2018

Optimization of Turbulent Prandtl Number in Turbulent, Wall Bounded Flows

Donald Edward Bernard
University of Vermont

Follow this and additional works at: https://scholarworks.uvm.edu/graddis
Part of the Aerodynamics and Fluid Mechanics Commons, and the Thermodynamics Commons

Recommended Citation
https://scholarworks.uvm.edu/graddis/824

This Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks @ UVM. It has been accepted for inclusion in Graduate College Dissertations and Theses by an authorized administrator of ScholarWorks @ UVM. For more information, please contact donna.omalley@uvm.edu.
OPTIMIZATION OF TURBULENT PRANDTL NUMBER IN TURBULENT, WALL-BOUNDED FLOW

A Thesis Presented

by

Donald E. Bernard

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Mechanical Engineering

January, 2018

Defense Date: November 10, 2017
Thesis Examination Committee:

Yves Dubief, Ph.D., Advisor
Hamid Osserah, Ph.D., Chairperson
Douglas Fletcher, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College
ABSTRACT

After nearly 50 years of development, Computational Fluid Dynamics (CFD) has become an indispensable component of research, forecasting, design, prototyping and testing for a very broad spectrum of fields including geophysics, and most engineering fields (mechanical, aerospace, biomedical, chemical and civil engineering). The fastest and most affordable CFD approach, called Reynolds-Average-Navier-Stokes (RANS) can predict the drag around a car in just a few minutes of simulation. This feat is possible thanks to simplifying assumptions, semi-empirical models and empirical models that render the flow governing equations solvable at low computational costs. The fidelity of RANS model is good to excellent for the prediction of flow rate in pipes or ducts, drag, and lift of solid objects in Newtonian flows (e.g. air, water). RANS solutions for the prediction of scalar (e.g. temperature, pollutants, combustable chemical species) transport do not generally achieve the same level of fidelity. The main culprit is an assumption, called Reynolds analogy, which assumes analogy between the transport of momentum and scalar. This assumption is found to be somewhat valid in simple flows but fails for flows in complex geometries and/or in complex fluids.

This research explores optimization methods to improve upon existing RANS models for scalar transport. Using high fidelity direct numerical simulations (numerical solutions in time and space of the exact transport equations), the most common RANS model is a-priori tested and investigated for the transport of temperature (as a passive scalar) in a turbulent channel flow. This one constant model is then modified to improve the prediction of the temperature distribution profile and the wall heat flux. The resulting modifications provide insights in the model’s missing physics and opens new areas of investigation for the improvement of the modeling of turbulent scalar transport.
ACKNOWLEDGEMENTS

I would like to thank Dr. Yves Dubief for his continued support and immense effort on this project. His research was the founding of this idea and his uplifting spirit helped shape this project.

To my family and friends I owe a sincere thank you for their support, encouragement, and dedication to push me.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Motivation</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Methods</td>
<td>9</td>
</tr>
<tr>
<td>1.4</td>
<td>Opportunities in Predicting Turbulence</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>CHAPTER 2: MODELING TECHNIQUES</td>
<td>13</td>
</tr>
<tr>
<td>2.1</td>
<td>RANS Definition</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>k-ε Model</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>k-ω Model</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2.3.1 SST Model</td>
<td>20</td>
</tr>
<tr>
<td>2.4</td>
<td>Spalart-Allmaras Model</td>
<td>21</td>
</tr>
<tr>
<td>2.5</td>
<td>v2-f Model</td>
<td>23</td>
</tr>
<tr>
<td>2.6</td>
<td>Convex Optimization</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>CHAPTER 3: ADJOINT METHOD</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>Motivation</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>Derivation of Adjoint Method</td>
<td>28</td>
</tr>
</tbody>
</table>
3.3 The Inverse Function

3.3.1 Derivation of Inverse Equations

3.4 The Tridiagonal Matrix Algorithm

CHAPTER 4: FINDINGS

4.1 Initial Attempts

4.2 Coding

4.2.1 Setup

4.2.2 Optimization Coding

4.2.3 General Solutions

4.2.3 Oscillating Flow Setup

4.2.4 Oscillating Flow Solutions

4.3 Verification

4.4 Results and Future Work

Bibliography
LIST OF FIGURES

Figure Page

Figure 1: TDMA Style Matrix Equation.................................................................34
Figure 2: TDMA Solver for temperature equations............................................37
Figure 3: Velocity Profile from simulated data...................................................38
Figure 4: Temperature Profile from simulated data............................................39
Figure 5: Viscosity Profile from simulated data..................................................39
Figure 6: Temperature Profile comparison, with calculated $\alpha$.....................40
Figure 7: Temperature optimization coding example........................................42
Figure 8: Tabulated Values of Turbulent Prandtl Number...............................44
Figure 9: Prandtl number per position in channel............................................45
Figure 10: First Optimized Prt Value Comparison..........................................46
Figure 11: Optimization of Prt With Respect to Wall......................................47
Figure 12: Nelder-Mead Simplex Solution........................................................48
Figure 13: Nelder-Mead Optimized Temperature Profile..................................49
Figure 14: Pulsatile mean velocity profiles .................................................................50

Figure 15: Temperature profiles by phase (normalized) ..........................................51

Figure 16: Turbulent Prandtl number Sensitivity profiles by phase .....................52

Figure 17: Prt Normalized - Macro View .................................................................53

Figure 18: Prt Normalized: Zoom View ..................................................................53

Figure 19: Graphical damping functions ...............................................................54

Figure 20: Momentum profiles by phase normalized to wall .............................55

Figure 21: Diffusion profiles by phase normalized to wall ..................................56

Figure 22: Heat Flux comparison by phase with constant Prt .............................57

Figure 23: First attempt at optimization of Prt - heat flux by phase .................58

