Application of High-Order Summation-by-Parts Operators to the Steady Reynolds-Averaged Navier-Stokes Equations

Xiaoyue Shen

Supervisor: Prof. David W. Zingg

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science, Graduate Department of Aerospace Science and Engineering, University of Toronto

Abstract

Third- and fourth-order methods have been implemented to solve the steady Reynolds-averaged Navier-Stokes equations with the Spalart-Allmaras turbulence model. Summation-by-parts operators are used for spatial discretization along with simultaneous approximation terms to enforce boundary and interface conditions in a weak sense. Two validation cases provided by the Turbulence Modelling Resource website are tested to verify the high-order implementation of the Spalart-Allmaras model and to examine the efficiency and robustness of the high-order methods. They are a two-dimensional zero pressure gradient flat plate and a two-dimensional bump in a channel. Each case has a grid family consisting of four grid levels. Numerical errors are estimated via grid convergence studies. Compared to the results obtained by the second-order method, it is shown that raising the spatial discretization to third-order global accuracy leads to a substantial reduction in numerical error on a given mesh. The use of the fourth-order method also shows the potential of increasing numerical accuracy over the second-order method on coarse meshes. The computational cost of each method is measured in terms of total computational time. High-order methods are overall more expensive to solve given the same mesh size. However, the results demonstrate that the high-order methods are more efficient as the same level of accuracy can be achieved within less computational time on a coarser mesh.

Contents

List of Figures

List of Tables

Nomenclature

$\mathbf{A_1}$	first-order flow Jacobian matrix
$\mathbf{E}$, $\mathbf{F}$, $\mathbf{E}$	inviscid flux vectors
$\mathbf{E_v}$, $\mathbf{F_v}$, $\mathbf{E_v}$	viscous flux vectors
$\mathbf{Q}$	conservative flow variable vector
$\mathbf{R}$	residual vector
Δx, h	mesh size
γ	specific heat ratio
κ	Von Karman constant
μ	dynamic viscosity
μ_t	turbulent eddy viscosity
ρ	density
τ	stress tensor
$\tilde{\mu}$	turbulence model working variable
a	speed of sound
A_D	artificial dissipation
c	reference chord length
C_d, C_D	drag coefficient
C_l, C_L	lift coefficient
d	nearest wall distance

D_1, D_2	SBP operators for first and second derivatives
D_b	approximation to the first derivative at a boundary
e	energy
f	functional
H	norm matrix
I	identity matrix
M	Mach number
N	number of nodes
p	pressure
Re	Reynolds number
S, $\tilde{S}$	magnitude of vorticity, modified magnitude of vorticity
S^*	rate of strain tensor
S_t	Sutherland constant
T	Temperature
U, V, W	contravariant velocity components
u, v, w	Cartesian velocity components
CFD	Computational Fluid Dynamics
DG	Discontinuous Galerkin
HO	High Order
RANS	Reynolds-Averaged Navier-Stokes
SA	Spalart-Allmaras
SAT	Simultaneous-Approximation-Term
SBP	Summation-by-Parts
TMR	Turbulence Modelling Resource

Chapter 1

Introduction

1.1 Motivation

Air travel has been forecast to experience rapid growth in the the next few decades. Air travel as measured in Passenger-Kilometres (PKM) was 40 trillion PKM back in 2010 and is expected to reach 103 trillion PKM by 2050 [1]. As a result of rapid growth in air travel, annual carbon emissions from commercial aviation are estimated to increase from 670 million tonnes in 2010 to 1.2-1.5 billion tonnes by 2025 [1]. There is a great need to address the environmental impact posed by the increased demand of commercial aviation through technological innovations, such as optimized aircraft designs, new engine technologies, alternative biofuels and materials. During the past four decades, computational fluid dynamics (CFD) programs have been used to complement wind tunnel testing in the process of aircraft design and optimization. Enormous efforts have been made to make CFD software using second-order methods more robust and affordable. However, there is no evidence showing that we should stop at second-order method or it is the optimal option in terms of accuracy and efficiency. In fact, many believe that high-order methods will complement low-order methods in the future and the use of mesh and order adaptation is the optimal approach to realize the promise of high-order methods [2].

1.2 Project Description

As part of the group code Jetstream, the flow solver Diablo uses a multi-block finite difference spatial differentiation method. The overall grid domain is divided into multiple blocks and these blocks are joined together with interface conditions. Summation-by-parts (SBP) operators with simultaneous-approximation terms (SATs) are used for spatial discretization, which guarantees

time stability. The present flow solver is able to apply second-order and high-order accuracy to inviscid and viscous flows. The ultimate goal of this thesis project is to investigate the use of high-order SBP methods in the solution of Reynolds-averaged Navier-Stokes (RANS) equations. A loose definition of a high-order method is that the order of accuracy is equal to or higher than 3. With a given mesh size, a high-order method normally takes longer to obtain a converged solution; however, it can reach the same level of accuracy as a second-order method on a much coarser mesh. One way to examine the efficiency of a CFD method is to compare the computational cost given the same error threshold. The benefit of using a high-order method increases as the level of required accuracy becomes more stringent. In practice, high-order methods may still exhibit a lower convergence rate when singularities are present in the solutions. In such cases, high-order methods may still be more efficient than second-order methods on a given grid thanks to lower numerical errors. In the numerical solution of the Euler equations, it has been shown that high-order spatial discretization produces lower numerical error at the same level of computational cost [3]. High-order methods have also been shown to provide significant efficiency benefits in the numerical solution of the Reynolds-averaged Navier-Stokes (RANS) equations [4].

The in-house flow solver Diablo with second-order SBP-SAT spatial discretization has been shown to provide accurate numerical solutions in the computation of high Reynolds number turbulent flows over aerodynamic geometries in previous work [5]. In this work, high-order SBP-SAT spatial discretization is implemented in the RANS code. This thesis project involves four stages. The initial stage of the project is to verify the current flow solver with the use of high order methods for inviscid and viscous flows. The second stage is to implement the high-order SBP operators in the RANS code. After the implementation, the codes and the solutions obtained with RANS with the higher-order spatial discretization are verified using benchmark test cases. Grid convergence studies are conducted to examine the efficiency of the high-order methods compared to the second-order method in solving the RANS equations. The last stage of this project is to identify and address the issues that arise when high-order methods are used to solve RANS equations. In an effort to deal with convergence issues, different dissipation models and the negative turbulence model are introduced. The algorithm is also tuned to provide fast and accurate numerical solutions. This thesis mainly focuses on the implementation of the third- and fourth-order methods and the study of the efficiency of the high-order methods.

1.3 Grid Refinement Studies

The verification process involves two stages: code verification and solution verification [6]. The error associated with the simulation model arises from two main sources: physical modelling er-

rors and numerical errors [7]. Numerical errors are related to the choice of spatial discretization methods, time marching methods, etc. Grid refinement studies are carried out in this thesis to estimate numerical errors and to examine the efficiency of the high-order methods. The order of the discretization error in a CFD simulation can be determined via a grid refinement study. The error in the simulated solution approaches zero when the grid is infinitely refined. This can be expressed as:

$$f_{exact} = f(h) + Ch^p + H.O.T$$

where f is a functional, such as the lift coefficient C_L or the drag coefficient C_D, C is a constant, h is a measure of grid spacing, p is the order of convergence, and H.O.T is short for higher order terms. With a fine mesh of grid spacing h, a medium mesh of grid spacing $2h$, and a coarse mesh of grid spacing $4h$, one can estimate the order of the discretization method, p, from:

$$p \approx \frac{\ln \frac{f_c - f_m}{f_m - f_f}}{\ln 2}$$

where f_c, f_m, and f_f are obtained with the coarse, medium, and fine meshes, respectively. The exact solution, f_{exact}, can be estimated as:

$$f_{exact} \approx f_f + \frac{f_f - f_m}{2^p - 1}$$

To verify the order of numerical accuracy of high-order methods, grid refinement studies have been employed to assess whether the expected order of discretization is achieved and whether the implementation of high-order methods increases the overall efficiency of the numerical simulation [4, 8]. Another approach is the use of manufactured solutions. The method of manufactured solutions involves the exercise of manufacturing an analytic solution to the governing equations. The numerical accuracy is assessed by comparing the numerical and analytical results at discrete nodes.

1.4 Literature Review

For a second-order accurate method, the numerical error is approximately proportional to h^2 in the asymptotic range, where h represents the mesh size Δx. For a p-th order method, the resulting numerical error is then proportional to h^p. If the mesh size h is reduced to $\frac{1}{2}h$, the error will be reduced by a factor of 2^p with a pth-order scheme, if the theoretical order is achieved. The use of high-order methods can be beneficial, as they provide more accurate results and potentially reduce

the computational cost to achieve a given level of accuracy.

For second-order algorithms, extremely high grid resolution is required for accurate simulations of vortical flow features because vorticity can be prematurely dissipated through numerical diffusion [9], leading to high computational cost. Thus, the use of high-order methods can be beneficial to simulate vorticity dominated flows. In the study of numerical simulations of wingtip vortices in the near field carried out by Churchfield and Blaisdell [10], high-order numerical schemes are applied to RANS equations and turbulence models. It is found that the use of high-order schemes significantly improves the prediction of the wing-tip vortex. In the simulation of leading edge vortices over sharp-edged delta wings, the benefits of using high-order methods is also recognized [11]. It is shown that high-order spatial discretization yields more accurate results and requires lower grid resolution.

The discontinuous Galerkin (DG) method is a popular approach for obtaining high-order accuracy. Nguyen, Persson and Peraire [12] numerically solved the RANS equations with the Spalart-Allmaras (SA) one-equation turbulence model using high-order discontinuous Galerkin approximations. They found abrupt changes in the eddy viscosity profile near the boundaries when the grids are coarse. An artificial viscosity term is therefore introduced to stabilize the RANS solutions where the grids are too coarse. In their work, turbulent flows over a flat plate and a NACA0012 airfoil were considered. The results showed that the numerical errors in drag decrease rapidly with increasing order of approximation on a given mesh, and thus showed the potential for lower computational cost. Morgan et al.[13] applied fourth- and sixth-order compact-difference approximations to a k-ϵ turbulence model. The high-order implementation was validated with a flat plate case, and the high-order solutions were also compared with second-order solutions in the case of wall-mounted hump. The results demonstrated that less grid resolution is required when using a high-order discretization of the k-ϵ turbulence equations.

The challenges when applying high-order methods to RANS equations are to make them robust and truly high-order. High-order methods for the RANS equations can be difficult to converge to the steady-state because of the numerical stiffness of the turbulence closure model. The abrupt changes of the turbulence working variables at the interface of boundary layers can lead to negative transient values of the turbulence working variables [14]. A modified version of the SA turbulence model presented by Moro et al.[15] has shown improved convergence properties of high-order computations on grids where the boundary-layer edge is under-resolved. A negative SA model is formulated by Allmaras et al.[16] to address the issues with under-resolved grids and negative transient values, with details to be described in Chapter 2. Meanwhile, the theoretical order of accuracy may not be obtained because of under-resolved solutions at the edge of a boundary layer [17]. Meanwhile, the turbulent/non-turbulent interface can impact the observed order of accuracy of the discretization.

A study carried out by Oliver and Darmofal [17] on the impact of solution irregularity of the SA turbulence model on high-order DG methods shows that the optimal order of accuracy for turbulent flows is not achieved while the laminar case attains optimal order of accuracy without the effects of the turbulence model. The boundary layer profile taken from the the upper surface of the airfoil at 70% chord shows a fluctuating turbulence model working variable and its first derivative, indicating a non-smooth solution at the boundary-layer edge. Error analysis based on the computed drag shows that local drag error is largest at the trailing edge while the boundary-layer edge region also plays a significant role in contributing to the total numerical error.

Another challenge of applying high-order methods to the RANS equations is the increasing computational cost due to the computation of nearest wall distance, d, as higher-order of accuracy of d is required to ensure its smoothness in higher-order methods. To reduce this cost, d is sometimes calculated along grid lines or computed as the distance from a field point to the nearest wall point, instead of taking the normal distance to the surface. In grids with multiple blocks, the nearest wall distance may be associated with another block. However, in order to minimize interblock communication, the nearest wall distance sometimes is determined inside each block. These approaches may lead to inexact or non-smooth d distributions and thus impact accuracy and convergence [18]. Even if the nearest wall distance is calculated as the normal distance to the wall surface, some issues still exist. For example, some turbulence models like the Spalart-Allmaras one-equation model and the SST k-ω two-equation model are formulated in such way that they are dependent on the distance d to the nearest wall to calculate the production or destruction terms with constants involved in these terms determined empirically on geometries with relatively flat surfaces. As a result, the calculation of the normal wall distance does not account for the wall features, such as convex for concave surfaces. Using some distance function $\mathcal{D}$ can help alleviate the above issues for turbulence models with d approximated in this form. Tucker et al [18] compared three approaches to approximate d based on differential equations: implicit Eikonal [19], implicit Hamilton-Jacobi, and the Poisson method [20]. It is found that the implicit d equations are relatively easy to implement and have robust convergence. In comparison, the Poisson method is the easiest to implement and is comparably accurate to the Eikonal approach. Fares and Schröder present a new distance function $\mathcal{D}$ to replace the regular normal wall distance to enhance the performance of turbulence models on geometries with strong curved surfaces and sharp edges [21]. Results show that the distance function reduces computational and communication cost compared to direct computation and can be easily implemented for the Spalart-Allmaras and the SST k-ω turbulence models.

The first International Workshop on High-order CFD Methods was held in 2012. In the review summarized by Wang et al.[2], the major findings from the workshop are described. In the case of inviscid flow through a channel with a bump, results for entropy error convergence from different

groups using second- and fourth-order methods are compared. Results obtained with a second-order finite-volume method are used as a reference. It is shown that the high-order method is more efficient than the reference second-order method. The asymptotic high-order convergence rates are attained in this case. In the simulation of flows over the NACA0012 airfoil, three flow conditions are considered: inviscid subsonic, inviscid transonic, and viscous subsonic. Compared to the reference solution obtained using a second-order finite-volume method, high-order solutions are more accurate with the same level of cost. The asymptotic convergence rates for the fourth-order methods are not achieved due to the presence of a singularity at the trailing edge. Both cases see a smaller spread of high-order results among groups than second-order results. In the case of a turbulent transonic flow over an RAE airfoil with a shock, two groups solved the RANS equations with the SA turbulence model while the other group used the explicit algebraic Reynolds-Stress model and the k-ω model. The results show that fourth-order methods generally generate more accurate numerical solutions than second-order methods given the same mesh or the same level of work units. However, there is a large spread of results among groups and no improvement is seen using high-order methods under mesh refinement.

1.5 Summary of Objectives

The main objective of this thesis project is to apply high-order methods to the RANS-SA model. High-order SBP operators are used in spatial discretization with SAT terms to impose boundary conditions. Two-dimensional test cases are carried out for error analysis to examine the efficiency of the high-order methods. A high-order dissipation model for the SA model is also implemented and examined to see how it works compared to the first-order upwinding scheme.

1.6 Thesis Outline

Starting in Chapter 2, the governing equations – the compressible Reynolds-averaged Navier-Stokes equations and the Spalart-Allmaras one-equation turbulence model – are introduced. Chapter 3 includes some background theory about the use of summation-by-parts operators and simultaneous approximation terms in spatial discretization. High-order methods are described in this chapter with emphasis on the spatial discretization of the corresponding terms in the turbulence model. Test cases that are used to verify the code are discussed in Chapter 4. Chapter 5 brings the thesis to a conclusion, summarizing the findings of the current work and also presenting a brief discussion on the limitations.

Chapter 2

Governing Equations

This chapter describes the governing equations that are used to model turbulent flows in this work. They are the Navier-Stokes equations and the one-equation Spalart-Allmaras turbulence model along with their boundary and initial conditions. The negative turbulence model to deal with negative transient states is also discussed in this chapter. Nondimensionalization and coordinate transformation techniques will be included in the end of this chapter.

