 1986.

[131] Buffum, D. H., Capaece, V. R., and El-Aini, Y. M., “Oscillating Cascade Aerodynamics at Large Mean Incidence,” *Journal of Turbomachinery*, Vol. 120, 1998, pp. 122–130.

[132] Délery, J. M., “Experimental Investigation of Turbulence properties in transonic shock/boundary-layer interaction,” *AIAA Journal*, Vol. 21, 1983.

[133] Benay, R., Délery, J., and Pot, T., “Etude Expérimentale d’une interaction onde de choc-couche limite en canal tridemensionnel. Propriétś du champ moyen.” Rapport Technique no 70/7078 AN, ONERA, 1986.

[134] Gacherieu, C. and Weber, C., “Assessment of Algebraic and One-Equation Turbulence Models for the Transonic Turbulent Flow around a Full Aircraft Configuration,” AIAA Paper 98-2737, 1998.

[135] Chima, R. V., McFarland, E. R., Wood, J. R., and Lepicovsky, J., “On flowfield periodicity in the NASA transonic flutter cascade, Part II–Numerical study,” Technical report NASA/TM–2000-209933, National Aeronautics and Space Administraion, Glenn Research Center, Cleveland, Ohio 44135, 2000.

[136] Hu, Z., Zha, G., and Lepicovsky, J., “Numerical study on flow separation of a transonic cascade,” Jan 2004, AIAA Paper 2004–0199.

[137] Lepicovsky, J., McFarland, E. R., Capece, V. R., and Hayden, J., “Unsteady Pressures in a Transonic Fan Cascade Due to A Single Oscillating Airfoil,” ASME Paper GT-2002-30312, 2002.

[138] Lepicovsky, J., Capece, V. R., and Ford, C. T., “Resonance Effects in the NASA Transonic Flutter Cascade Facility,” ASME Paper GT-2003-38344, 2003.

[139] Carta, F. O., “Aeroelasticity And Unsteady Aerodynamics,” United Technologies Research Center, 1988.