Figure 24: Second attempt at optimization of Prt - heat flux by phase ............58
CHAPTER 1: INTRODUCTION

1.1. Overview

The simulation of turbulent flows has become an integral component of design, prototyping, and testing in virtually all engineering fields, from cooling of computer chips, to combustion, to atmospheric entry space vehicles. Fluid dynamics is governed by the Navier-Stokes equations, a complex set of nonlinear partial differential equations that require immense computational power to solve accurately. The range of scales occurring and driving turbulence is the reason for this cost. In a turbulent flow of velocity fluctuation scale $u'$ and integral scale $L$, the ratio between the largest (integral) scale and the Kolmogorov scale (the smallest dissipative scale of turbulence) is

$$\frac{L}{\eta_K} = Re^{3/4} = \left( \frac{u'L}{\nu} \right)^{3/4},$$

(1)

where $\nu$ is the kinematic viscosity of the fluid. Consequently, the resolution of all turbulent scales, a necessity for the highest fidelity prediction of turbulent flow simulation, requires $Re^{9/4}$ degrees of freedom. Current supercomputers are only powerful enough for the high fidelity direct numerical simulation (DNS) of moderate Reynolds number flows $O(10^4) - O(10^5)$, whereas most engineering applications are high Reynolds numbers $\gtrsim O(10^6)$.

More affordable simulations can be performed on smaller, high performance computing clusters or workstations, by seeking the solution of the mean flow field only. Such simulations, called Reynolds Average Navier-Stokes (RANS) are based on the time-average Navier-Stokes equations but simplified by assumptions and empirical
models of the dynamics of turbulence. The resulting computational saving is orders of magnitude smaller than that of a DNS, enabling, for instance, rapid iterative processes in designs.

The most widely used transport model is the standard $k - \varepsilon$ equation developed by Launder and Spalding in 1964. [18] This model is accurate and efficient at high levels, and has remained mostly unchanged since its inception. Its name reflects the two drivers of turbulence. The turbulent kinetic energy (TKE), $k$, of the velocity fluctuations is a local measure of the intensity of turbulence which is a function of the forcing of the flow, the fluid viscosity, $\nu$, and the boundary conditions. TKE is driven by the large scales of the flow. Like any physical system, the energy input, here resulting in TKE, must be balanced by a dissipation process. In turbulence, The Kolmogorov cascade [22] describes such a balance as a cascade of energy from the largest scales to a finite, small scale, the Kolmogorov scale $\eta_K$. The energy transfer between scales of decreasing size is defined by the TKE dissipation rate, $\varepsilon$, and the Kolmogorov scale, defined as $\eta_K = (\nu^3/\varepsilon)^{1/4}$. This is the scale at which the energy is dissipated in heat through molecular friction. In the $k - \varepsilon$ turbulence model, the model of the turbulence dissipative process of the mean flow is derived from simple dimensional arguments involving $k$ and $\varepsilon$, simplified transport equations for these very variables, and a number of empirical constants determined to match canonical turbulent flows. Here canonical turbulent flows define the simplest flows in which turbulence occurs and for which extensive theoretical understanding and statical characterization exist. Such flows include fully developed turbulent channel flow occurring between
two parallel, flat, hydrodynamically smooth plates of infinite dimensions, i.e. with negligible side effects. Under such conditions, the flow is said to be at equilibrium.

It is therefore not surprising that RANS models perform well in equilibrium canonical flows. The fidelity of RANS solutions degrades in non-equilibrium canonical and non-canonical flows, at equilibrium or not, with an uncertainty expanding with increasing departure from equilibrium and/or canonical boundary conditions. Improving the fidelity of RANS in non-equilibrium/non canonical flows has been a long standing focus of turbulence research but has gained renewed interest in recent years driven by the greater affordability and therefore use of CFD in a wide range of industries. The task of improving the fidelity of RANS is daunting. As mentioned earlier, improvements, with a few exceptions, have been incremental at best. The traditional approach is to include more physics in the model, but rarely our understanding of the missing physics of a particular model is so complete and straightforward that an accurate new model or transport equation may be derived. One of the rare success story is the $v^2 - f$ model. In the $k - \varepsilon$ model the contribution of turbulence to the mean flow field was found to be too dissipative in the near wall region of wall-bounded turbulence, resulting in large uncertainties and even the inability to simulate turbulence under certain conditions. The workaround was the development of low-Reynolds number models by deriving ad-hoc empirical damping functions of the turbulent models close to the wall. These functions were developed from canonical flows and are a function of a distance from the wall. The latter constraint can be cumbersome to implement for complex geometries. The former carries the uncertainty
that scaling laws valid in equilibrium canonical flows rarely apply to non-equilibrium/non-canonical flows. In the $v^2-f$ model, the wall-normal velocity is a new transport variable of the model and the contribution of turbulence to the mean flow field is a function of $v^2$. The rapid decay of wall-normal velocity fluctuations close to the wall acts as a wall-damping function, yet without any ad-hoc exponential function of the distance from the wall. The model was derived from the understanding of the dynamics of near-wall turbulence and in particular the dynamics of coherent structures.

This manuscript reports some progress on another weakness of turbulence modeling: The Reynolds analogy. The Reynolds analogy states that a scalar transported by a turbulent flow mixes on the same time and length scales as the turbulent mixing of momentum (or contribution of turbulence to the mean velocity field). Here a scalar could be temperature, the concentration of chemical specie, or particles. To better illustrate the Reynolds analogy, consider that the mixing of momentum by turbulence is modeled, in RANS, as a viscosity, a turbulent viscosity to be precise $\nu_t$, multiplied by the mean strain. The turbulent mixing of scalar is modeled as a turbulent diffusivity $\alpha_t$ multiplied by the mean temperature gradient. The Reynolds analogy assumes that proportionality

$$\alpha_t = \frac{\nu_t}{Pr_t} \quad (2)$$

where the turbulent Prandtl number $Pr_t$ is a constant. As stated by Bradshaw, there is no particular rigorous reason why the Reynolds analogy should work, and indeed it works in canonical flows. However it is well known to fail in non-
equilibrium/non-canonical flows, to the point that the turbulence community has very little faith in the predictions of scalar transport using RANS simulations.