2.1 Navier-Stokes Equations

The Navier-Stokes equations are coupled nonlinear PDEs that describe the conservation of mass, momentum, and energy for viscous flows. The compressible Navier-Stokes equations in three dimensional Cartesian coordinates can be written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \frac{\partial \mathbf{E_v}}{\partial x} + \frac{\partial \mathbf{F_v}}{\partial y} + \frac{\partial \mathbf{G_v}}{\partial z} \tag{2.1}$$

where the variables are the fluid density, ρ, velocity components in Cartesian coordinates, $\langle u, v, w \rangle$, and the total energy per unit volume, e:

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}$$

The total energy, e includes internal energy per unit volume and kinetic energy per unit volume:

$$e = \rho\epsilon + \frac{1}{2}\rho(u^2 + v^2 + w^2)$$

where $\epsilon = \frac{R}{\gamma - 1}T$, and T is the temperature. The inviscid flux vectors are given as

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

and the viscous flux vectors are

$$\mathbf{E_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ E_{v,5} \end{bmatrix} \qquad \mathbf{F_v} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ F_{v,5} \end{bmatrix} \qquad \mathbf{G_v} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ G_{v,5} \end{bmatrix}$$

where

$$\begin{aligned}
\tau_{xx} &= \frac{4}{3}(\mu + \mu_t)u_x - \frac{2}{3}(\mu + \mu_t)(v_y + w_z) \\
\tau_{xy} &= (\mu + \mu_t)(u_y + v_x) \\
\tau_{xz} &= (\mu + \mu_t)(u_z + w_x) \\
\tau_{yx} &= \tau_{xy} \\
\tau_{yy} &= \frac{4}{3}v_y - \frac{2}{3}(\mu + \mu_t)(u_x + w_z) \\
\tau_{yz} &= (\mu + \mu_t)(v_x + w_y) \\
\tau_{zx} &= \tau_{xz} \\
\tau_{zy} &= \tau_{yz} \\
\tau_{zz} &= \frac{4}{3}(\mu + \mu_t)w_z - \frac{2}{3}(\mu + \mu_t)(u_x + v_y)
\end{aligned}$$

and

$$E_{v,5} = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + (\mu Pr^{-1} + \mu_t Pr_t^{-1})(\gamma - 1)^{-1}\partial_x a^2$$
$$F_{v,5} = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + (\mu Pr^{-1} + \mu_t Pr_t^{-1})(\gamma - 1)^{-1}\partial_y a^2$$
$$G_{v,5} = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + (\mu Pr^{-1} + \mu_t Pr_t^{-1})(\gamma - 1)^{-1}\partial_z a^2$$

where μ is the coefficient of viscosity and μ_t is the turbulent eddy viscosity; γ is the specific heat ratio and is set to 1.4 for air. The laminar and turbulent Prandtl numbers are $Pr = 0.72$ and $Pr_t = 0.90$. Dynamic viscosity is calculated as a function of temperature using Sutherland's law:

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_\infty + S_t}{T + S_t}$$

The Sutherland constant, S_t, is $198.6R$. T_∞ and μ_∞ represent the free-stream temperature and dynamic viscosity, respectively.

These equations are also supplemented with the equation of state for an ideal gas to compute pressure, p:

$$p = \rho RT$$

Thus, pressure is computed from

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

2.2 Turbulence Model

The turbulence model being used in the flow solver is the one-equation Spalart-Allmaras turbulence model [22]. It is a linear eddy viscosity model based on the Boussinesq assumption with the last term ignored:

$$\tau_{ij} = 2\mu_t(S^*_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}) - \frac{2}{3}\rho k\delta_{ij}$$

where S^* represents the strain rate tensor. A turbulent kinematic viscosity, ν_t, is introduced to solve the turbulent flow and it is related to $\tilde{\nu}$ by:

$$\nu_t = f_{v1}\tilde{\nu}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$

and

$$\chi = \frac{\tilde{\nu}}{\nu}, \qquad c_{v1} = 7.1$$

The one-equation turbulence model solves the turbulent quantity $\tilde{\nu}$ in the following equation:

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + u_i \frac{\partial \tilde{\nu}}{\partial x_i} =& \frac{c_{b1}}{Re}(1 - f_{t2})\tilde{S}\tilde{\nu} + \frac{1 + c_{b2}}{\sigma Re}\nabla \cdot [(\nu + \tilde{\nu})\nabla\tilde{\nu}] \\ & - \frac{c_{b2}}{\sigma Re}[(\nu + \tilde{\nu})\nabla^2\tilde{\nu}] - \frac{1}{Re}(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2})(\frac{\tilde{\nu}}{d})^2 + \frac{1}{Re}f_{t1}\Delta U^2 \end{aligned} \tag{2.2}$$

The term $u_i \frac{\partial \tilde{\nu}}{\partial x_i}$ in equation (2.2) accounts for advection. The first term on the right-hand side of the equation, $\frac{c_{b1}}{Re}(1 - f_{t2})\tilde{S}\tilde{\nu}$, represents production. The variable $\tilde{S}$ is calculated based on the magnitude of the vorticity S:

$$\tilde{S} = Re(S + \bar{S})$$

where

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

$$\bar{S} = \frac{\tilde{\nu} f_{v2}}{\kappa^2 d^2}$$

The magnitude of vorticity, S is calculated as

$$S = \left[(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z})^2 + (\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x})^2 + (\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})^2 \right]^{1/2}$$

and d is the distance to wall surface. The remaining constants in this term are defined as:

$$c_{b1} = 0.1355 \qquad \kappa = 0.41$$

The second and third terms on the right-hand side account for diffusion. The constants are given as:

$$c_{b2} = 0.622 \qquad \sigma = \frac{2}{3}$$

The fourth term, $\frac{1}{Re}(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2})(\frac{\tilde{\nu}}{d})^2$, represents destruction. The functions f_w and f_{t2} are

given by

$$f_w = g[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}]^{\frac{1}{6}}$$
$$f_{t2} = c_{t3}\exp(-c_{t4}\chi^2)$$

where

$$g = r + c_{w2}(r^6 - r) \qquad r = \min(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10)$$
$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1}{\sigma}(1+c_{b2}),$$
$$c_{w2} = 0.3 \qquad c_{w3} = 2.0$$
$$c_{t3} = 1.2 \qquad c_{t4} = 0.5$$

The last term on the right-hand side of equation (2.2) specifies an explicit laminar to turbulent transition location. If the trip point is not specified and a fully turbulent flow is assumed, this term is set to zero. The function f_{t1} is defined as

$$f_{t1} = c_{t1} g_t \exp\big[-c_{t2}\frac{w_t^2}{\Delta U^2}(d^2 + g_t^2 d_t^2)\big]$$
$$g_t = \min\big[0.1, \frac{\Delta U}{\omega_t \Delta x_t}\big]$$

where ΔU is the difference between the velocity at the field point and that at the trip point, Δx_t is the grid spacing at the trip, ω_t is the vorticity at the trip, and d_t is the distance to the trip. The constants are given as

$$c_{t1} = 5.0 \qquad c_{t2} = 2.0$$

2.2.1 Negative Turbulence Model

The negative model was developed in order to address non-physical values during transient states when solving the flow discretely [16]. First, modifications are made to prevent negative modified vorticity, $\tilde{S}$. $\tilde{S}$ is a function of f_{v2}, which can become negative over a range of χ. In the negative turbulence model, when $\bar{S}$ is smaller than $-0.7S$, $\tilde{S}$ is computed in the following form:

$$\tilde{S} = Re[S + \frac{S(c_{v2}^2 S + C_{v3}\bar{S})}{(c_{v3} - 2c_{v2})S - \bar{S}}] \qquad \text{if} \quad \bar{S} < -c_{v2}S$$

where $c_{v2} = 0.7$ and $c_{v3} = 0.9$.

In the presence of negative $\tilde{\nu}$, $\tilde{\nu}$ is set to zero. The negative S-A model is in the following form:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_i \frac{\partial \tilde{\nu}}{\partial x_i} = P_n - D_n + \frac{1}{\sigma Re} \nabla \cdot [(\nu + \tilde{\nu} f_n) \nabla \tilde{\nu}] + \frac{c_{b2}}{\sigma Re} (\nabla \tilde{\nu})^2$$

where P_n and D_n represent production and wall destruction respectively. To ensure stability, two constraints must be satisfied. First, it is required that

$$P_n - D_n \geq 0$$

The production term is defined to be positive while the destruction term negative:

$$P_n = \frac{c_{b1}}{Re}(1 - c_{t3}) S \tilde{\nu} \qquad D_n = -\frac{c_{w1}}{Re} \left[\frac{\tilde{\nu}}{d}\right]^2$$

In the negative model, the production term is a function of the magnitude of vorticity, S, instead of the modified value, $\tilde{S}$.

The second constraint is that the diffusion coefficient must be positive for negative turbulence eddy viscosity; therefore, f_n must satisfy

$$1 + \chi f_n \geq 0$$

Thus, in the negative model, f_n is formulated as

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}$$

where $c_{n1} = 16$ to ensure that the diffusion coefficient, $\nu + f_n \tilde{\nu}$, is always positive.

2.3 Nondimensionalizaion

In the compressible Navier Stokes equations, the conservative variables are non-dimensionalized with reference quantities (the non-dimensional quantities are indicated by the bar):

$$\bar{t} = \frac{ta}{c} \qquad \bar{\rho} = \frac{\rho}{\rho_\infty} \qquad \bar{e} = \frac{e}{\rho_\infty a_\infty^2}$$
$$\bar{x} = \frac{x}{c} \qquad \bar{y} = \frac{y}{c} \qquad \bar{z} = \frac{z}{c}$$
$$\bar{u} = \frac{u}{a_\infty} \qquad \bar{v} = \frac{v}{a_\infty} \qquad \bar{w} = \frac{w}{a_\infty}$$

where the subscript ∞ represents free-stream values. Substituting these quantities, the Reynolds number is thus defined as

$$Re = \frac{\rho_\infty a_\infty c}{\mu_\infty}$$

where c is a reference length, and the speed of sound, a, is calculated using the ideal gas law, $a = \sqrt{\gamma p/\rho}$. Dynamic viscosity and temperature are nondimensionalized with the free-stream values.

$$\bar{\mu} = \frac{\mu}{\mu_\infty} \qquad \bar{T} = \frac{T}{T_\infty}$$

Therefore, $\bar{\mu}$ is calculated as

$$\bar{\mu} = \bar{a}^3 \left(\frac{1 + S_t/T_\infty}{\bar{T} + S_t/T_\infty} \right)$$

In the turbulence model, the turbulent quantity, $\tilde{\nu}$ is normalized with $\frac{\mu_\infty}{\rho_\infty}$.

2.4 Coordinate Transformation

To apply finite-difference approximations to spatial derivatives, the physical space in the Cartesian coordinates is transformed into a uniform computational grid with $\Delta\xi = \Delta\eta = \Delta\zeta$. This section will describe the coordinate transformation for the turbulence model. More information about the transformation of the mean-flow equations can be found in [23]. In the turbulence model, the derivatives with respect to the Cartesian coordinates need to be replaced with derivatives with respect to the computational coordinates. After the coordinate transformation, the advective term is in the form of

$$U_i \frac{\partial \tilde{\nu}}{\partial \xi_i} = U \frac{\partial \tilde{\nu}}{\partial \xi} + V \frac{\partial \tilde{\nu}}{\partial \eta} + W \frac{\partial \tilde{\nu}}{\partial \zeta}$$

where $\langle U, V, W \rangle$ are the contravariant velocities defined as

$$U = u\xi_x + v\xi_y + w\xi_z$$
$$V = u\eta_x + v\eta_y + w\eta_z$$
$$W = u\zeta_x + v\zeta_y + w\zeta_z$$

In the coordinate transformation of the diffusive terms, the cross-derivative terms are dropped to

be consistent with the thin layer approximation. The resulting terms are given by

$$\begin{aligned}\nabla \cdot [(\nu+\tilde{\nu})\nabla\tilde{\nu}] \approx\ & \xi_x\partial_\xi[(\nu+\tilde{\nu})\xi_x\partial_\xi\tilde{\nu}] + \xi_y\partial_\xi[(\nu+\tilde{\nu})\xi_y\partial_\xi\tilde{\nu}] + \xi_z\partial_\xi[(\nu+\tilde{\nu})\xi_z\partial_\xi\tilde{\nu}] \\ & + \eta_x\partial_\eta[(\nu+\tilde{\nu})\eta_x\partial_\eta\tilde{\nu}] + \eta_y\partial_\eta[(\nu+\tilde{\nu})\eta_y\partial_\eta\tilde{\nu}] + \eta_z\partial_\eta[(\nu+\tilde{\nu})\eta_z\partial_\eta\tilde{\nu}] \\ & + \zeta_x\partial_\zeta[(\nu+\tilde{\nu})\zeta_x\partial_\zeta\tilde{\nu}] + \zeta_y\partial_\zeta[(\nu+\tilde{\nu})\zeta_y\partial_\zeta\tilde{\nu}] + \zeta_z\partial_\zeta[(\nu+\tilde{\nu})\zeta_z\partial_\zeta\tilde{\nu}]\end{aligned}$$

$$\begin{aligned}\nabla^2\tilde{\nu} \approx\ & \xi_x\partial_\xi(\xi_x\partial_\xi\tilde{\nu}) + \xi_y\partial_\xi(\xi_y\partial_\xi\tilde{\nu}) + \xi_z\partial_\xi(\xi_z\partial_\xi\tilde{\nu}) \\ & + \eta_x\partial_\eta(\eta_x\partial_\eta\tilde{\nu}) + \eta_y\partial_\eta(\eta_y\partial_\eta\tilde{\nu}) + \eta_z\partial_\eta(\eta_z\partial_\eta\tilde{\nu}) \\ & + \zeta_x\partial_\zeta(\zeta_x\partial_\zeta\tilde{\nu}) + \zeta_y\partial_\zeta(\zeta_y\partial_\zeta\tilde{\nu}) + \zeta_z\partial_\zeta(\zeta_z\partial_\zeta\tilde{\nu})\end{aligned}$$

Meanwhile, the vorticity magnitude, S, is defined as

$$S = (S_1^2 + S_2^2 + S_3^2)^{\frac{1}{2}}$$

with S_1, S_2 and S_3 components transformed into the following form

$$\begin{aligned}S_1 &= (\xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\eta) - (\xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta) \\ S_2 &= (\xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\eta) - (\xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta) \\ S_3 &= (\xi_x v_\xi + \eta_y v_\eta + \zeta_x v_\eta) - (\xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta)\end{aligned}$$

2.5 Boundary Conditions

2.5.1 Farfield Boundary

For the turbulence model, for a fully turbulent flow, the farfield value of $\tilde{\nu}$ is set to 3.0. If the laminar-to-turbulent transition region is explicitly specified, the value of $\tilde{\nu}$ is set to 0.1.

2.5.2 Wall Surface

The enforcement of the boundary conditions for a solid surface is designed such that it serves for both slip-wall and non-slip conditions. For inviscid flows, the velocity of the flow normal to the wall is required to be zero. For viscous flows, it is also required that the flow velocity at the surface is zero.

The value of $\tilde{\nu}$ on any solid surface is set to zero. In addition, an adiabatic condition is assumed,

which requires zero temperature gradient normal to the solid surface.

2.5.3 Symmetry Boundary

For the mean flow, the gradients of all flow variables are zero normal to the symmetry plane. Similarly, at the symmetry boundary, the gradient of $\tilde{\nu}$ is zero normal to the boundary:

$$\frac{\partial \tilde{\nu}}{\partial n} = 0$$

2.6 Initial Conditions

The initial values of the mean flow are set to the values at the farfield boundary. Similarly, if the flow is assumed fully turbulent, the initial value of $\tilde{\nu}$ is set to 3.0. If the transition region is explicitly specified, the initial values of $\tilde{\nu}$ is set to 0.1 upstream of the transition region to ensure low turbulence for the laminar flow. For nodes downstream of the transition region, $\tilde{\nu}$ is set to 10 at the initial state.

Chapter 3

Spatial Discretization

The in-house flow solver uses structured multi-block grids which can fit complex geometries and allows for parallelization. The SBP-SAT method is applied for spatial discretization in solving turbulent flows. An SBP operator is constructed to satisfy a discrete summation by parts property and is proven to be stable with the use of SATs [24, 25]. SATs correct the solution values at boundary or interface nodes by adding a penalty term. The use of SATs imposes weak boundary terms without destroying the SBP property of the difference operator. Dias [26] has applied the high-order SBP-SAT approach to the Euler equations. High-order viscous terms have become available in the work of applying the high-order spatial discretization to aerodynamic optimization of laminar flows [27]. Now the work is extended to turbulent flows. This chapter will mainly focus on the application of SBP operators and SATs to the turbulence model while the details on the mean flow spatial discretization can be found in the work mentioned above.