The motivation of this work is to investigate possible ways to improve upon the modeling of passive scalar by using the Reynolds analogy (Eq. 2) as a basis, but with a variable $Pr_t$ in space for steady state flows, and in time and space for unsteady flows. In this work, the turbulent viscosity $\nu_t$ is always known. The variability, either in space or time or both, will be introduced by (i) extraction of the optimal behavior of $Pr_t$ and (ii) optimization of ad-hoc functions or models that optimally mimic the behavior of the optimal $Pr_t$. The goal is not to derive a universal model for high fidelity prediction of scalar transport; the quasi-stagnant evolution of its model suggests that many more years of research might be needed. The immediate goal is to use optimal solutions to identify the dynamics of turbulence that causes departure from the Reynolds Analogy. In most RANS the recommended turbulent Prandtl number is \(~0.9-1.0\), therefore assuming a quasi-equality between the strength of momentum mixing and scalar mixing. Yet in canonical flows, there are regions where $Pr_t$ is significantly higher or lower than 1. When $Pr_t > 1$, the scalar mixes less than momentum and vice versa for $Pr_t < 1$. Our study will therefore seek to identify regions of the flow where the turbulent Prandtl number departs significantly from unity, and assess the impact of the modeling, or lack thereof, of these departures on the RANS solutions in order to derive the foundations or constraints that a high fidelity model of scalar transport should meet.
The present study first revisits the canonical channel flow case with temperature transported as a passive scalar\(^1\). The Reynolds analogy mostly works in this flow but not exactly, especially close to the wall. It is therefore important to ascertain the impact of the departure near the wall and gain an understanding, then explore an oscillatory channel which was the basis of a previous study on the assessment of heat transfer modeling in non-equilibrium flows [1].

### 1.2 Motivation

Current computational fluid dynamics (CFD) software rely on the estimation of variables to compute desired flows. This system is effective in simple flows, but fails when heat flux, oscillatory flows, or non-linear boundaries occur. The k-\(\varepsilon\) model is the most widely used form of determining the bulk velocity and turbulent viscosity, but other methods were devised in order to account for the opportunities, such as the k-\(\omega\), Spalart-Allmaras, and \(\nu^2\)-f. Each of these methods is described below and defined to show where their opportunities lie. This thesis will call out the significant variables in the k-\(\varepsilon\) and Wilcox k-\(\omega\) equations and optimize them to best fit direct numerical simulation (DNS) data of the same flow. This system has been in use nearly since the inception of these equations, as it is best to validate each of the results to a known experiment. This is yet another task that will validate the optimized field of variables, but with continued success can be left out of the process.

It cannot be assumed that any of the mathematics that derive the optimized fields are any better than the minds that have devised the original equations. With this

---

\(^1\) Temperature is a passive scalar, i.e. its transport by the flow does not affect the flow, when the maximum temperature difference is small enough that the buoyancy force is negligible.
consideration the optimization is an attempt to better utilize given systems and
equations and improve upon the opportunities in existing models. The basis of their
research works due its simplicity, but can be improved with updated technology. None
of the variables will change and each of the system of equations will remain intact, but
the goal is to optimize them over time and space to improve the strength of the models
in computational fluid dynamics.

The focus of the project is the prediction of scalar transport in turbulent flows. A
scalar could be temperature, or a chemical specie. In RANS, the turbulent mixing of
scalars employs the Reynolds Analogy, which assumes that the mixing of a scalar
operates on the same length and time scales as the mixing of momentum. This
encompasses the mixing length model, which utilizes a generalized theory of transport
for most parameters in the flow. In this way the analogy assumes that the shear stresses
of the wall in the transition layers, such as the viscous layer or log region, have little
effect on the bulk flow. Since the viscous layer is usually regarded as small, the changes
in scalar such as turbulent Prandtl number or temperature are disregarded as
insignificant. However, in flows with rapid changes, or large stream-wise gradients, the
Reynolds Analogy fails in that it cannot predict the fluctuations caused by the
temperature differential or flow changes in the viscous region. As discussed by
Bradshaw in his paper “The Law of the Wall in Turbulent Flow” the bulk Reynolds
number holds high significance in determination of the production of flow and even
goes to say: “This raises the question 'Is the tenacity of the U-law just good luck, and if
so when does our luck run out?'.” [3]
The turbulent Prandtl number is the focused scalar of the project, as it has been discovered that the effect diffusion has on the boundary layer of flow across a surface is drastic. The Pr is the ratio of momentum diffusivity to thermal diffusivity,

$$Pr_t = \frac{\nu_T}{\alpha_T},$$  \hspace{2cm} (3)  

from the equation:

$$Pr_t = \frac{\bar{u}'\bar{\theta}' \, d\bar{\theta}/d\bar{y}}{\bar{\theta}' \, d\bar{u}/d\bar{y}},$$ \hspace{2cm} (4)  

which will change the separation of the boundary layer depending on the ratio varying above or below unity. If the Pr$_t > 1$, this indicates that the momentum diffusivity is greater than the thermal diffusivity. This will allow a greater distance from the wall on boundary layer separation, as the heat flux will not be able to overcome the turbulent mixing properties of the flow. If the Pr$_t < 1$ the thermal diffusivity is greater than the momentum diffusivity, thus leading to a flow that is much more energetic than expected. This is where the Reynolds analogy breaks down as the thermal diffusivity is expected to act on the same time scale, but does not act as the Kolmogorov Scale expects. With the admission of the much greater turbulent energy dissipation it follows that the increase in energy affects the flow.

This question has gone unnoticed or unanswered previously, but has great potential in better explaining the physics of turbulent flow. It is for this reason that this thesis will attempt to create a framework to explore the effect the turbulent Prandtl
number has on turbulent flow and lead to higher fidelity computations of fluid dynamics.

1.3 Methods

In order to model with techniques such as the $k$-$\varepsilon$ equation, the incorporation of a diffusivity term and averaged bulk flow must be utilized. The diffusivity accounts for the energy siphon in the turbulence and the bulk flow can be easily averaged with little error as it maintains a steady state more easily. These included terms and assumptions, however, result in the previously outlined failures when the flow is non-linear. For example, the Reynold’s Analogy assumes that heat flux in a turbulent model is analogous to momentum flux. Yet this does not apply if drag is present, or if the fluid in question is not a gas, or again when the gradient of temperature is low; thus it is not strong enough to accurately compute the thermal diffusivity in a channel flow. [7]

Recent advances in cloud computing software, faster computer processing, and new open-source programming have made it possible to improve upon the given modeling techniques. Knowing the pitfalls of the $k$-$\varepsilon$ equation has helped shape newer, more effective two-equation methods. However, despite knowing the models lack the power to compute precisely, the methods have not changed. This research will determine what can be done different to further improve the accuracy of existing RANS models by examining the significant variables in each set of equations while comparing them to direct numerical simulation (DNS) of the same flow.

Using the optimization library, CVXOPT, from Stanford University and sensitivity analysis in Python, the goal is to have each variable examined to show its
value to the flow and then optimized accordingly to best predict the flow. [2] Doing so will allow any future modeling to be efficient and effective at determining a likely (closest to DNS) value for each variable with given constants. The known results of the k-\(\varepsilon\), \(v^2\)-f, and Spalart-Allmaras turbulence models on a prescribed flow will serve as the basis of the research data, and Python coding will be used to simulate all models and solve the optimization equations. Optimizing each variable will not create an exact solution, but will reduce the error between the DNS and the k-\(\varepsilon\), k-\(\omega\), and \(v^2\)-f models. [2]

After an initial attempt using brute-force methods of testing, an error value between DNS and the bulk velocity of a turbulence model, with \(Re_\tau = 137\), can be reduced from 17% to 3%. This was simply achieved via sensitivity analysis and compared to the DNS for the same flow. An attempt was then made at deriving the correct equations to be used in CVXOPT, but it was ultimately determined that linear regression or an adjoint method could be used to more efficiently optimize the given variables. The only variables currently considered in this thesis are the turbulent viscosity, Prandtl number, and the temperature gradient.

If permissible, a second objective of the project will be to model this with turbulent flows as well as oscillating flows in a channel. Then when modelers wish to see mathematical values or models of flow, the optimization matrix can account for specific time snapshots within the simulation and give accurate data. Within the oscillatory flow the optimization should help reduce the number of constants from five
to three by finding optimized values and setting them equal to each other through interdependent equations.

Both of these techniques will utilize a form of a cost function to optimize the bulk velocity and the error from the DNS to the model. When complete, this project will be the beginning of research to more effectively allow the use of the k-ε model, k-ω model, the Spalart-Allmaras model, and the $\nu^2$-f model to accurately predict turbulent Prandtl number and turbulent viscosity in flow regimes while accounting for heat flux, turbulent boundary layers, and thermal diffusivity within a channel where they previously could not. It uses existing modeling techniques and current versions of open-source coding to be accessible to all prospectives interested in modeling wall bounded flows. Since each model uses $A^+$ as a variable with respect to the distance from the wall, the scale is irrelevant; it can be used from large duct flow to capillary flows.

### 1.4 Opportunities in Predicting Turbulence

The most difficult part of calculating turbulence is its inherent unpredictability at transition, and its non-linearity. This requires more constitutive equations and higher level correlations to calculate; hence it is more difficult to simulate. Most programs utilize an averaged state of turbulence, hence Reynold’s Averaged Navier-Stokes, and are still inaccurate near the wall. Turbulence occurs when a flow is increasing in velocity and the fluid no longer remains in a steady state, smooth, flow. Turbulence is still loosely defined, but is easily characterized by its ability to mix fluids and dissipate kinetic energy. Turbulent eddies begin to siphon energy from the laminar flow (this is
the transition region) and increase the velocity, shear rate, and kinetic energy. More specifically, stream-wise vortices extract energy from the mean flow and produce alternating streaks of stream-wise velocity, which in turn produce the quasi-stream-wise vortices. [7]

This mechanism is strongest in near-wall flow. Since this cycle is recursive, there exists a limit that the flow reaches through entropic expansion called the Kolmogorov vortex cascade limit. This limit was postulated and later proven by Andrey Kolmogorov in 1941 that all turbulent eddies separate equally and in inviscid flow until the kinetic energy is simply dissipated by the smallest eddies via viscous action into heat. [1] The largest eddies are unstable and break down into smaller and smaller eddies, passing off energy to the smaller eddies when they do so. The limit of this energy cascade is reached when the Re$_0$ is sufficiently small so that the flow is again stable, and therefore transfers energy off as heat. The importance of this dissipation is that in the first step $\varepsilon$, the diffusion term, is determined as the rate of the transfer from the largest eddies. Each of the smallest eddies is locally isotropic due to the near molecular size of the step, despite turbulent flow being always anisotropic. The scales of energy dissipation are of particular interest in this research, as the rate is determined by the viscosity, which will be optimized for each locally isotropic point.
CHAPTER 2: MODELING TECHNIQUES