3.1 Background Theory

3.1.1 Summation-by-Parts Operators for a First Derivative

The difference operator for a first derivative, $D_1 = H^{-1}\theta$, is an SBP operator if it satisfies the following three properties [28, 29]:

1. $H\mathbf{u_x} = \theta\mathbf{u}, \quad H\mathbf{v_x} = \theta\mathbf{v} + H\mathbf{T}$ where $\mathbf{u}$ is the discrete solution vector, $\mathbf{v}$ is the exact solution at the nodes, and $\mathbf{T}$ is the truncation error. For a method of order q, the truncation error is of order $O(\triangle x^q)$.

2. H is a symmetric positive definite matrix.

3. $\theta + \theta^T = E$, where

$$E = \begin{bmatrix} -1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1 \end{bmatrix}$$

An SBP operator is constructed to satisfy a discrete summation by parts property and is proven to be stable without boundary conditions. To form a well-posed problem, boundary conditions are required for a partial-differential-equation (PDE) system. One approach to impose weak boundary conditions without destroying the SBP property of the difference operator is the use of SATs, which correct the values of boundary or interface nodes to the desired values by adding a penalty term. The penalty parameters are selected so that both stability and conservation are guaranteed. With the implementation of SATs, boundary values at two neighbouring interfaces are not exactly the same. The flow can be solved with the information within the grid block and from the adjacent block, thus minimal information is passed from one block to another.

Diagonal-norm SBP operators are constructed with a diagonal H matrix in the form of

$$\begin{bmatrix} h_{1,1} & & & & & & \\ & h_{2,2} & & & & & \\ & & \ddots & & & & \\ & & & h_{2p,2p} & & & \\ & & & & 1 & & \\ & & & & & \ddots & \end{bmatrix}$$

with all $h_{i,i}$ being positive [24], where p is a parameter. Diagonal-norm SBP operators have an interior accuracy of $2p$. The order of accuracy of boundary or interface nodes is p, thus limiting the global accuracy to the order of $p + 1$.

For example, a diagonal-norm SBP operator constructed with $p = 1$ results in first-order accuracy at the boundary nodes and second-order accuracy at the interior nodes. The global order of accuracy

is 2. The H matrix is given as

$$H = h \begin{bmatrix} 1/2 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & 1/2 & \end{bmatrix}$$

The θ matrix is given as

$$\theta = \begin{bmatrix} -1/2 & 1/2 & & & & \\ -1/2 & 0 & 1/2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1/2 & 0 & 1/2 & \\ & & & -1/2 & 1/2 & \end{bmatrix}$$

The lower portion of H is symmetric to the upper portion about the antidiagonal whereas the lower portion of θ is antisymmetric to the upper 4×6 block. When p is larger than 2, the entries of the H and θ matrices are determined with free parameters. The free parameters can be determined by minimizing the truncation error.

Block-norm SBP operators are constructed with a block-diagonal H matrix in the form of

$$\begin{bmatrix} \hat{H} & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & \hat{H} \end{bmatrix} \text{ with } \hat{H} \text{ in the form of } \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,2p} \\ h_{1,2} & h_{2,2} & \dots & h_{2,2p} \\ \vdots & \vdots & \vdots & \vdots \\ h_{1,2p} & h_{2,2p} & \dots & h_{2p,2p} \end{bmatrix}$$

and I is an identity matrix [24].

It is shown by Strand [24] that using block-norm SBP operators will increase the accuracy of boundary nodes to the order of $2p - 1$, thus increasing the global accuracy to $2p$. On the other hand, block-norm SBP operators have the disadvantage of not being provably stable when used in

curvilinear coordinates.

3.1.2 Artificial Dissipation

The dissipation operator is designed to be compatible with the SBP derivative operator so that it preserves the stability properties of the numerical approximation. The artificial dissipation operator first constructed by Mattsson, Svärd, and Nordström [30] of $2p$ interior order is given by

$$A_D = -\tilde{H}^{-1}\tilde{D}_p^T B \tilde{D}_p$$

The tilde sign again indicates that the matrices are undivided difference operators:

$$\frac{1}{h}\tilde{H} = H, \qquad \frac{1}{h^p}\tilde{D}_p = D_p$$

where H is the $2p$ order diagonal norm, D_p is a consistent approximation of d^p/dx^p, and B is a positive diagonal matrix $B = \mathrm{diag}(0, c, ...c)$.

For example, the dissipation operator A_2 with positive constant c that preserves second-order accuracy is given as

$$\tilde{D}_1 = \begin{bmatrix} -1 & 1 & 0 & & & \\ -1 & 1 & 0 & & & \\ 0 & -1 & 1 & & & \\ & & & \ddots & & \\ & & & -1 & 1 & 0 \\ & & & 0 & -1 & 1 \\ & & & 0 & -1 & 1 \end{bmatrix}, \qquad A_2 = c \begin{bmatrix} -2 & 2 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -2 & 1 & \\ & & & -2 & 2 & \end{bmatrix}.$$

High-order finite difference SBP operators are optimized to minimize the bandwidth and reduce truncation errors [31]. Meanwhile, high-order SBP dissipation operators are developed with the optimized derivative SBP operators.

3.1.3 Summation-by-Parts Operators for a Second Derivative

A difference operator for a second derivative can be constructed by applying the difference operator for a first derivative twice, known as a non-compact operator or a wide-stencil operator. This results

in a wider interior stencil and, as result, higher computational cost. A minimum-stencil operator for a second derivative can be derived with constant or variable coefficients. The second-derivative SBP operator with coefficient β is constructed as [25, 32]:

$$\begin{aligned} D_2(B) =& H^{-1}[-D_1^T HBD_1 + EBD_b] \\ & - \frac{1}{h} H^{-1} \sum_{i=p+1}^{2p} \alpha_i ((\tilde{D}_i)^T C_i B \tilde{D}_i \end{aligned}$$

where B is the diagonal matrix containing the variable coefficient β, and C_i is a diagonal matrix in the form of $\text{diag}(c_1, \ldots, c_{2p}, \ldots, c_{2p}, c_1)$. D_b is an approximation to the first derivative at a boundary. $\tilde{D}_i$ contains the repeated coefficients for interior nodes. The tilde sign indicates that the operator is an undivided difference operator. For second-order, third-order and fourth-order methods, the α coefficients are

- second-order method: $p = 1, \alpha_2 = \frac{1}{4}$
- third-order method: $p = 2, \alpha_3 = \frac{1}{18}, \alpha_4 = \frac{1}{48}$
- fourth-order method: $p = 3, \alpha_4 = \frac{1}{80}, \alpha_5 = \frac{1}{100}, \alpha_6 = \frac{1}{720}$

For example, the second-order operator ($p = 1$) is given as [33]

$$D_2(\beta) = H^{-1} \Big[- D_1^T HBD_1 - \frac{1}{4h} \tilde{D}_2^T C_2 B \tilde{D}_2 + EBD_1^{(2)} \Big],$$

where

$$\tilde{D}_2 = \begin{bmatrix} 1 & -2 & 1 & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix}, \quad EBD_1^{(2)} = \frac{1}{h} \begin{bmatrix} \frac{3\beta_1}{2} & -2\beta_1 & \frac{\beta_1}{2} & & & \\ 0 & 0 & 0 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 0 & 0 & 0 & \\ & & \frac{\beta_N}{2} & -2\beta_N & \frac{3\beta_N}{2} & \end{bmatrix}$$

and D_1 represents the corresponding SBP operator for the first derivative as described above.

3.1.4 Simultaneous Approximation Terms

Multiblock grids can be used with the flow solver which allows parallel computations on multiple processors and fits complicated geometries. Penalty terms can be used to impose weak boundary and interface conditions while preserving the SBP property [34]. The simultaneous approximation terms are used in this work coupled with SBP operators. The derivation of SATs for the Navier-Stokes equations can be found in [35, 36, 37]. The SATs are added to the residual vector in the form of $\alpha_{SAT}(S - S_{targ})$, where α_{SAT} represents a penalty parameter chosen for stability, S is based on flow solutions and the $targ$ subscript indicates the targeted values at the boundaries. For the Euler equations, the SAT on the low side is added as [38]:

$$SAT_{inv} = -H_b^{-1} J^{-1} A_\xi^+ (\mathbf{Q} - \mathbf{Q_{targ}})$$

where H_b contains the boundary node values of the diagonal norm matrix H, and

$$A_\xi^+ = \frac{A_\xi + X^{-1}|\Lambda|X}{2},$$

$$A_\xi = \frac{\partial \mathbf{E}}{\partial \mathbf{Q}}.$$

X is the right eigenmatrix of A_ξ, while Λ contains the eigenvalues along its diagonal. For the Navier-Stokes equations, the portion of SATs for the viscous fluxes in the ξ direction is in the form of:

$$SAT_{visc,flux} = \frac{H_b^{-1}\sigma^V}{Re}(\mathbf{E_v} - \mathbf{E_{v,targ}})$$

with σ^V set to 1 at the low side, -1 at the high side.

3.2 Turbulence Model

3.2.1 Advective Terms

The advection term in the turbulence model is given by

$$U_i \frac{\partial \tilde{\nu}}{\partial x_i}$$

where U_i represents the contravariant velocity. The diagonal-norm SBP operator for a first derivative is applied to spatially discretize this term. For second-order global order of accuracy, the following example shows the resulting discretization in the $\xi-$direction for the interior nodes:

$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_j \approx U_j(-\frac{1}{2}\tilde{\nu}_{j-1} + \frac{1}{2}\tilde{\nu}_{j+1}).$$

For the boundary or interface nodes, the resulting approximation is the following form:

$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_1 \approx U_1(-\tilde{\nu}_1 + \tilde{\nu}_2)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_N \approx U_N(-\tilde{\nu}_{N-1} + \tilde{\nu}_N)$$

For third-order global accuracy, the interior nodes have 4th-order accuracy while at the boundary nodes it is second-order accurate. To use this SBP operator, it is required that there are at least 9 nodes in each direction in one domain. The norm matrix, H, to construct the operator is given as

$$H = \begin{bmatrix} 17/48 & & & & & & \\ & 59/48 & & & & & \\ & & 43/48 & & & & \\ & & & 49/48 & & & \\ & & & & 1 & & \\ & & & & & \ddots & \end{bmatrix}$$

The θ matrix is given as

$$\theta = \begin{bmatrix} -1/2 & 59/96 & -1/12 & -1/32 & & & & \\ -59/96 & 0 & 59/96 & 0 & & & & \\ 1/12 & -59/96 & 0 & 59/96 & -1/12 & & & \\ 1/32 & 0 & -59/96 & 0 & 2/3 & -1/12 & & \\ & & 1/12 & -2/3 & 0 & 2/3 & -1/12 & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

Thus, the resulting spatial discretization is as summarized below:

low side:

$$
\begin{aligned}
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_1 &\approx U_1(-\frac{24}{17}\tilde{\nu}_1+\frac{59}{34}\tilde{\nu}_2-\frac{4}{17}\tilde{\nu}_3-\frac{3}{34}\tilde{\nu}_4)\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_2 &\approx U_2(-\frac{1}{2}\tilde{\nu}_1+\frac{1}{2}\tilde{\nu}_3)\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_3 &\approx U_1(\frac{4}{43}\tilde{\nu}_1-\frac{59}{86}\tilde{\nu}_2+\frac{59}{86}\tilde{\nu}_4-\frac{4}{43}\tilde{\nu}_5)\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_4 &\approx U_1(\frac{3}{98}\tilde{\nu}_1-\frac{59}{98}\tilde{\nu}_3+\frac{32}{49}\tilde{\nu}_5-\frac{4}{49}\tilde{\nu}_6)
\end{aligned}
$$

interior nodes:

$$
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_j \approx U_j(\frac{1}{12}\tilde{\nu}_{j-2}-\frac{2}{3}\tilde{\nu}_{j-1}+\frac{2}{3}\tilde{\nu}_{j+1}-\frac{1}{12}\tilde{\nu}_{j+2})
$$

high side:

$$
\begin{aligned}
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_{N-3} &\approx U_{N-3}(-\frac{3}{98}\tilde{\nu}_N+\frac{59}{98}\tilde{\nu}_{N-3}-\frac{32}{49}\tilde{\nu}_{N-4}+\frac{4}{49}\tilde{\nu}_{N-5})\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_{N-2} &\approx U_{N-2}(\frac{4}{43}\tilde{\nu}_{N-4}-\frac{59}{86}\tilde{\nu}_{N-3}+\frac{59}{86}\tilde{\nu}_{N-1}-\frac{4}{43}\tilde{\nu}_N)\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_{N-1} &\approx U_{N-1}(-\frac{1}{2}\tilde{\nu}_{N-2}+\frac{1}{2}\tilde{\nu}_N)\\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_N &\approx U_N(\frac{24}{17}\tilde{\nu}_N-\frac{59}{34}\tilde{\nu}_{N-1}+\frac{4}{17}\tilde{\nu}_{N-2}+\frac{3}{34}\tilde{\nu}_{N-3})
\end{aligned}
$$

For fourth-order global accuracy, the interior nodes are sixth-order accurate and at least 13 nodes are required in each direction. The fourth-order accurate SBP operator for the first derivative involves a free parameter, Q_{15}. In this work, Q_{15} by default is set to 0.018464384393334 [27]. The norm matrix and the θ matrix for the fourth-order diagonal-norm SBP operator is included in the appendix and the resulting discretization are given by:

low side:

$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_1 \approx U_1(a_{11}\tilde{\nu}_1 + a_{12}\tilde{\nu}_2 + a_{13}\tilde{\nu}_3 + a_{14}\tilde{\nu}_4 + a_{15}\tilde{\nu}_5 + a_{16}\tilde{\nu}_6)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_2 \approx U_2(a_{21}\tilde{\nu}_1 + a_{22}\tilde{\nu}_2 + a_{23}\tilde{\nu}_3 + a_{24}\tilde{\nu}_4 + a_{25}\tilde{\nu}_5 + a_{26}\tilde{\nu}_6)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_3 \approx U_3(a_{31}\tilde{\nu}_1 + a_{32}\tilde{\nu}_2 + a_{33}\tilde{\nu}_3 + a_{34}\tilde{\nu}_4 + a_{35}\tilde{\nu}_5 + a_{36}\tilde{\nu}_6)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_4 \approx U_4(a_{41}\tilde{\nu}_1 + a_{42}\tilde{\nu}_2 + a_{43}\tilde{\nu}_3 + a_{44}\tilde{\nu}_4 + a_{45}\tilde{\nu}_5 + a_{46}\tilde{\nu}_6 + a_{47}\tilde{\nu}_7)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_5 \approx U_5(a_{51}\tilde{\nu}_1 + a_{52}\tilde{\nu}_2 + a_{53}\tilde{\nu}_3 + a_{54}\tilde{\nu}_4 + a_{55}\tilde{\nu}_5 + a_{56}\tilde{\nu}_6 + a_{57}\tilde{\nu}_7 + a_{58}\tilde{\nu}_8)$$
$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_6 \approx U_6(a_{61}\tilde{\nu}_1 + a_{62}\tilde{\nu}_2 + a_{63}\tilde{\nu}_3 + a_{64}\tilde{\nu}_4 + a_{65}\tilde{\nu}_5 + a_{66}\tilde{\nu}_6 + a_{67}\tilde{\nu}_7 + a_{68}\tilde{\nu}_8 + a_{69}\tilde{\nu}_69)$$

where

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{16} \end{bmatrix} = \begin{bmatrix} -\frac{21600}{13649} \\ \frac{224881}{109192} - \frac{10800}{13649}Q_{15} \\ -\frac{10073}{40947} + \frac{43200}{13649}Q_{15} \\ -\frac{16033}{54596} - \frac{64800}{13649}Q_{15} \\ \frac{43200}{13649}Q_{15} \\ \frac{20539}{327576} - \frac{10800}{13649}Q_{15} \end{bmatrix}, \qquad \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{25} \\ a_{26} \end{bmatrix} = \begin{bmatrix} -\frac{224881}{480520} + \frac{2160}{12013}Q_{15} \\ 0 \\ \frac{49967}{144156} - \frac{21600}{12013}Q_{15} \\ -\frac{1683}{12013} + \frac{43200}{12013}Q_{15} \\ \frac{1915}{96104} - \frac{32400}{12013}Q_{15} \\ -\frac{6964}{180195} + \frac{8640}{12013}Q_{15} \end{bmatrix}$$

$$\begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \\ a_{35} \\ a_{36} \end{bmatrix} = \begin{bmatrix} \frac{10073}{81330} - \frac{4320}{2711}Q_{15} \\ -\frac{49967}{65064} + \frac{10800}{2711}Q_{15} \\ 0 \\ \frac{28279}{32532} - \frac{21600}{2711}Q_{15} \\ -\frac{4651}{16266} + \frac{21600}{2711}Q_{15} \\ \frac{6591}{108440} - \frac{6480}{2711}Q_{15} \end{bmatrix}, \qquad \begin{bmatrix} a_{41} \\ a_{42} \\ a_{43} \\ a_{44} \\ a_{45} \\ a_{46} \\ a_{47} \end{bmatrix} = \begin{bmatrix} \frac{16033}{214360} + \frac{6480}{5359}Q_{15} \\ -\frac{1683}{10718} - \frac{21600}{5359}Q_{15} \\ -\frac{25157}{42872} - \frac{21600}{5359}Q_{15} \\ 0 \\ \frac{25157}{42872} - \frac{10800}{5359}Q_{15} \\ -\frac{547}{6990} + \frac{4320}{5359}Q_{15} \\ \frac{72}{5359} \end{bmatrix}$$

$$\begin{bmatrix} a_{51} \\ a_{52} \\ a_{53} \\ a_{54} \\ a_{55} \\ a_{56} \\ a_{57} \\ a_{58} \end{bmatrix} = \begin{bmatrix} -\frac{8460}{7877}Q_{15} \\ -\frac{1915}{63016} + \frac{32400}{7877}Q_{15} \\ \frac{4651}{23631} - \frac{43200}{7877}Q_{15} \\ -\frac{25157}{31508} + \frac{21600}{7877}Q_{15} \\ 0 \\ \frac{147127}{189048} - \frac{2160}{7877}Q_{15} \\ -\frac{1296}{7877} \\ \frac{144}{7877} \end{bmatrix}, \quad \begin{bmatrix} a_{61} \\ a_{62} \\ a_{63} \\ a_{64} \\ a_{65} \\ a_{66} \\ a_{67} \\ a_{68} \\ a_{69} \end{bmatrix} = \begin{bmatrix} -\frac{20539}{1051244} + \frac{10800}{43801}Q_{15} \\ \frac{6964}{131403} - \frac{43200}{43801}Q_{15} \\ -\frac{6591}{175204} + \frac{64800}{43801}Q_{15} \\ \frac{12581}{131403} - \frac{43200}{43801}Q_{15} \\ \frac{735635}{1051244} + \frac{10800}{43801}Q_{15} \\ 0 \\ \frac{32400}{43801} \\ -\frac{6480}{43801} \\ \frac{720}{43801} \end{bmatrix}$$

interior nodes:

$$[U\frac{\partial\tilde{\nu}}{\partial\xi}]_j \approx U_j(-\frac{1}{60}\tilde{\nu}_{j-3} + \frac{3}{20}\tilde{\nu}_{j-2} - \frac{3}{4}\tilde{\nu}_{j-1} + \frac{3}{4}\tilde{\nu}_{j+1} - \frac{3}{20}\tilde{\nu}_{j+2} + \frac{1}{60}\tilde{\nu}_{j+3})$$

high side:

$$\begin{aligned}
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_1 &\approx U_1(a_{11}\tilde{\nu}_1 + a_{12}\tilde{\nu}_2 + a_{13}\tilde{\nu}_3 + a_{14}\tilde{\nu}_4 + a_{15}\tilde{\nu}_5 + a_{16}\tilde{\nu}_6) \\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_2 &\approx U_2(a_{21}\tilde{\nu}_1 + a_{22}\tilde{\nu}_2 + a_{23}\tilde{\nu}_3 + a_{24}\tilde{\nu}_4 + a_{25}\tilde{\nu}_5 + a_{26}\tilde{\nu}_6) \\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_3 &\approx U_3(a_{31}\tilde{\nu}_1 + a_{32}\tilde{\nu}_2 + a_{33}\tilde{\nu}_3 + a_{34}\tilde{\nu}_4 + a_{35}\tilde{\nu}_5 + a_{36}\tilde{\nu}_6) \\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_4 &\approx U_4(a_{41}\tilde{\nu}_1 + a_{42}\tilde{\nu}_2 + a_{43}\tilde{\nu}_3 + a_{44}\tilde{\nu}_4 + a_{45}\tilde{\nu}_5 + a_{46}\tilde{\nu}_6 + a_{47}\tilde{\nu}_7) \\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_5 &\approx U_5(a_{51}\tilde{\nu}_1 + a_{52}\tilde{\nu}_2 + a_{53}\tilde{\nu}_3 + a_{54}\tilde{\nu}_4 + a_{55}\tilde{\nu}_5 + a_{56}\tilde{\nu}_6 + a_{57}\tilde{\nu}_7 + a_{58}\tilde{\nu}_8) \\
[U\frac{\partial\tilde{\nu}}{\partial\xi}]_6 &\approx U_6(a_{61}\tilde{\nu}_1 + a_{62}\tilde{\nu}_2 + a_{63}\tilde{\nu}_3 + a_{64}\tilde{\nu}_4 + a_{65}\tilde{\nu}_5 + a_{66}\tilde{\nu}_6 + a_{67}\tilde{\nu}_7 + a_{68}\tilde{\nu}_8 + a_{69}\tilde{\nu}_6 9)
\end{aligned}$$

where

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{16} \end{bmatrix} = \begin{bmatrix} -\frac{21600}{13649} \\ \frac{224881}{109192} - \frac{10800}{13649}Q_{15} \\ -\frac{10073}{40947} + \frac{43200}{13649}Q_{15} \\ -\frac{16033}{54596} - \frac{64800}{13649}Q_{15} \\ \frac{43200}{13649}Q_{15} \\ \frac{20539}{327576} - \frac{10800}{13649}Q_{15} \end{bmatrix}, \quad \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{25} \\ a_{26} \end{bmatrix} = \begin{bmatrix} -\frac{224881}{480520} + \frac{2160}{12013}Q_{15} \\ 0 \\ \frac{49967}{144156} - \frac{21600}{12013}Q_{15} \\ -\frac{1683}{12013} + \frac{43200}{12013}Q_{15} \\ \frac{1915}{96104} - \frac{32400}{12013}Q_{15} \\ -\frac{6964}{180195} + \frac{8640}{12013}Q_{15} \end{bmatrix}$$

$$\begin{bmatrix} a_{31} \\ a_{32} \\ a_{33} \\ a_{34} \\ a_{35} \\ a_{36} \end{bmatrix} = \begin{bmatrix} \frac{10073}{81330} - \frac{4320}{2711}Q_{15} \\ -\frac{49967}{65064} + \frac{10800}{2711}Q_{15} \\ 0 \\ \frac{28279}{32532} - \frac{21600}{2711}Q_{15} \\ -\frac{4651}{16266} + \frac{21600}{2711}Q_{15} \\ \frac{6591}{108440} - \frac{6480}{2711}Q_{15} \end{bmatrix}, \qquad \begin{bmatrix} a_{41} \\ a_{42} \\ a_{43} \\ a_{44} \\ a_{45} \\ a_{46} \\ a_{47} \end{bmatrix} = \begin{bmatrix} \frac{16033}{214360} + \frac{6480}{5359}Q_{15} \\ -\frac{1683}{10718} - \frac{21600}{5359}Q_{15} \\ -\frac{25157}{42872} - \frac{21600}{5359}Q_{15} \\ 0 \\ \frac{25157}{42872} - \frac{10800}{5359}Q_{15} \\ -\frac{547}{6990} + \frac{4320}{5359}Q_{15} \\ \frac{72}{5359} \end{bmatrix}$$

$$\begin{bmatrix} a_{51} \\ a_{52} \\ a_{53} \\ a_{54} \\ a_{55} \\ a_{56} \\ a_{57} \\ a_{58} \end{bmatrix} = \begin{bmatrix} -\frac{8460}{7877}Q_{15} \\ -\frac{1915}{63016} + \frac{32400}{7877}Q_{15} \\ \frac{4651}{23631} - \frac{43200}{7877}Q_{15} \\ -\frac{25157}{31508} + \frac{21600}{7877}Q_{15} \\ 0 \\ \frac{147127}{189048} - \frac{2160}{7877}Q_{15} \\ -\frac{1296}{7877} \\ \frac{144}{7877} \end{bmatrix}, \qquad \begin{bmatrix} a_{61} \\ a_{62} \\ a_{63} \\ a_{64} \\ a_{65} \\ a_{66} \\ a_{67} \\ a_{68} \\ a_{69} \end{bmatrix} = \begin{bmatrix} -\frac{20539}{1051244} + \frac{10800}{43801}Q_{15} \\ \frac{6964}{131403} - \frac{43200}{43801}Q_{15} \\ -\frac{6591}{175204} + \frac{64800}{43801}Q_{15} \\ \frac{12581}{131403} - \frac{43200}{43801}Q_{15} \\ \frac{735635}{1051244} + \frac{10800}{43801}Q_{15} \\ 0 \\ \frac{32400}{43801} \\ -\frac{6480}{43801} \\ \frac{720}{43801} \end{bmatrix}$$

3.2.2 Artificial Dissipation

The numerical dissipation term is added to the difference operator approximating the first derivatives. It was recommended that an upwinding scheme be used in the spatial discretization of the advection term to help eliminate high-frequency modes [22]. This approach is equivalent to adding a first-order dissipation operator to the centered difference operator [23]:

$$U\partial_\xi \tilde{\nu} \approx UD_1\tilde{\nu} + A_D$$

where

$$A_D = \frac{1}{2}|U|H^{-1}D_d^T C_1 D_d \tilde{\nu}$$

matrices D_d and C_1 are defined as

$$D_d = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 1 \end{bmatrix}, \qquad C_1 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix}.$$

Therefore, the resulting discretization for the dissipation component is:

$$\begin{aligned} &\text{low side:} && -|U_1|(\tilde{\nu}_2 - \tilde{\nu}_1) \\ &\text{interior:} && -\frac{1}{2}|U_j|(\tilde{\nu}_{j-1} - 2\tilde{\nu}_j + \tilde{\nu}_{j+1}) \\ &\text{high side:} && |U_N|(\tilde{\nu}_N - \tilde{\nu}_{N-1}) \end{aligned}$$

The first-order upwinding scheme affects the order of accuracy of the spatial discretization and thus another two dissipation models are introduced in an effort to make the dissipation model higher order. The first approach is the use of the classic dissipation model based on the formulation of the dissipation model for the mean flow [28]. It consists of two components: a fourth-difference operator to provide sufficient damping to the high-frequency modes and a second-difference operator which prevents oscillation at discontinuities. The classical difference operator is then applied with the diagonal norm H to ensure stability with the use of SBP operators. It is given as

$$A_D = H^{-1} D_d^T \big(C_{\xi,U}^{(2)} D_d + C_{\xi,U}^{(4)} D_d C_2 D_d^T D_d \big) \tilde{\nu},$$

where

$$C_{\xi,U}^{(2)} = \begin{bmatrix} \kappa_{2t}|U|_{\frac{3}{2}} \epsilon_{\frac{3}{2}}^{(\xi)} & & & & \\ & \kappa_{2t}|U|_{\frac{5}{2}} \epsilon_{\frac{5}{2}}^{(\xi)} & & & \\ & & \ddots & & \\ & & & \kappa_{2t}|U|_{N-\frac{1}{2}} \epsilon_{N-\frac{1}{2}}^{(\xi)} & \\ & & & & 0 \end{bmatrix},$$

$$C^{(4)}_{\xi,U} = \begin{bmatrix} \hat{\kappa}_{4t}|U|_{\frac{3}{2}} & & & & \\ & \hat{\kappa}_{4t}|U|_{\frac{5}{2}} & & & \\ & & \ddots & & \\ & & & \hat{\kappa}_{4t}|U|_{N-\frac{1}{2}} & \\ & & & & 0 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix},$$

and $|U|_{i+\frac{1}{2}} = \frac{1}{2}(|U|_i + |U|_{i+1})$. The coefficient for the fourth-difference term $\hat{\kappa}_{4t}$ is calculated as

$$\hat{\kappa}_{4t} = \max[0, \kappa_{4t} - \kappa_{2t}\epsilon^{(\xi)}_{i+\frac{1}{2}}].$$

A $\tilde{\nu}$-switch similar to the pressure sensor controls the contribution of the second-difference term. The coefficient for the second-difference operator will be much larger when there is a sudden change in the turbulence quantity and will turn off the fourth-difference operator if the coefficient becomes larger than κ_{4t}. The $\tilde{\nu}$-switch is defined as follows [38]:

$$\begin{aligned} \epsilon^{(\xi)}_{i+\frac{1}{2}} &= \epsilon^{(\xi)}_i + \epsilon^{(\xi)}_{i+1}, \\ \epsilon^{(\xi)}_i &= \frac{1}{4}(\Gamma^*_{i-1} + 2\Gamma^*_i + \Gamma^*_{i+1}), \\ \Gamma^*_i &= max[\Gamma_{i-1}, \Gamma_i, \Gamma_{i+1}], \end{aligned}$$

$$\Gamma_i = \begin{cases} \frac{|\tilde{\nu}_{i-1} - 2\tilde{\nu}_i + \tilde{\nu}_{i+1}|}{|\tilde{\nu}_{i-1} + 2\tilde{\nu}_i + \tilde{\nu}_{i+1}|} & \text{if } |\tilde{\nu}_{i-1} + 2\tilde{\nu}_i + \tilde{\nu}_{i+1}| > 0 \\ 0 & \text{if } |\tilde{\nu}_{i-1} + 2\tilde{\nu}_i + \tilde{\nu}_{i+1}| = 0 \end{cases}$$

The second approach is the combination of a second-difference operator and a high-order SBP dissipation operator based on the high-order dissipation scheme proposed by Dias for the Euler equations [26]. It is formulated as

$$A_D = H^{-1}(D_d^T C_{\xi,U}^{(2)} D_d + D_p^T C_{\xi,U}^{(4)} D_p)\tilde{\nu}$$

where H^{-1} is consistent with those used in derivative operators. The contribution of the second-difference term and the high-order SBP dissipation term is again controlled by the $\tilde{\nu}$-switch. For the third-order accurate SBP dissipation, D_p is given as

$$D_p = \begin{bmatrix} -1 & 3 & -3 & 1 & & & \\ -1 & 3 & -3 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & & 1 & -3 & 3 & 1 \\ & & & 1 & -3 & 3 & 1 \\ & & & & & & 0 \end{bmatrix}.$$

The third-order dissipation operator maps onto the half nodes and thus the matrix $C_{\xi,U}^{(4)}$ is given as

$$C_{\xi,U}^{(4)} = \begin{bmatrix} \hat{\kappa}_{4t}|U|_{\frac{3}{2}} & & & \\ & \ddots & & \\ & & \hat{\kappa}_{4t}|U|_{N-\frac{1}{2}} & \\ & & & 0 \end{bmatrix}$$

For the fourth-order SBP dissipation the D_p and $C_{\xi,U}^{(4)}$ are given as

$$D_p = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 6 & -4 & 1 \end{bmatrix},$$

$$C_{\xi,U}^{(4)} = \begin{bmatrix} \hat{\kappa}_{4t}|U|_1 & & & \\ & \ddots & & \\ & & \hat{\kappa}_{4t}|U|_{N-1} & \\ & & & \hat{\kappa}_{4t}|U|_N \end{bmatrix}.$$

For the fifth-order SBP dissipation the D_p and $C^{(4)}_{\xi,U}$ are given as follows (again it maps onto half nodes):

$$D_p = \begin{bmatrix} -1 & 5 & -10 & 10 & -5 & 1 & & \\ -1 & 5 & -10 & 10 & -5 & 1 & & \\ -1 & 5 & -10 & 10 & -5 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 5 & -10 & 10 & -5 & 1 \\ & & -1 & 5 & -10 & 10 & -5 & 1 \\ & & -1 & 5 & -10 & 10 & -5 & 1 \\ & & & & & & & 0 \end{bmatrix},$$

$$C^{(4)}_{\xi,U} = \begin{bmatrix} \hat{\kappa}_{4t}|U|_{\frac{3}{2}} & & & \\ & \ddots & & \\ & & \hat{\kappa}_{4t}|U|_{N-\frac{1}{2}} & \\ & & & 0 \end{bmatrix}.$$