2.1. RANS Definition

In all forms of modeling fluid flow, turbulent or laminar, the continuity, momentum, and energy dissipation equations are the basis of all calculations. These comprise the Navier-Stokes equations, specifically momentum conservation (Eq. 5), mass conservation (Eq. 6), and the energy equation (Eq. 7); these are shown respectively below. [7]

\[ \partial_t \tilde{u}_i + \tilde{u}_j \partial_j \tilde{u}_i = -\frac{1}{\rho} \partial_i \tilde{p} + \nu \nabla^2 \tilde{u}_i, \]  (5)

\[ \partial_i \tilde{u}_i = 0 \]  (6)

\[ \frac{\partial}{\partial t} (\rho e_0) + \frac{\partial}{\partial x_j} (\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}) = 0. \]  (7)

The velocity component is then deconstructed into its mean and fluctuating components in the form of \( \tilde{u} = U + u \) and substituted back into equations (Eq. 5) and (Eq. 6). From this we obtain the set of equations:

\[ \partial_t (U + u)_i + (U + u)_j \partial_j (U + u)_i = -\frac{1}{\rho} \partial_i (P + p) + \nu \nabla^2 (U + u) \]  (8)

\[ \partial_i (U + u)_i = 0 \]  (9)

For the averaged equations the consideration must be that \( \bar{U} = U \), and \( \bar{u} = 0 \) so we then obtain the governing RANS equations in the form of:

\[ \partial_t U_i + U_j \partial_j U_i = -\frac{1}{\rho} \partial_i P + \nu \nabla^2 U_i - \partial_j \bar{u}_j \bar{u}_i \]  (10)

\[ \partial_i U_i = 0 \]  (11)
The Navier-Stokes equations are not closed, and are in fact often five equations short of solving for all the variables, so they can only be used as a basis to solve more complicated regimes. Reynolds Average Navier-Stokes is a process of averaging these above equations for a given point in the flow. The pressure and velocity are assumed constant for said step in space/time. This means: \[
\frac{du}{dx} = 0,
\] or the derivative of velocity with respect to the point in the flow, is set to zero; this is continuity. The momentum equation and energy equation are then averaged as shown below (Eq. 12), (Eq. 13), which leads to the Reynolds Stress term, the last section of equation (Eq. 10), which is then closed via the Boussinesq method. [26]

\[
\frac{Du_i}{Dt} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial (u_i u_j)}{\partial x_j} \tag{12}
\]

\[
\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = \frac{-1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} - \frac{\partial (u'_i u'_j)}{\partial x_j} \tag{13}
\]

Or,

\[
\partial_i u_i u_j + U_k \partial_k u_i u_j = -\frac{1}{\rho} (u_j \partial_i p + u_i \partial_j p) - 2\nu \partial_k u_i \partial_k u_j
\]

\[-\partial_k u_i u_j u_k - u_j u_k \partial_k U_i - u_i u_k \partial_k U_j + \nu \nabla^2 u_i u_j \tag{14}\]

Where equation (Eq. 14) is the Reynolds Stress transport equation. This is the equation that requires the use of the methods described below, such as k-ε, to close the system and solve for the unknown variables. Since RANS modeling is an averaged
value per various steps in the flow, it cannot predict changes well, and therefore is weaker at the transition point to turbulence, or where a change in the flow occurs such as a bend in a pipe flow.

Turbulent transition areas are difficult to simulate due to non-linearity. Each of the RANS equations sets the slope at the current point to zero, which does not allow for easy prediction when the flow shifts. Hence, in computational fluid dynamics (CFD), the systems are taken at various time steps so as to account for the transition. This is highly taxing on CPU time, however, and this research hopes to reduce that with optimization of the variables used.

The RANS equations are based off the Reynold’s stress, $\tau$, which has a linear relationship to the velocity gradient as postulated by Newton. [7] This relationship is, in part, what helps close the system of equations and provides the building blocks for the $k$-$\varepsilon$ equations and $k$-$\omega$ equations. Each of these, however, is still an averaged value and thus the calculations incur error through the minimum and maximum values per point. This is also why all current models are based off laminar flow and easily produced turbulent models such as high speed flow in a duct.

### 2.2 $k$-$\varepsilon$ Model

As a widely used method of fluid dynamic modeling, the $k$-$\varepsilon$ equations serve as a basis for most other models. This technique focuses on the turbulent kinetic energy and a dissipation rate to determine the turbulent viscosity. As a two-equation model it determines the energy, $k$, and the scale, $\varepsilon$, of the turbulence (i.e. smoothing). The now “standard” $k$-$\varepsilon$ model was put forth by Launder and Jones in 1972, improving upon the
mixing length model and outlining the strategy for prescribing turbulent length scales in highly complex flows. Launder and Sharma then introduced the empirical constants in 1974 that are used in the standard model. Despite not being as robust or possibly as accurate as later models, the $k$-$\varepsilon$ is used as a prototype for other models and is thus the basis of this research as well. [22]

The model begins with the transport equations:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k + P_b - \rho \varepsilon - Y_M + S_k$$

(15)

and

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] + C_{1\varepsilon} \frac{\varepsilon}{k} \left( P_k + C_{3\varepsilon} P_b \right) - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} + S_\varepsilon$$

(16)

for kinetic energy and dissipation, respectively. The constants in the equations are as follows: $C_{1\varepsilon} = 1.44$, $C_{2\varepsilon} = 1.92$, $C_\nu = 0.09$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$. [22] From here the turbulent viscosity can be modeled as such:

$$\nu_t = \rho \frac{C_\nu k^2}{\varepsilon}$$

(17)
The density in our problem is assumed to be unity, thus simplifying our equations further. The production of kinetic energy, $P_k$, is related to the mean rate of strain tensor via the Boussinesq Approximation, $S$, and is shown as:

\[
P_k = -\rho \overline{u'_i u'_j} \frac{\partial u_j}{\partial x_i}
\]  
(18)

\[
P_k = \nu_i S^2
\]  
(19)

\[
S \equiv \sqrt{2S_{ij}S_{ij}}
\]  
(20)

The purpose of the k-ε model, and any that follow, are to predict eddy viscosities. This is shown through the connection of turbulent kinetic energy and its dissipation from the above equations. However, at high Reynolds numbers the rate of dissipation and production are of similar magnitude and can be estimated as $\varepsilon \approx P$. [22] The formula for the turbulent viscosity then becomes, in our case:

\[
\nu_T = \frac{C_v k^2}{\varepsilon},
\]  
(21)

with the same $C_v$ value of 0.09. [22] Since turbulent flow is anisotropic, the constant values cannot easily be measured. They can be proven, however, by isolating each term in isotropic, homogeneous flow. This has been the practice to verify values used in these equations and will continue to be the best method to verify our work.

As the basis of each of the following models it is important to outline the specific pitfalls of the k-ε equations. The representation of mixing suppression is critical to accurate skin friction and heat transfer predictions, however this model fails catastrophically below the log-layer and thus cannot predict these values. As mentioned
in section one, the transfer and boundary layers are important to the model as it is the 
region that energy transfer occurs. The log-layer is the region above the buffer layer 
that is unaffected by the flow boundaries. Since the viscous layer and buffer region 
vary depending on wall shear, the log-layer is assumed from a log-law variation of the 
shear stresses in the viscous sublayer that ignores the bounding stresses. Our system 
also assumes a high Reynold’s Number which incorporates Kolmogorov scaling as a 
surface scaling. The equation for this scaling of the time scale then becomes:

$$T = \sqrt{\frac{\nu}{\varepsilon} F \left( \sqrt{\frac{k^2}{\nu \varepsilon}} \right)}.$$ (22)

**2.3 k-ω Model**

The k-ω equations were first posited by Kolmogorov, but in 1998 Wilcox 
created a system of equations to be used in the now accepted method. [22] The 
turbulent viscosity is written as:

$$\nu_T = \frac{k}{\omega},$$ (23)

where k is the kinetic energy and ω is the specific dissipation. In this model the 
transport equations become:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = \tau_{ij} \frac{\partial U_i}{\partial x_j} - \beta^* k \omega + \frac{\partial}{\partial x_j} \left[ (\nu + \sigma^* \nu_T) \frac{\partial k}{\partial x_j} \right],$$ (24)

and
for turbulent kinetic energy and specific dissipation, respectively. The constants become $\alpha = 0.556$, $\beta = 0.075$, $\beta^* = 0.09$, $\sigma = 0.5$, $\sigma^* = 0.5$, and $\varepsilon = \beta^* \omega k$. [22]

This model is unique in that it is usable near wall boundaries. [7] This is because it can be computed without a requirement for wall functions and ignores wall damping; this is due to the extra dissipation produced in near wall flow. Once the $\omega$ equation is rewritten as an $\varepsilon$ equation, this distinction becomes clear. [7] Since the rate of dissipation in this set of equations is $k\omega = \varepsilon$, the evolution of dissipation is then shown as:

$$\frac{\partial \varepsilon}{\partial t} + U_j \frac{\partial \varepsilon}{\partial x_j} = \alpha \frac{\omega}{k} \tau_{ij} \frac{\partial U_i}{\partial x_j} - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[ (\nu + \nu_T) \frac{\partial \omega}{\partial x_j} \right], \quad (25)$$

with

$$\frac{\partial \varepsilon}{\partial t} + U_j \frac{\partial \varepsilon}{\partial x_j} = \frac{C_{\epsilon 1} P - C_{\epsilon 2} \varepsilon}{T} + \partial_j \left( \left( \nu + \nu_T \right) \frac{\partial \varepsilon}{\partial x_j} \right) + S_{\omega}, \quad (26)$$

with $T = \frac{1}{\omega}$, $C_{\epsilon 1} = 1 + C_{\omega 1}$, $C_{\epsilon 2} = 1 + C_{\omega 2}$, and $\sigma_{\varepsilon} = \sigma_{\omega}$. [7] This is a reproduction of the standard k-$\varepsilon$ model with the newly added $S_{\omega}$ term. This source term is:

$$S_{\omega} = \frac{2}{T} \left( \nu + \nu_T \right) \left[ \frac{|\nabla k|^2}{k} - \frac{\nabla k \cdot \nabla \varepsilon}{\varepsilon} \right], \quad (27)$$

where $S_{\omega} > 0$. Since the viscous sublayer increases kinetic energy with wall distance and the dissipation decreases, this extra $S_{\omega}$ term is a source of dissipation, thus a larger $\varepsilon$ is produced. [7]
Two notable failings of the k-ω model were pointed out by Menter in that it both over predicts the level of shear stress in adverse pressure gradient boundary layers and its spurious sensitivity to free-stream conditions. [7] This means this model is not reliable in flows with detached boundary layers. These can be overcome, however, with the use of the SST - shear stress transport, model.

2.3.1 SST Model

The shear stress transport model utilizes a blending function to tackle the pitfalls of the k-ω and k-ε disparities. The first step improves the predictions in adverse pressure gradient boundary layers, and the second step attempts to solve the problem of free stream sensitivity. This is done by imposing a bound on the stress-intensity ratio, \( \frac{|\tau y|}{k} \), which is generally found to be around 0.3. [7] This bound is imposed by setting the stress-intensity ratio to a minimum limiter function in terms of turbulent viscosity. The equation then becomes

\[
\frac{|\tau y|}{k} = \min \left[ C_\mu \frac{|\partial_y U|}{\omega}, \sqrt{C_\mu} \right]. \tag{28}
\]

This limiter improves the prediction of adverse pressure gradient and separated flow. [7]

From here the blending function is then introduced as:

\[
F_2 = \tanh(\arg_g^2), \tag{29}
\]