3.2.3 Diffusive Terms

The diffusion terms with double derivatives can be approximated either using the minimum-stencil operator for a second derivative or by applying the first derivative twice. If the minimum-stencil SBP operator is used, a double derivative is approximated as

$$\partial_\xi(\beta\partial_\xi\alpha) \approx D_2(\beta)\alpha$$

where β represents a variable coefficient. By default, the flow solver Diablo uses the minimum-stencil operator for the second derivative [33] which results in the following spatial discretization:

$$\begin{aligned} &\text{low side:} && -\beta_1[2(\alpha_2-\alpha_1)-(\alpha_3-\alpha_2)]+\beta_2(\alpha_2-\alpha_1), \\ &\text{interior:} && \frac{1}{2}(\beta_{j+1}+\beta_j)(\alpha_{j+1}-\alpha_j)-\frac{1}{2}(\beta_j+\beta_{j-1})(\alpha_j-\alpha_{j-1}), \\ &\text{high side:} && \beta_N[2(\alpha_N-\alpha_{N-1})-(\alpha_{N-1}-\alpha_{N-2})]-\beta_{N-1}(\alpha_N-\alpha_{N-1}). \end{aligned} \tag{3.1}$$

If the non-compact operator is used, the resulting spatial discretization is given as:

$$
\begin{aligned}
&\text{first row:} \quad -\beta_1(-\alpha_1+\alpha_2)+\frac{1}{2}\beta_2(-\alpha_1+\alpha_3),\\
&\text{second row:} \quad -\frac{1}{2}\beta_1(\alpha_2-\alpha_1)+\frac{1}{4}\beta_3(\alpha_4-\alpha_2),\\
&\text{interior:} \quad -\frac{1}{4}\beta_{j-1}(\alpha_j-\alpha_{j-2})+\frac{1}{4}\beta_{j+1}(\alpha_{j+2}-\alpha_j),\\
&\text{second last row:} \quad -\frac{1}{4}\beta_{N-1}(-\alpha_{N-3}+\alpha_{N-1})+\frac{1}{2}\beta_N(\alpha_N-\alpha_{N-1}),\\
&\text{last row:} \quad \beta_N(-\alpha_{N-1}+\alpha_N)-\frac{1}{2}\beta_{N-1}(-\alpha_{N-2}+\alpha_N).
\end{aligned}
$$

3.2.4 Interface SATs

The interface SATs consist of three SATs, one for the advective term and two for the diffusion term. The interface SATs for the turbulence model equation are formed in a similar way as the SATs for the Navier-Stokes equations [39]. The SAT associated with the advection term is given as [38]

$$H_b^{-1}\sigma_a(\tilde{\nu}-\tilde{\nu_2})$$

where H_b takes elements for the boundary nodes from the diagonal norm matrix H, and σ_a is calculated as

$$\sigma_a=-\frac{1}{2}[\max(|U|,\theta)+\delta_a U], \quad \delta_a=\begin{cases}+1, & \text{low side}\\ -1, & \text{high side}\end{cases}$$

The contravariant velocity U is taken as the average value between the coincident nodes at the interface. θ is a limiting factor defined as a function of the spectral radius

$$\theta=V_l(|U|+a\sqrt{\xi_x^2+\xi_y^2+\xi_z^2}), \quad V_l=0.025.$$

The SATs for the diffusive term are given as:

$$-H_b^{-1}\frac{1}{4\sigma Re}\sigma_{dv}(\tilde{\nu}-\tilde{\nu_2})+H_b^{-1}\sigma_{df}(g-g_2)$$

where ν_2 is the turbulence quantity value from the coincident node on the adjacent block, and g_2 is the turbulence quantity gradient from the coincident node on the adjacent block. The first term accounts for the difference in the coincident turbulence quantities where σ_{dv} is calculated as

$$\sigma_{dv}=(\nu+\tilde{\nu})(\xi_x^2+\xi_y^2+\xi_z^2).$$

It should be noted that at the interfaces, average state values are used to calculate σ_{dv} while the grid metrics are based on local information only. The second term accounts for the difference in the gradients of the turbulence quantities. The parameter σ_{df} is $= +1$ at the low side of a block and -1 at the high side. The gradient g is given as

$$g = \frac{1}{\sigma Re}(\nu + \tilde{\nu})(\xi_x^2 + \xi_y^2 + \xi_z^2)\delta_\xi \tilde{\nu},$$

where $\sigma = \frac{2}{3}$, and $\delta_\xi \tilde{\nu}$ is approximated using the first derivative operator consistent with the second derivative SBP operator. If a compact operator is used for the second derivative, D_b as defined for the minimum-stencil operator is used to approximate the first derivative at the interface. For the second-order method, $\delta_\xi \tilde{\nu}$ is approximated as

$$\begin{aligned} &\text{low side:} \qquad (\delta_\xi \tilde{\nu})_1 \approx -\frac{3}{2}\tilde{\nu}_1 + 2\tilde{\nu}_2 - \frac{1}{2}\tilde{\nu}_3 \\ &\text{high side:} \qquad (\delta_\xi \tilde{\nu})_N \approx \frac{3}{2}\tilde{\nu}_N - 2\tilde{\nu}_{N-1} + \frac{1}{2}\tilde{\nu}_{N-2} \end{aligned}$$

If the second derivative is approximated by applying the first derivative operator twice, then $\delta_\xi \tilde{\nu}$ for the second-order method is approximated as

$$\begin{aligned} &\text{low side:} \qquad (\delta_\xi \tilde{\nu})_1 \approx -\tilde{\nu}_1 + \tilde{\nu}_2 \\ &\text{high side:} \qquad (\delta_\xi \tilde{\nu})_N \approx -\tilde{\nu}_{N-1} + \tilde{\nu}_N \end{aligned}$$

The approximation of the turbulent quantity gradient at the interfaces for the 3rd- and 4th-method is included in the appendix.

In the negative turbulence model, the diffusion terms are given as

$$\frac{1}{\sigma}\nabla\cdot[(\nu + \tilde{\nu} f_n)\nabla\tilde{\nu}] + c_{b2}\sigma(\nabla\tilde{\nu})^2.$$

In this case, the gradient g is in the following form to account for the parameter f_n

$$g = \frac{1}{\sigma Re}(\nu + f_n\tilde{\nu})(\xi_x^2 + \xi_y^2 + \xi_z^2)\delta_\xi \tilde{\nu},$$

and

$$\sigma_{dv} = (\nu + f_n\tilde{\nu})(\xi_x^2 + \xi_y^2 + \xi_z^2).$$

3.2.5 Boundary SATs

At domain boundaries, the SATs are formed in the same way as the interface SATs except that the contravariant velocity U and the parameter σ_{dv} is calculated using the state values at the local node. At wall surfaces, the target value, $\tilde{\nu}_w$ is set to zero.

At the farfield boundary, the target value of $\tilde{\nu}_{far}$ is set to 3.0 for fully turbulent flows [40]. If the transition location is specified, the target value is set to 0.1 [38]. Meanwhile, the target turbulence quantity gradient is zero.

At the symmetry boundary, the target value of $\tilde{\nu}$ is taken from one node inside the boundary. The gradient in $\tilde{\nu}$ is zero normal to the symmetry boundary. The resulting SATs are summarized in Tables 3.1 and 3.2 for the advection and diffusion terms respectively.

Boundary Type	**Form**	**Target Value**
wall	$H_b^{-1}\sigma_a(\tilde{\nu} - \tilde{\nu}_{target})$	$\tilde{\nu}_w = 0$
symmetry		$\tilde{\nu}_{j+1}$
farfield		$\tilde{\nu}_{far}$
interface		$\tilde{\nu}_2$

Table 3.1: Advection term SATs at block boundaries

Boundary Type	**Form**	**Target Value**
wall	$-H_b^{-1}\frac{1}{4\sigma Re}\sigma_{dv}(\tilde{\nu} - \tilde{\nu_2})$	$\tilde{\nu}_w = 0$
symmetry		$\tilde{\nu}_{j+1}$
farfield	$H_b^{-1}\sigma_{df}(g - g_2)$	$g_{far} = 0$
interface	$-H_b^{-1}\frac{1}{4\sigma Re}\sigma_{dv}(\tilde{\nu} - \tilde{\nu_2})$ $+H_b^{-1}\sigma_{df}(g - g_2)$	$\tilde{\nu}_2$
		g_2

Table 3.2: Diffusion term SATs at block boundaries

Target	**Description**
$\tilde{\nu}_w$	Turbulence quantity at the surface boundary
$\tilde{\nu}_far$	Turbulence quantity at the farfield boundary
$\tilde{\nu}_{j+1}$	Turblence quantity from one node inside the symmetry boundary
$\tilde{\nu}_2$	Turbulence quantity at the coincident node on the adjacent block
g_{far}	Gradient in turbulence quantity at the farfield boundary
g_2	Gradient in turbulence quantity at the coincident node on the adjacent block

Table 3.3: Target term description

Chapter 4

Results

Two test cases are included in this chapter to verify the code and to examine the efficiency of high-order methods. This chapter includes two parts: the description of the test cases and the discussion of the results. The test cases included in this chapter are provided by the Turbulence Modelling Resource (TMR) website, and the reference values are generated by two independent RANS codes, FUN3D and CFL3D, given on the TMR website.

4.1 Test Cases and Grids

Two two-dimensional geometries are studied in this paper: a flat plate and a bump-in-channel. Both test cases and grid files are provided by the TMR website. For each test case, a grid family consisting of four grid levels is used for the purpose of grid convergence studies. In each grid family, a coarser mesh is generated by removing every other node from the finer mesh.

4.1.1 Case 1: Zero Pressure Gradient Flat Plate

The flow conditions for this case are: $M = 0.2$, $Re_L = 5 \times 10^6 (L = 1)$, $T_{ref} = 540R$ where the Reynolds number is based on a reference length, and the length of the flat plate is 2 reference units. The solid wall surface starts from $x = 0$ to $x = 2$. The inflow boundary conditions are given by $P_t/P_{ref} = 1.02828$ and $T_t/T_{ref} = 1.008$, and the outflow condition is $P/P_{ref} = 1.0$. Four grid levels are considered in this case, as summarized in Table 4.1. A sample grid is shown Figure 4.1.

Grid Level	Number of Nodes
Level 1	69 × 49
Level 2	137 × 97
Level 3	273 × 193
Level 4	545 × 385

Table 4.1: Four grid levels for Case 1

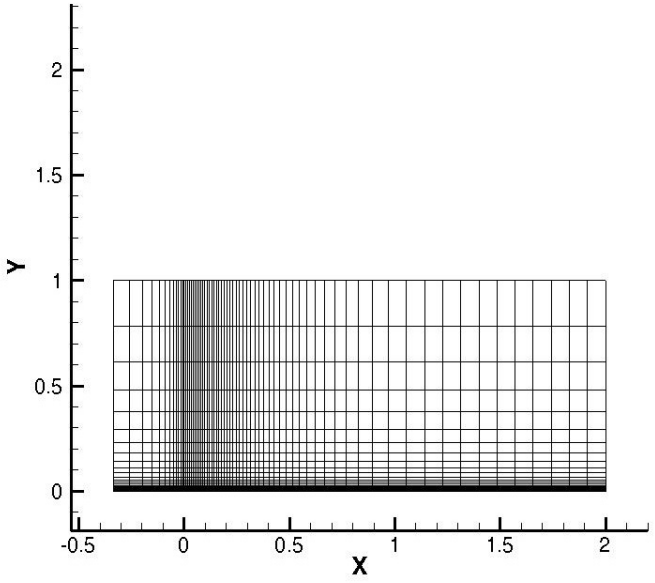

Figure 4.1: The 69 × 49 Grid. Grids are provided by the TMR website.

4.1.2 Case 2: Bump-in-Channel

The bump-in-channel case is run at $M = 0.2$ and a Reynolds number of $Re_L = 3 \times 10^6 (L = 1)$. The Reynolds number is based on a reference length, and the length of the channel is 1.5 reference units starting at $x = 0$ to $x = 1.5$. The maximum height of the bump is 0.05. The inflow conditions are $P_t/P_{ref} = 1.02828$ and $T_t/T_{ref} = 1.008$, and the outflow condition is given as $P/P_{ref} = 1.0$. The reference temperature is $T_{ref} = 540R$. Four grid levels are considered, as shown in Table 4.2.

A portion of a sample grid is shown in Figure 4.2. The upstream and downstream farfield boundaries are both 25 units length away from the wall surface. A symmetry boundary condition is imposed before the start of plate and after the end of plate.

Grid Level	Number of Nodes
Level 1	177 × 81
Level 2	353 × 161
Level 3	705 × 321
Level 4	1409 × 641

Table 4.2: Four grid levels for Case 2

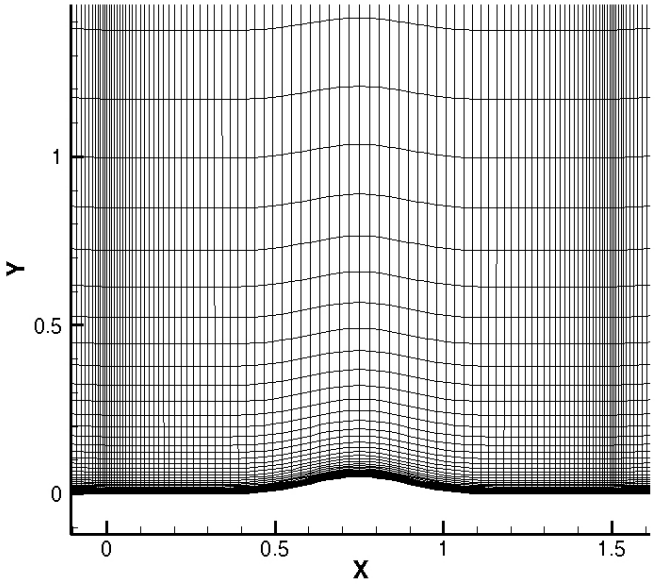

Figure 4.2: Portion of 177 x 81 grid for the bump-in-channel. The grids were provided by the TMR website

4.2 Results

4.2.1 Zero Pressure Gradient Flat Plate

The surface friction coefficient c_f obtained on the finest grid level using the second-order method is plotted in Figure 4.3 over the entire plate in comparison to results obtained using the flow solver CFL3D. The CFL3D results are provided on the TMR website for compressible RANS code validation. The present results (Diablo) show excellent agreement with the CFL3D results. Figure 4.4 shows the convergence of c_f at $x = 0.97008$ with mesh size using the second-order, third-order, and fourth-order methods in comparison with those from CFL3D and FUN3D. The x-axis shows $\sqrt{1/N}$ where N is the number of total nodes. It can seen that as the grid is refined, the difference among the results obtained by different codes diminishes. It also shows that the third- and fourth-order results display a reduction in error on the coarser meshes relative to the second-order results.

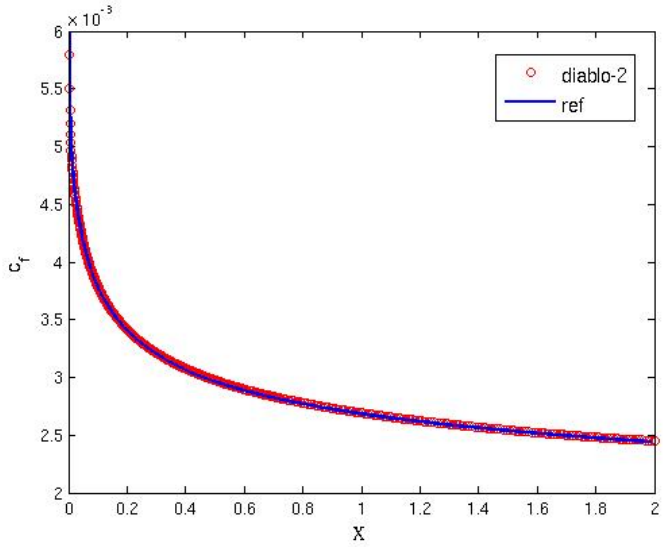

Figure 4.3: Surface friction coefficient on the finest mesh for Case 1, zero pressure gradient flat plate, computed using the second-order spatial discretization. Reference values are obtained with CFL3D provided on the TMR website.