Where \( \arg_g = \max \left[ \frac{2\sqrt{k}}{\omega y}, \frac{500C_\mu \nu}{\omega y^2} \right] \).
This blending is devised so that F2 drops from approximately one in most of the boundary layer to zero near the top and in free stream. In this way the k-ω model compromises on its pitfalls with the help of the SST model, and can be shown to be a better predictor for more flow regimes. Menter then posed that using the k-ω model near the wall and k-ε for the rest of the flow allows for smooth interpolation. [7]

2.4 Spalart-Allmaras Model

The Spalart-Allmaras model is a one equation model, instead of the two equation as with the k-ε and k-ω. [24] This changes the transport equation to a single solution based on a produced variable, ν*. This variable is sometimes defined as the “Spalart-Allmaras variable” and is used as a modifier of the transport equations shown below. [24]

The turbulent eddy viscosity is shown as:

\[ \nu_t = \bar{\nu} f_{v1}, \]  

(30)

where

\[ f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \]  

(31)

and

\[ \chi := \frac{\bar{\nu}}{\nu}. \]  

(32)

The transport equation then becomes:

\[ \frac{\partial \bar{\nu}}{\partial t} + u_j \frac{\partial \bar{\nu}}{\partial x_j} = C_{b1}[1 - f_{i2}]\bar{S} \bar{\nu} + \frac{1}{\sigma} \{ \nabla \cdot [(\nu + \bar{\nu}) \nabla \bar{\nu}] + C_{b2} |\nabla \bar{\nu}|^2 \} \]
\[- \left[ C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t1} \right] \left( \frac{\bar{\nu}}{d} \right)^2 + f_{t1} \Delta U^2 \]  

\text{(33)}

where

\[ \tilde{S} \equiv S + \frac{\bar{\nu}}{\kappa^2 d^2} f_{v2}, \]

\text{(34)}

and

\[ f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}. \]

\text{(35)}

With this the set of equations to define the transport then becomes:

\[ S \equiv \sqrt{2 \Omega_{ij} \Omega_{ij}}, \]

\text{(37)}

\[ \Omega_{ij} \equiv \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right), \]

\text{(38)}

\[ f_w = g \left[ 1 + \frac{C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}, \]

\text{(39)}

\[ g = r + C_{w2}(r^6 - r), \]

\text{(40)}

\[ r \equiv \frac{\bar{\nu}}{\tilde{S} \kappa^2 d^2}, \]

\text{(41)}

\[ f_{t1} = C_{t1} g_i \exp \left( -C_{i2} \frac{\omega_i^2}{\Delta U^2} [d^2 + g_i^2 d_i^2] \right), \]

\text{(42)}

and

\[ f_{t2} = C_{t3} \exp(-C_{t4} \chi^2). \]

\text{(43)}
The constants are $\sigma = 0.667$, $C_{b1} = 0.1355$, $C_{b2} = 0.622$, $\kappa = 0.41$, $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma}$, $C_{w2} = 0.3$, $C_{w3} = 2$, $C_{v1} = 7.1$, $C_{t1} = 1$, $C_{t2} = 2$, $C_{t3} = 1.1$, $C_{m} = 2$. [24]

The boundary conditions for the turbulence field are also $v_{t,wall} = 0$, and $\nu_{t, farfield} = 0.210438 \nu_{\infty}$.

It is important with this model that $S$ must never reach zero or fall below. A limiter is often placed on the $S$ term in order to ensure it stays positive, the reason for this is the equation will reach numerical problems if this term does fall below zero. This model also has source terms that are non-zero in the free stream even when vorticity is zero. These terms are very small, on the order of $\frac{1}{d^2}$, but must be recognized.

This model takes into account the method with which grid points are computed. When computing this model it is important to note the difference between computing a minimum distance to the nearest wall and finding the nearest wall grid point. [24] The latter is incorrect as it may incur grid-dependent differences that alter the outcome. This is due to the chance that the minimum distance to the wall is not on a grid point and therefore cannot be calculated exactly by using the grid points specifically.

### 2.5 v²-f Model

The $v^2$-f model is similar to the k-ε in that it adds diffusivity to the transport equations, but it separates the equations into isotropic ($k/\epsilon$) and anisotropic ($v^2/\epsilon$). The $v^2$-f model uses a velocity scale instead of the kinetic energy to determine the smoothing of the eddy turbulence. The following equations describe the turbulent viscosity and subsequent transport equations:
\[ v_0^2 = C_\mu \vec{v}^2 T, \]  
(44)

\[
\frac{\partial \vec{v}^2}{\partial t} + U_j \frac{\partial \vec{v}^2}{\partial x_j} = kf - \frac{\nu^2}{k} \varepsilon + \frac{\partial}{\partial x_j} \left[ \left( \nu + \frac{\nu_t}{\sigma_{\nu^2}} \right) \frac{\partial \vec{v}^2}{\partial x_j} \right],
\]
(45)

\[
L^2 \nabla^2 f - f = \frac{C_1 - 1}{T} \left( \frac{\nu^2}{k} - \frac{2}{3} \right) - C_2 \frac{P_k}{\varepsilon},
\]
(46)

\[
L = C_L \max \left[ \frac{k^{3/2}}{\varepsilon}, C_\eta \left( \frac{\nu^3}{\varepsilon} \right)^{1/4} \right],
\]
(47)

and

\[
T = \max \left[ \frac{k}{\varepsilon}, C_T \left( \frac{\nu}{\varepsilon} \right)^{1/2} \right].
\]
(48)

The constants for this set of equations is: \( C_\mu = 0.22, \sigma_{\nu^2} = 1, C_1 = 1.4, C_2 = 0.45, \) \( C_T = 6, C_L = 0.25, \) and \( C_\eta = 85. \) [28] The third equation is the Helmholtz relaxation elliptical function that is used to smooth the system. From this equation \( L \) is the turbulence length scale and \( T \) is the turbulence time scale.