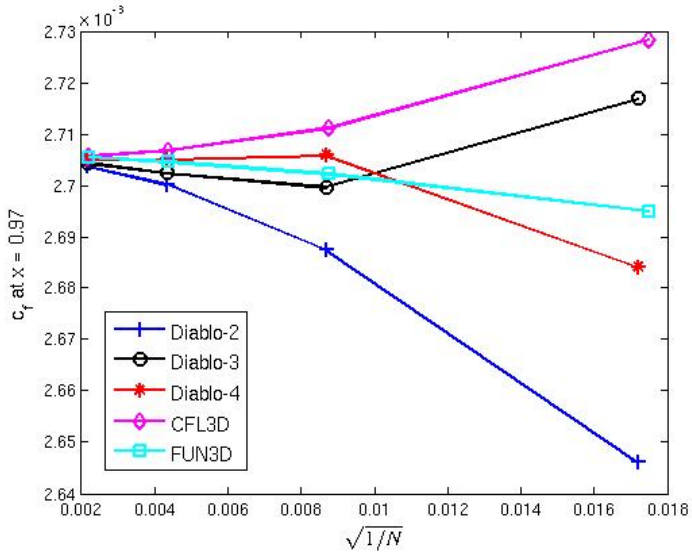

Figure 4.4: Surface friction coefficient at $x = 0.97$ for Case 1, zero pressure gradient flat plate. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

Figure 4.5 shows the eddy viscosity (μ_t) contours normalized with freestream laminar viscosity (μ_∞) near the solid wall surface on the 273×193 grid using the 4th-order method. The resulting contours show excellent agreement with those provided on the TMR website using CFL3D and FUN3D. Meanwhile, the nondimensionalized eddy viscosity profile at a specific point is also examined to

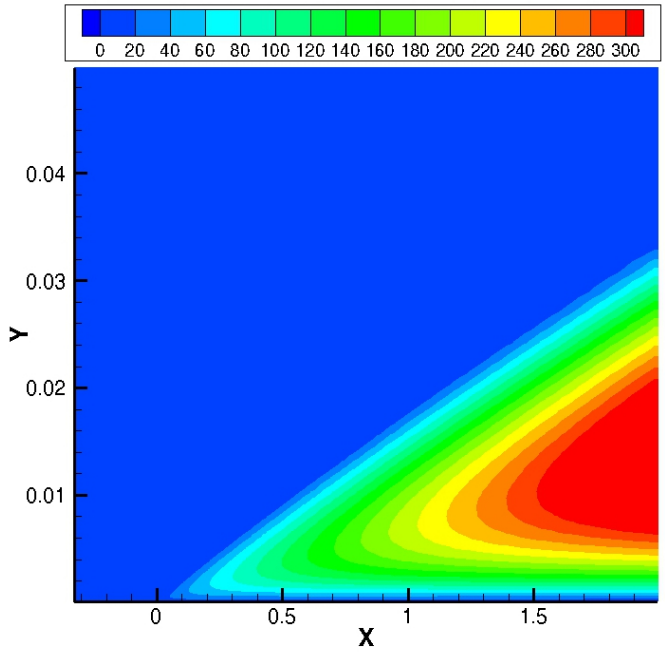

Figure 4.5: The eddy viscosity contour nondimensionalized by the freestream laminar viscosity on the 273 × 193 grid using the fourth-order method for Case 1

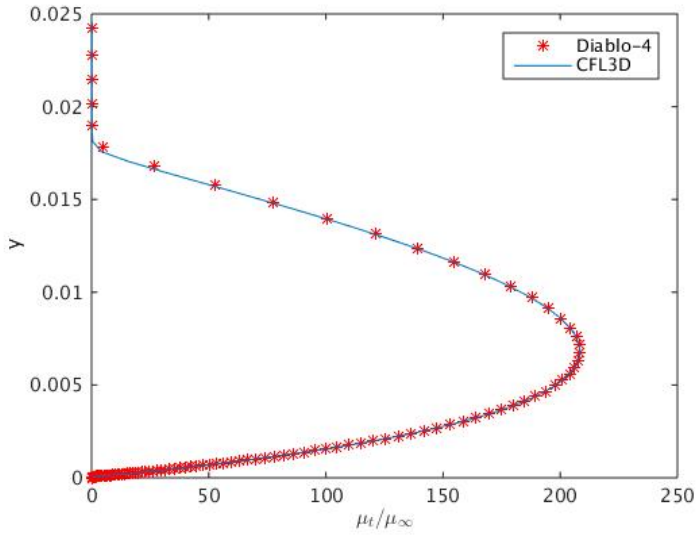

Figure 4.6: The eddy viscosity profile nondimensionalized by the freestream laminar viscosity on the 273 × 193 grid using the 4th-order method for Case 1

verify the implementation of the high-order turbulence model. At the point $x = 0.97$ the eddy viscosity profile obtained on the same 273 × 193 mesh using the fourth-order method is plotted in Figure 4.6 in comparison to the one generated by CFL3D. The resulting profile also shows excellent agreement.

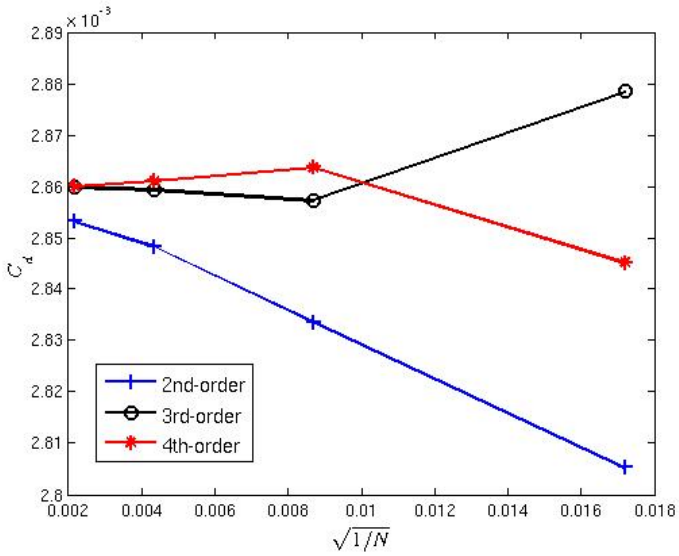

Figure 4.7: Grid convergence of C_d for Case 1, zero pressure gradient flat plate

Code	P	$\mathbf{C_d^*}$
Diablo-2	1.59	2.8556×10^{-3}
Diablo-3	1.77	2.8623×10^{-3}
Diablo-4	1.31	2.8593×10^{-3}

Table 4.3: Computed order P and the grid converged C_d^* via grid convergence studies. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

Figure 4.7 shows the drag coefficient produced by the high-order methods compared to the second-order method for Case 1. Grid convergence studies are carried out to estimate numerical errors. The computed order, P, and grid-converged value, C_d^*, from grid convergence studies are summarized in Table 4.3. The order achieved is less than two in all cases, presumingly due to the leading and trailing edge singularities. Numerical errors on each mesh are calculated relative to the corresponding grid-converged drag coefficient value, C_d^*. For example, the error in C_d obtained with the second-order method on the coarsest mesh is calculated relative to C_d^* obtained by the second-order method. Numerical errors as a function of mesh size are plotted in Figure 4.8, while Table 4.4 lists the numerical errors on the coarsest three grid levels. It is shown that the higher-order methods in general produce more accurate results than the second-order method. On coarse meshes, numerical errors can be reduced significantly by increasing the order of the spatial discretization from second-order to third- or fourth-order. On the coarsest mesh, the errors are reduced by 68.7% and 72.1% with the third- and fourth-order methods, respectively. The advantage of using the fourth-order method over the third-order method is less significant, as the errors with the third-order method are already very small. Still, error reductions by using the fourth-order method over the third-order

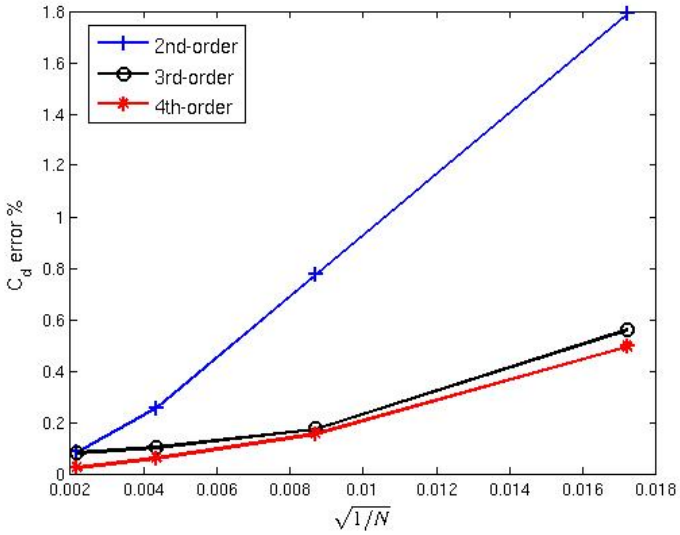

Figure 4.8: Numerical errors in C_d on each grid level for Case 1, zero pressure gradient flat plate. N is the total number of nodes in each mesh.

Algorithm	$\%\mathbf{C_d^{(1)}}$	$\%\mathbf{C_d^{(2)}}$	$\%\mathbf{C_d^{(3)}}$
2nd-order	1.79	0.78	0.26
3rd-order	0.56	0.18	0.10
4th-order	0.50	0.15	0.06

Table 4.4: Errors for Case 1. $\%C_d$ represents the percentage error relative to the predicted grid converged value. Superscript (n) represents the corresponding grid level. The finer the grid, the larger the n.

method ranging from 10% to 40% are seen on the coarse meshes.

Increased accuracy by using the high-order spatial discretization is achieved at the cost of increased computational cost per grid node. The computational cost in this case is measured in terms of total CPU time needed to reduce the residual by 12 orders of magnitude. Figure 4.9 is a log plot of CPU time vs percentage error in C_d. On coarse meshes, the third-order method is the most efficient as it takes the least amount of CPU time to achieve the same level of accuracy. The fourth-order method is less efficient than the third-order method but still performs better than the second-order method. As the error tolerance is tightened, the fourth-order method becomes the most efficient.

4.2.2 Bump-in-Channel

The surface friction coefficient c_f and surface pressure coefficient c_p obtained on grid level three using the second-order method are plotted in Figures 4.10 and 4.11 over the entire channel in comparison to results obtained using CFL3D. Despite local variations near the leading and trailing

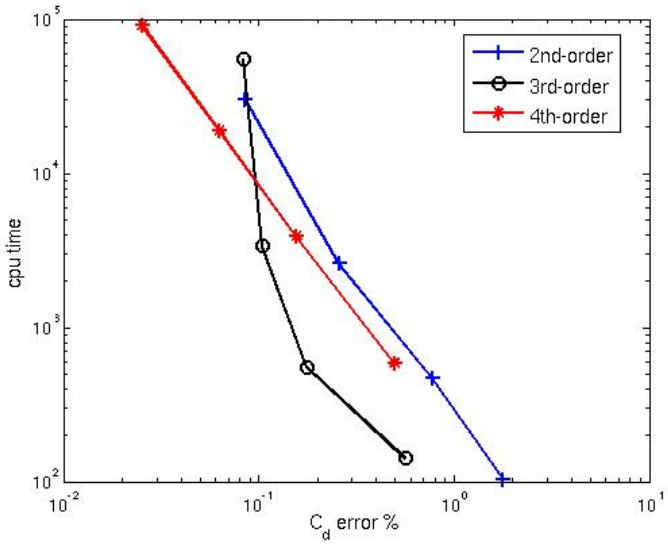

Figure 4.9: Computational cost for Case 1, zero pressure gradient flat plate

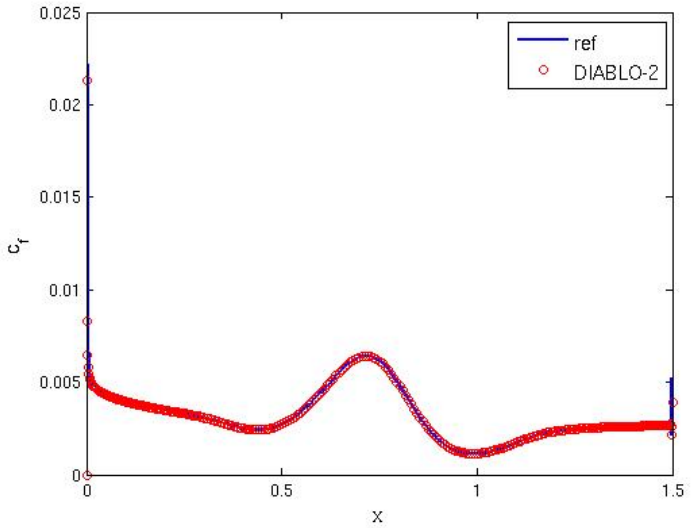

Figure 4.10: Surface friction coefficient obtained on the finest mesh in comparison to reference values for Case 2, bump-in-channel

edges, the results are nearly identical to the reference values.

Figures 4.12, 4.13 and 4.14 show the convergence of the surface friction coefficient at the bump peak, in front of the bump peak, and aft of the bump peak. Results obtained with the second-order and the high-order methods are plotted together with results by FUN3D and CFL3D. In all cases, the resulting surface friction coefficient approaches the same result as the grid is refined. From Figures 4.12 and 4.13, it can be seen that the high-order methods are more accurate than the second-order method at points in front of and at the bump peaks. At the point aft of the

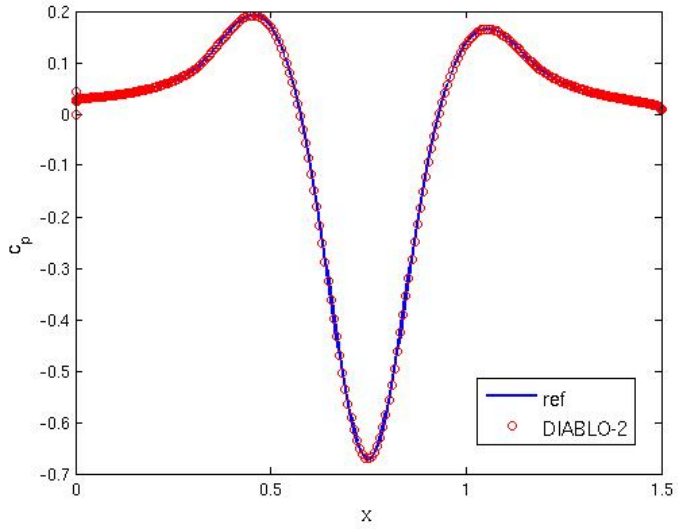

Figure 4.11: Surface pressure coefficient obtained on the finest mesh in comparison to reference values for Case 2, bump-in-channel

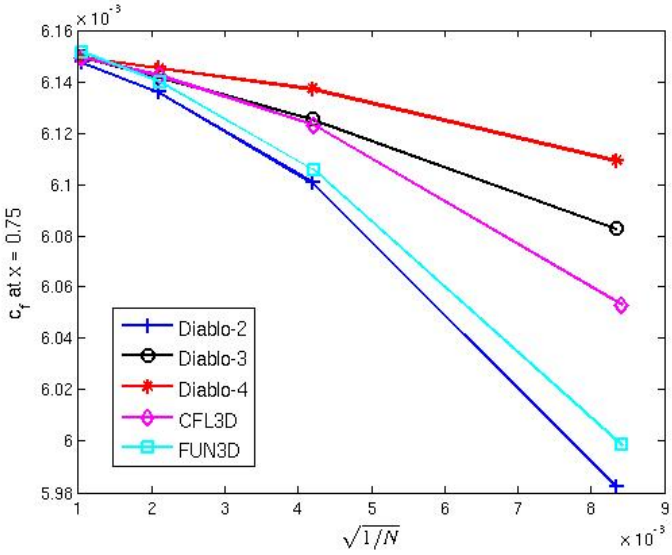

Figure 4.12: Surface friction coefficient at the bump peak ($x = 0.75$) for Case 2, bump-in-channel. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

bump peak, the fourth-order method produces more accurate results among the three while the third-order method is the least accurate.

Figure 4.15 shows the nondimensional eddy viscosity contours near the bump obtained on the 705×321 grid using the fourth-order method. The resulting contours show excellent agreement with those generated by CFL3D and FUN3D. In Figure 4.16 the normalized eddy viscosity profile generated on the same grid using the fourth-order method is plotted in comparison to the eddy

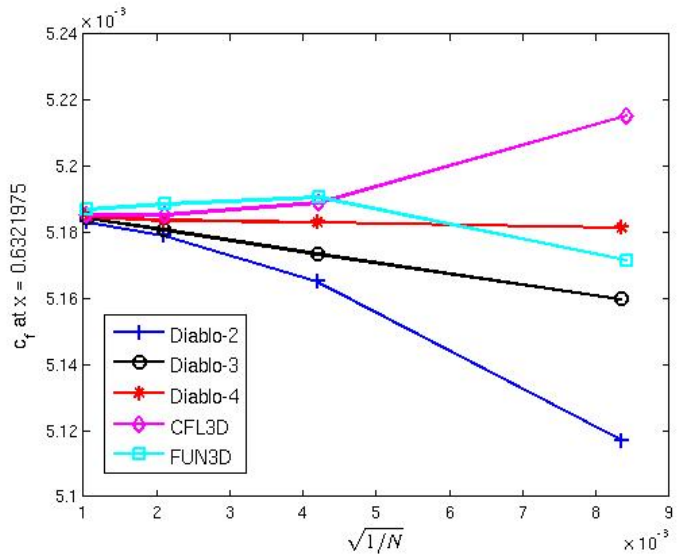

Figure 4.13: Surface friction coefficient in front of the bump peak ($x = 0.6321975$) for Case 2, bump-in-channel. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

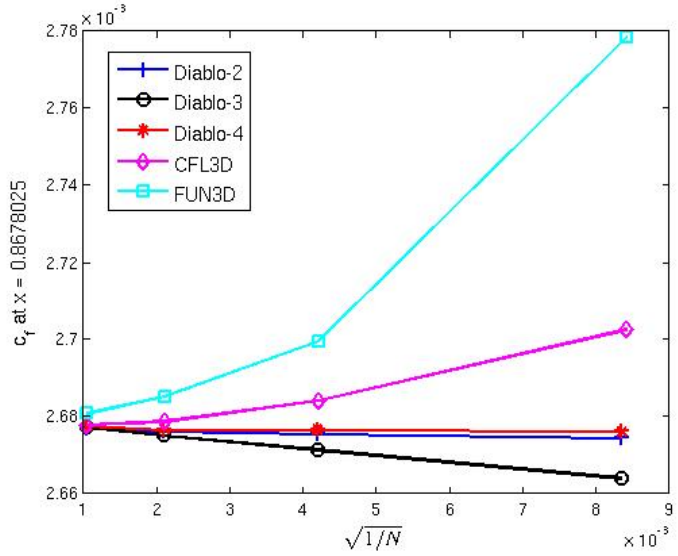

Figure 4.14: Surface friction coefficient aft of the bump peak ($x = 0.8678035$) for Case 2, bump-in-channel. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

viscosity profile generated using CFL3D to examine the numerical correctness of the results. The resulting eddy viscosity profile agrees well with the reference profile.

Figures 4.17 and 4.18 show the drag coefficient C_d and the lift coefficient C_l, respectively, produced by the high-order methods in comparison to the second-order method for Case 2. The computed order, P, and grid converged value, C_d^*, from the grid convergence studies are summarized in Table

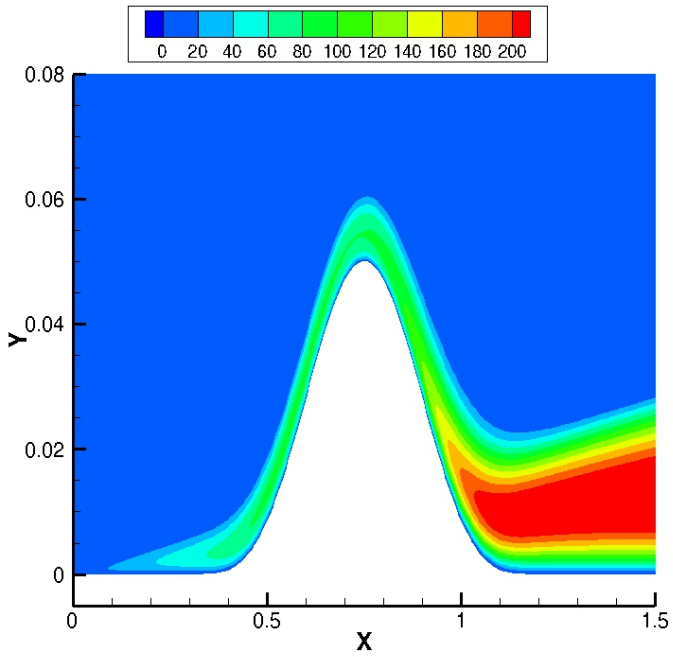

Figure 4.15: The eddy viscosity contour nondimensionalized by the freestream laminar viscosity on the 705 × 321 grid using the fourth-order method for Case 2

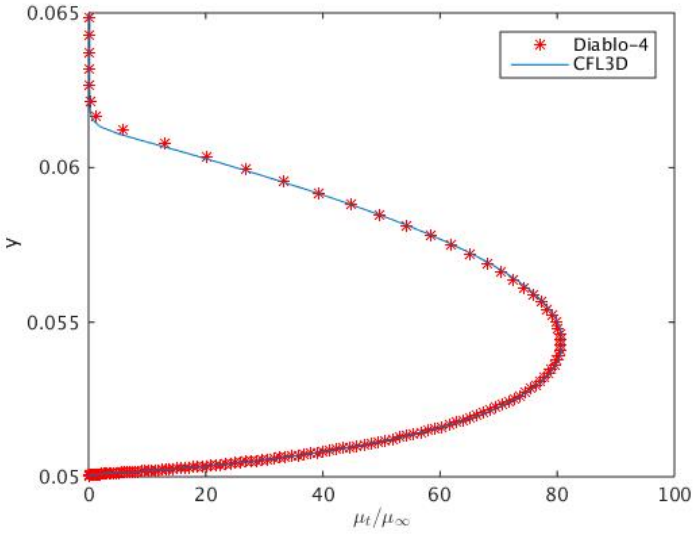

Figure 4.16: The eddy viscosity profile nondimensionalized by the freestream laminar viscosity on the 705 × 321 grid using the fourth-order method for Case 2

4.5. Similarly, numerical errors are estimated relative to grid converged values. Numerical errors in C_d and C_l over the varying mesh sizes are plotted in Figures 4.19 and 4.20. Table 4.6 lists the numerical errors in C_d on the three coarsest grid levels, and Table 4.7 the errors in C_l. In this case, the use of the third-order method has demonstrated reduced numerical errors in both C_d and C_l compared to the second-order method. Comparing to the results obtained by the second-order

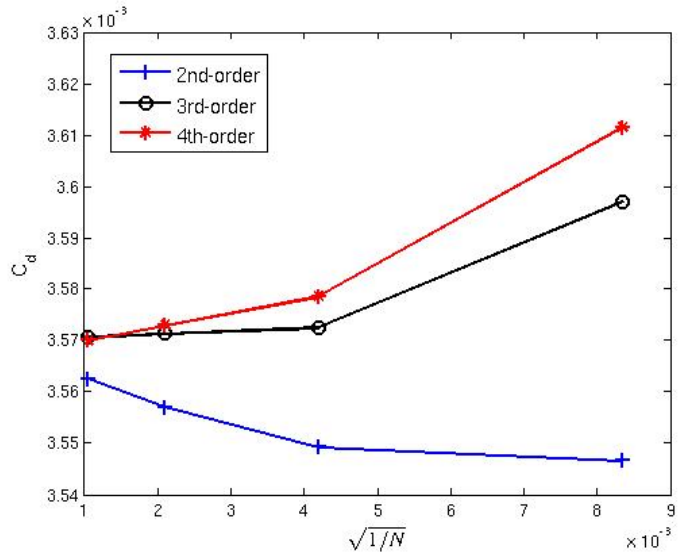

Figure 4.17: Grid convergence of C_d for Case 2, bump-in-channel

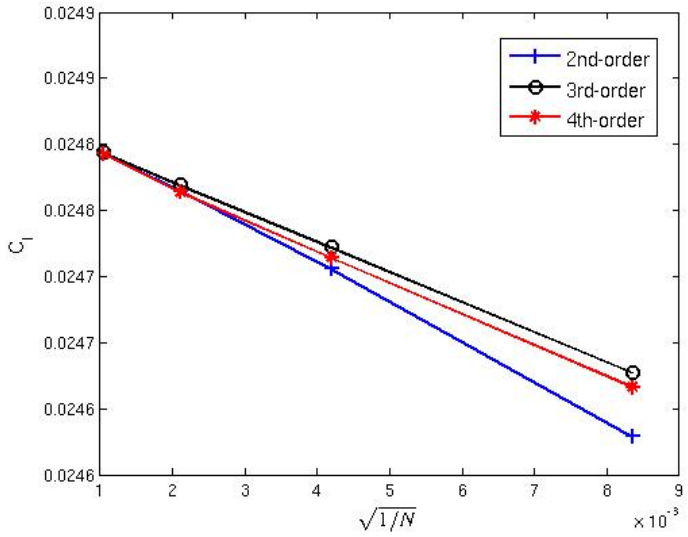

Figure 4.18: Grid convergence of C_l for Case 2, bump-in-channel

method, errors in C_d on grid levels two and three are reduced by approximately 90% by the use of third-order method. In the case of C_l, the error in C_l on the coarsest mesh is reduced by 19.4%. On the other hand, whether applying the fourth-order method is beneficial is arguable. Numerical errors in C_d on coarse meshes are slightly reduced, while the performance of the fourth-order method in terms of accuracy in C_l is on par with the second-order method.

In order to evaluate the efficiency of using high-order methods, the numerical errors are plotted against CPU time to converge in Figures 4.21 and 4.22. In this case, the solution is said to be converged when residual drops below -10 orders of magnitude or is reduced by 12 orders of

Code	$\mathbf{P_{C_d}}$	$\mathbf{C_d^* \times 10^{-3}}$	$\mathbf{P_{C_l}}$	$\mathbf{C_l^* \times 10^{-2}}$
Diablo-2	0.51	3.5754	1.06	2.4869
Diablo-3	0.95	3.5699	0.94	2.4871
Diablo-4	0.88	3.5664	0.79	2.4881

Table 4.5: Computed order P_{C_d} and P_{C_l}, and the corresponding grid-converged C_d^* and C_l^* via grid convergence studies. Diablo-2, Diablo-3, Diablo-4 represent results obtained using flow solver Diablo with second-order, third-order, fourth-order methods, respectively.

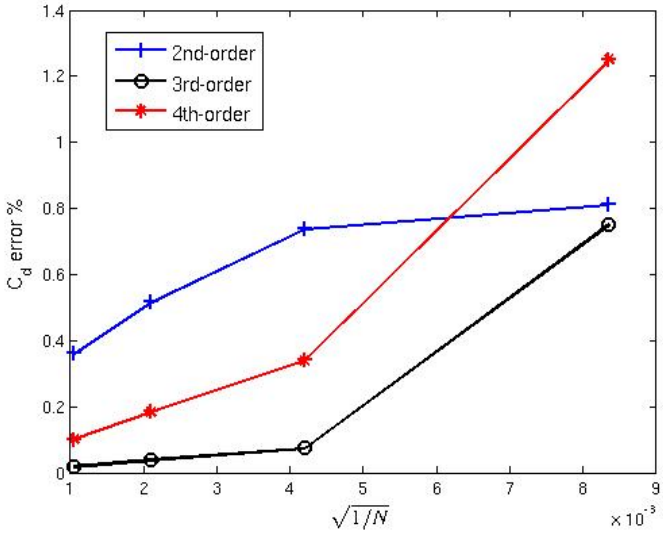

Figure 4.19: Numerical error in C_d for Case 2, bump-in-channel

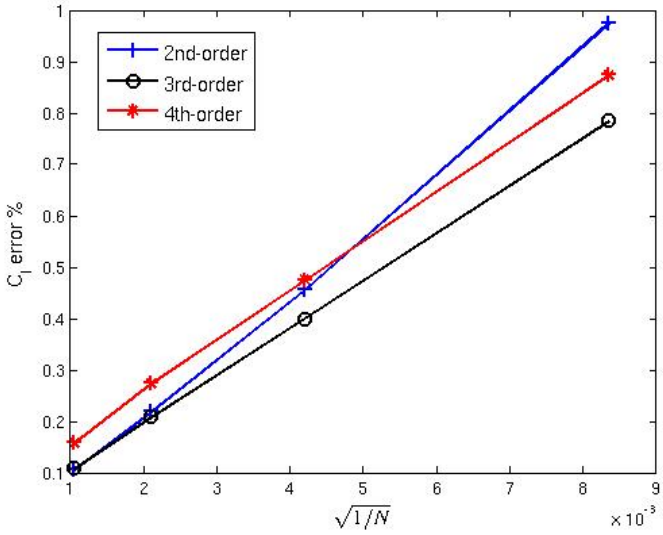

Figure 4.20: Numerical error in C_l for Case 2, bump-in-channel

Algorithm	$\%\mathbf{C_d^{(1)}}$	$\%\mathbf{C_d^{(2)}}$	$\%\mathbf{C_d^{(3)}}$
2nd-order	0.81	0.74	0.52
3rd-order	0.75	0.08	0.04
4th-order	1.25	0.33	0.18

Table 4.6: Errors in C_d for Case 2. $\%C_d$ represents the percentage error relative to the predicted grid converged values. Superscript (n) represents the corresponding grid level. The finer the grid, the larger the n.

Algorithm	$\%\mathbf{C_l^{(1)}}$	$\%\mathbf{C_l^{(2)}}$	$\%\mathbf{C_l^{(3)}}$
2nd-order	0.98	0.46	0.22
3rd-order	0.79	0.40	0.21
4th-order	0.87	0.48	0.27

Table 4.7: Errors in C_l for Case 2. $\%C_l$ represents the percentage error relative to the predicted grid converged values. Superscript (n) represents the corresponding grid level. The finer the grid, the larger the n.

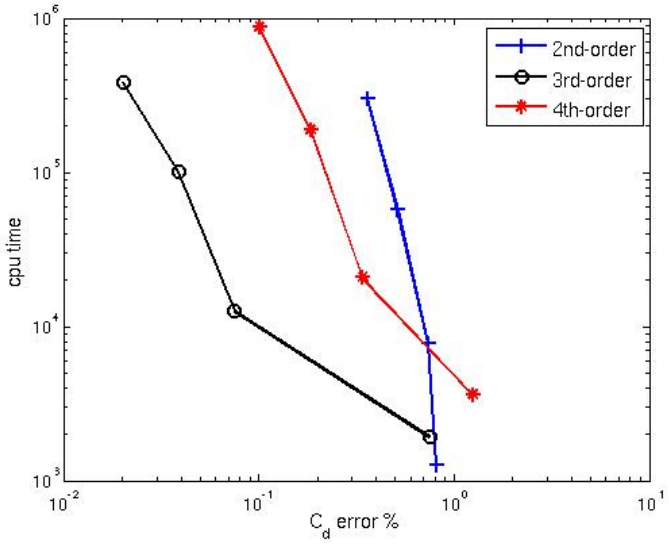

Figure 4.21: Numerical errors in C_d vs compuational cost for Case 2, bump-in-channel

magnitude from the first iteration. As discussed above, the use of the third-order method has shown the ability to reduce the errors significantly on coarse meshes. In terms of efficiency, the third-order method has demonstrated to be the best option among the three. In Figure 4.21, at the same level of computational cost numerical errors in C_d with the third-order method are smaller than those with the second-order method by one order of magnitude, whereas in Figure 4.22 the third-order method is slightly less efficient than the second-order method. Again, the performance of the fourth-order method in terms of efficiency is ambiguous. It demonstrates to be more efficient

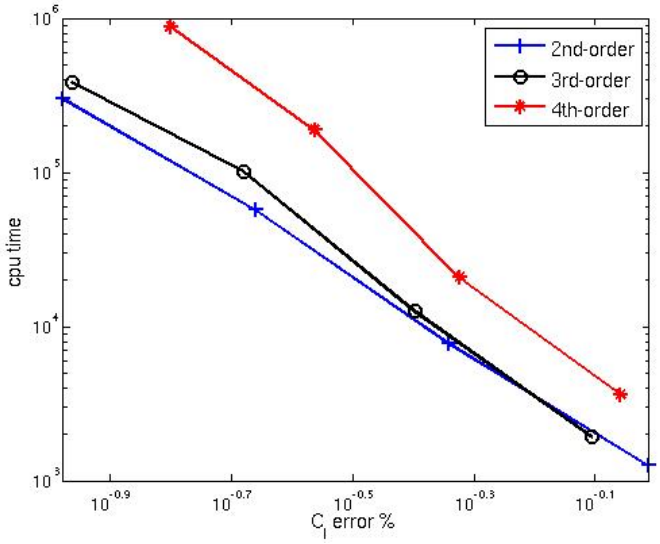

Figure 4.22: Numerical errors in C_l vs computational cost for Case 2, bump-in-channel

than the second-order method in the first plot but too expensive to solve in the second. Further study is needed to understand why the results for C_l differ from those for C_d.

Chapter 5

Conclusions and Recommendations

This chapter includes two sections: conclusions and recommendations. In the first section, work done in this thesis is summarized with emphasis on the investigation into the efficiency of the high-order methods. In the section for recommendations, a few possible additions to the turbulence model are discussed.

5.1 Conclusions

The high-order SBP-SAT method is implemented to the RANS-SA model. Test cases provided by the TMR website are used to verify and validate the code. Grid convergence studies are carried out to investigate the efficiency of the high-order methods. The results from the test cases have shown that using high-order methods generally reduces the numerical errors compared to the second-order method on a given mesh. In particular, the third-order method leads to a significant reduction in errors on coarse meshes in these two cases. On a given mesh, the high-order methods solve the flows at higher computational cost per grid node. However, by comparing the computational cost of each method given the same error threshold, the third-order method demonstrates to be the most efficient among all. Therefore, high-order methods are favourable options when the mesh is coarse or a high-level of accuracy is required.

5.2 Recommendations

Three different dissipation models are described in this thesis for use with the convective terms of the turbulence model. They are the first-order upwind scheme, the classical dissipation model, and the SBP dissipation model. In the investigation of the use of different approaches when numerically

solving the RANS equations in high-orders, it has been shown that the first-order upwind scheme is the most robust. However, the order of convergence is limited by the use of first-order dissipation model. Converged solutions are typically difficult to achieve using higher-order dissipation models and thus there is no sufficient data to analyze the efficiency of using the classical dissipation model and the SBP dissipation model. Further studies are recommended to improve the robustness of the SBP dissipation model in solving the RANS equations.

The algorithm used in the flow solver to compute steady-state flow solutions can be broken into two phases [37]: the start-up phase and the inexact-Newton phase. In the start-up phase, a modified Euler time-marching scheme is used to find a suitable initial iterate. In this phase, a first-order flow Jacobian matrix with reduced matrix entries, $\mathbf{A_1}$, is approximated from the full Jacobian matrix by eliminating the fourth-difference dissipation terms and modifying the coefficient for the second-difference dissipation. The first-order Jacobian, $\mathbf{A_1}$, is also used to construct the preconditioner with block incomplete lower-upper factorization of fill level p. The inexact-Newton phase uses the initial iterate achieved from the start-up phase and a different time step scheme to get a converged solution more quickly. In this phase, only the Jacobian-vector product is approximated for savings in storage. In this work, the Jacobian-vector product is approximated using the Frechet method [37]:

$$\mathbf{A^{(n)}_V} \approx \frac{\mathbf{R(Q^{(n)} + \epsilon V)} - \mathbf{R(Q^{(n)})}}{\epsilon}$$

where ϵ is determined with the number of unknowns and a perturbation parameter. In this phase, the approximate Jacobian, $\mathbf{A_1}$, is still used for preconditioning. Convergence issues have occurred in some cases when using the high-order methods for the RANS code. When using the high-order methods, the Jacobian-vector product approximation and Jacobian matrix used for the preconditioner still remain first order. A modified Jacobian matrix that is closer to the high-order right-hand side is more expensive to compute but it would be interesting to see if it will help with the convergency issues of the high-order methods. Meanwhile, further investigation of other preconditioning techniques would also be helpful.

The high-order SBP-SAT approach is also implemented to the negative SA turbulence model as described in Section 2.2.1. The negative model can be turned on when experiencing negative turbulence quantities in the process of solving the RANS equations. With very limited experience of negative turbulence quantities, this negative model is not fully tested in this thesis. More extensive tests of the negative model are recommended.

The turbulence model described in this thesis also includes a term that explicitly defines a trip location. In this thesis, this term is not used because all the test cases assume fully turbulent flows. It would be of great interest to see how the high-order methods will perform for flows with

laminar-turbulent transition prediction.

References

[1] R. K. Agarwal, "Review of technologies to achieve sustainable (green) aviation," *Recent Advances in Aircraft Technology*, vol. 19, pp. 427–464, 2012.

[2] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, and H. Huynh, "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, 2013.

[3] S. C. Dias and D. W. Zingg, "A high-order parallel Newton-Krylov flow solver for the Euler equations," in *19th AIAA Computational Fluid Dynamics*, 2009.

[4] S. De Rango and D. W. Zingg, "Higher-order spatial discretization for turbulent aerodynamic computations," *AIAA Journal*, vol. 39, no. 7, pp. 1296–1304, 2001.

[5] M. Osusky and D. W. Zingg, "Parallel Newton-Krylov-Schur flow solver for the Navier-Stokes equations," *AIAA Journal*, vol. 51, no. 12, pp. 2833–2851, 2013.

[6] I. Babuska and J. T. Oden, "Verification and validation in computational engineering and science: basic concepts," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 36, pp. 4057–4066, 2004.

[7] C. J. Roy, C. Nelson, T. Smith, and C. Ober, "Verification of Euler/Navier–Stokes codes using the method of manufactured solutions," *International Journal for Numerical Methods in Fluids*, vol. 44, no. 6, pp. 599–620, 2004.

[8] D. W. Zingg, S. De Rango, M. Nemec, and T. H. Pulliam, "Comparison of several spatial discretizations for the Navier–Stokes equations," *Journal of Computational Physics*, vol. 160, no. 2, pp. 683–704, 2000.

[9] J. A. Ekaterinaris, "High-order accurate, low numerical diffusion methods for aerodynamics," *Progress in Aerospace Sciences*, vol. 41, no. 3, pp. 192–300, 2005.

[10] M. J. Churchfield and G. A. Blaisdell, "Numerical simulations of a wingtip vortex in the near field," *Journal of Aircraft*, vol. 46, pp. 230–243, 2015/09/02 2009.

[11] J. Ekaterinaris, "Effects of spatial order of accuracy on the computation of vortical flowfields," *AIAA Journal*, vol. 32, no. 12, pp. 2471–2474, 1994.

[12] N. C. Nguyen, P.-O. Persson, and J. Peraire, "RANS solutions using high order discontinuous Galerkin methods," *AIAA Paper*, vol. 914, p. 2007, 2007.

[13] P. E. Morgan, D. P. Rizzetta, and M. R. Visbal, "High-order numerical simulation of turbulent flow over a wall-mounted hump," *AIAA Journal*, vol. 44, no. 2, pp. 239–251, 2006.

[14] C. Zhou and Z. J. Wang, "CPR high-order discretization of the RANS equations with the SA model," in *53rd AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, AIAA-2015-1286, 2015.

[15] D. Moro, N. Nguyen, and J. Peraire, "Navier–Stokes solution using hybridizable discontinuous Galerkin methods," in *20th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*, AIAA-2011-3407, 2011.

[16] S. R. Allmaras and F. T. Johnson, "Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model," in *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, pp. 1–11, 2012.

[17] T. A. Oliver and D. L. Darmofal, "Impact of turbulence model irregularity on high-order discretizations," in *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, AIAA-2009-953, 2009.

[18] P. G. Tucker, C. L. Rumsey, P. R. Spalart, R. B. Bartels, and R. T. Biedron, "Computations of wall distances based on differential equations," *AIAA Journal*, vol. 43, no. 3, pp. 539–549, 2005.

[19] J. A. Sethian *et al.*, "Level set methods and fast marching methods," *Journal of Computing and Information Technology*, vol. 11, no. 1, pp. 1–2, 2003.

[20] P. Tucker, "Differential equation-based wall distance computation for des and rans," *Journal of Computational Physics*, vol. 190, no. 1, pp. 229–248, 2003.

[21] E. Fares and W. Schröder, "A differential equation for approximate wall distance," *International Journal for Numerical Methods in Fluids*, vol. 39, no. 8, pp. 743–762, 2002.

[22] P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-92-0439, 2012.

[23] T. H. Pulliam and D. W. Zingg, *Fundamental algorithms in computational fluid dynamics*. Springer, 2014.

[24] B. Strand, "Summation by parts for finite difference approximations for d/dx," *Journal of Computational Physics*, vol. 110, no. 1, pp. 47–67, 1994.

[25] D. C. Del Rey Fernández, J. E. Hicken, and D. W. Zingg, "Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations," *Computers and Fluids*, vol. 95, pp. 171 – 196, 2014.

[26] S. C. Dias, "A high-order parallel Newton-Krylov flow solver for the Euler equations," Master's thesis, University of Toronto, 2009.

[27] N. A. L. Holt, "High-fidelity aerodynamic shape optimization with high-order spatial discretization," Master's thesis, University of Toronto, 2014.

[28] J. E. Hicken, "Efficient algorithms for future aircraft design: contributions to aerodynamic shape optimization," PhD thesis, University of Toronto, 2009.

[29] H.-O. Kreiss and G. Scherer, "Finite element and finite difference methods for hyperbolic partial differential equations," in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pp. 195–212, Academic Press, 1974.

[30] K. Mattsson, "Summation-by-parts operators for high order finite difference methods," *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology*, 2003.

[31] P. Diener, E. N. Dorband, E. Schnetter, and M. Tiglio, "Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions," *Journal of Scientific Computing*, vol. 32, no. 1, pp. 109–145, 2007.

[32] D. C. Del Rey Fernández and D. W. Zingg, "Generalized summation-by-parts operators for the second derivative," *SIAM Journal on Scientific Computing*, vol. 37, no. 6, pp. A2840–A2864, 2015.

[33] K. Mattsson, M. Svärd, and M. Shoeybi, "Stable and accurate schemes for the compressible Navier–Stokes equations," *Journal of Computational Physics*, vol. 227, no. 4, pp. 2293–2316, 2008.

[34] K. Mattsson, "Boundary procedures for summation-by-parts operators," *Journal of Scientific Computing*, vol. 18, no. 1, pp. 133–153, 2003.

[35] M. Svärd and J. Nordström, "A stable high-order finite difference scheme for the compressible Navier–Stokes equations: no-slip wall boundary conditions," *Journal of Computational Physics*, vol. 227, no. 10, pp. 4805–4824, 2008.

[36] M. Svärd, M. H. Carpenter, and J. Nordström, "A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions," *Journal of Computational Physics*, vol. 225, no. 1, pp. 1020–1038, 2007.

[37] J. E. Hicken and D. W. Zingg, "Parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms," *AIAA Journal*, vol. 46, no. 11, pp. 2773–2786, 2008.

[38] M. Osusky, "A parallel Newton-Krylov-Shur algorithm for the Reynolds-averaged Navier-Stokes equations," PhD thesis, University of Toronto, 2013.

[39] J. Nordström, J. Gong, E. Van der Weide, and M. Svärd, "A stable and conservative high order multi-block method for the compressible Navier–Stokes equations," *Journal of Computational Physics*, vol. 228, no. 24, pp. 9020–9035, 2009.

[40] P. R. Spalart and C. L. Rumsey, "Effective inflow conditions for turbulence models in aerodynamic calculations," *AIAA Journal*, vol. 45, no. 10, pp. 2544–2553, 2007.

Appendices

Appendix A

Special Boundary Conditions

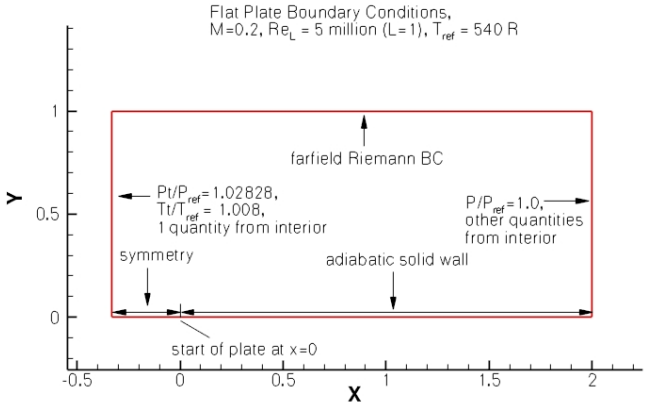

Figure A.1: The boundary conditions for Case 1, 2D zero pressure gradient flat plate, provided on the TMR website

The validation cases for the turbulence model require special boundary treatment to accommodate inflow and outflow boundary conditions. The boundary conditions for the 2D flat plate with zero pressure gradient and 2D bump-in-channel cases are summarized in Figure A.1 and Figure A.2, respectively. At inflow and outflow boundaries, the residual is calculated as

$$R = J^{-1}(\mathbf{Q} - \mathbf{Q_{targ}})$$

where $\mathbf{Q}$ contains local flow variables and $\mathbf{Q_{targ}}$ contains targeted flow variables due to boundary conditions. J is the metric Jacobian.

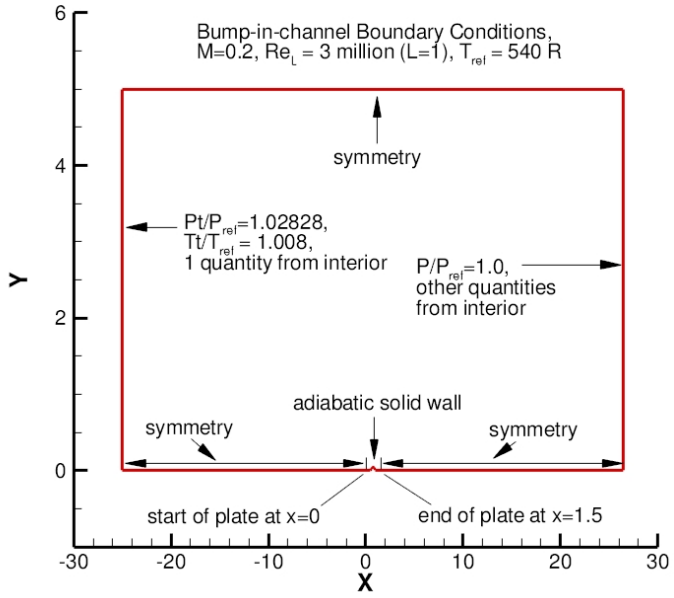

Figure A.2: The boundary conditions for Case 1, 2D bump-in-channel, provided on the TMR website

A.1 Inflow boundary

The first Riemann invariant, R_1 remains constant between the boundary node with a j index and the adjacent node with a $j+1$ index.

$$R_{1,j} = V_{n,j} - \frac{2a_j}{\gamma - 1} = R_{1,j+1} = V_{n,j+1} - \frac{2a_{j+1}}{\gamma - 1}$$

where V_n is the velocity normal to the boundary, which we can substitute with velocity component u. The speed of sound in normalized form can be calculated as:

$$a = \sqrt{T}.$$

The tangential velocity is set to zero in this case. With the local values from the adjacent node, u_j can be calculated using the Riemann equation expressed above. Pressure and density values can be calculated using the following equations:

$$p = \frac{P_t}{1 + \frac{1}{2}\gamma M^2)^{\frac{\gamma}{\gamma-1}}},$$

$$\rho = \frac{\gamma p}{a^2}.$$

The turbulence quantity target is set to the freestream value of $\tilde{\nu}$, which is 3.0 in this case.

A.2 Outflow boundary

At the outflow boundary node with a j index, we have the other two Riemann invariants from the adjacent node with a $j-1$ index:

$$R_{2,j-1} = u_{j+1} + \frac{2a_{j-1}}{\gamma - 1},$$

$$R_{3,j-1} = \frac{p_{j-1}}{\rho_{j-1}^{\gamma}}.$$

The tangential velocity, v, is taken as the value of the adjacent node and the normal velocity, u, can be calculated using constant R_2 at the boundary node and the boundary adjacent node. Density is calculated using:

$$\rho = (\frac{p}{R_3})^{\frac{1}{\gamma}}.$$

The turbulence quantity target is set to be same as the local value.