Though the \( \nu^2-f \) model is in use currently, this thesis will not utilize this method extensively, and will in fact simply touch on its uses.

### 2.6 Convex Optimization

The main goal of this research is to optimize the necessary, statistically significant variables in RANS modeling. This was attempted through the use of convex optimization software, CVXOPT, developed by Stanford University and available for
free as an extension through Python coding. Through the use of these functions the error of sensitive variables in the Navier-Stokes equations can allegedly be effectively reduced. Convex optimization relies on restraints and limits placed on the parameters that will limit the calculations and guide the data to a smallest misfit or prediction error. In our problem the restraints are the boundaries of the system (i.e. the initial conditions and boundary conditions of the flow), and the limits are the best fit to DNS data, which must be nonnegative. The objective function is the prediction error between DNS and the predicted values, or a statistical measurement of unlikeness or implausibility of parameters. Therefore the optimization problem is to find parameter values that are consistent with prior information and give the best fit value. CVXOPT cannot solve the system completely, but can guide the user to an optimal model through error reduction.

Convex optimization begins with the inequality:

\[
\text{minimize} : f_0(x) \\
\text{subject to} : f_i(x) < b_i, i = 1, \ldots, m \tag{49}
\]

Where \(f_0, \ldots, f_m : \mathbb{R}^n t \rightarrow \mathbb{R}\) are convex, and therefore satisfy the equation:

\[
f_i(\alpha x + \beta y) < \alpha f_i(x) + \beta f_i(y) \tag{50}
\]

For all \(x, y\) as an element of \(\mathbb{R}^n\), and all \(\alpha, \beta\) as an element of \(\mathbb{R}\), \(\alpha + \beta = 1\), \(\alpha > 0\), \(\beta > 0\). \cite{2}

The momentum equation is the main focus of the convex problems, as it is the determining factor in the bulk flow, as well as the most sensitive to the turbulent viscosity. If the momentum equation can be optimized to the best fit turbulent viscosity, \(v_t\), the equations can be set to predict the DNS of turbulent flows with high accuracy. In
order to achieve this both $A^+$ and $k$ will need to be optimized. This is achieved by applying a convex optimization algorithm to the normalized set of equations for the bulk velocity and momentum.

With the above defined, an attempt was made at using convex optimization to efficiently mimic DNS data with CVXOPT. However, due to the constraints in the mathematics and the form of which all equations must take, the system is not any more efficient at recreating the turbulent viscosity fields. The main goal of this thesis was to determine a method which would reduce the computing cost of optimizing Navier-Stokes equations for a given flow. Convex optimization was too taxing in terms of mathematics and computing power to be a viable option. This lead the team to seek another option such as linear regression, or a more tractable way to pose the problem than the excessive mathematics of convex optimization.

The adjoint method and utilization of the inverse equations from MIT was found to reduce the equations to a tractable form and allow for optimization of the variables in question. [6] The next section describes the derivation, use, and methodology of the adjoint method and its success in optimizing the turbulent Prandtl number.
CHAPTER 3: ADJOINT METHOD

3.1 Motivation

The adjoint method is used increasingly in computational assisted engineering to solve the optimized RANS equations. The construction of the equations and derivation of the linearized boundary conditions is not inherently difficult, in addition to the effectiveness gained from optimizing the perturbation scalars. For example, Giles and Pierce use the adjoint method and describe its value in determining an optimized geometry for an airfoil in their work. [10] This is the same method used in this research to optimize the turbulent Prandtl number. This approach requires an adjoint equation in the form of equation (51):

\[ \left( \frac{d}{dz} \left( \alpha + \frac{\nu_t}{Pr_t} \right) \frac{d}{dz} \right) \hat{T} = -2(T(Pr_t) - T_{Ref}), \]  

(51)

from the original function:

\[ \left( \frac{d}{dz} \left( \alpha + \frac{\nu_t}{Pr_t} \right) \frac{d}{dz} \right) T = 0, \]  

(52)

as the objective function, and several iterations of RANS equations to be used as the defining and subject functions. The adjoint method is used thus to linearize and smooth the RANS functions via its use of the conjugate transform, or adjoint, matrix. By linearizing the objective function of the RANS model to the discretized variables, the adjoint method can solve for the perturbation in the turbulent Prandtl number via the changes in the temperature.
3.2 Derivation of Adjoint Method

The inverse problem is used to describe the difference in the \( L_2 \) norm of each of the flow fields; from the RANS model and the DNS. The original inverse RANS equations are calculated from the Wilcox k-\( \omega \) model, and are derived as such to describe the dissipation and turbulent kinetic energy:

\[
\text{Turbulent kinetic energy} : \quad k = \frac{1}{2} [u'_i u'_j], \quad (53)
\]

\[
\rho \frac{\partial k}{\partial t} + \rho \langle u'_j \rangle \frac{\partial k}{\partial x_j} = -\rho \langle u'_i u'_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j}
\]

\[
-\rho \varepsilon + \frac{\partial}{\partial x_j} \left[ \nu \frac{\partial k}{\partial x_j} - \frac{1}{2} \rho \langle u'_i u'_j \rangle - \langle p'_i u'_j \rangle \right], \quad (54)
\]

where the right hand side of the dissipation equation (54) is described by the production, dissipation, diffusion, turbulent transport, and pressure diffusion in respective order. [6]

The specific dissipation is then shown as:

\[
\text{Dissipation per unit mass} : \quad \varepsilon = \nu \left( \frac{\partial u'_i}{\partial x_k} \frac{\partial u'_i}{\partial x_k} \right). \quad (55)
\]

The turbulent transport and pressure diffusion are grouped together and assumed to behave according to the gradient transport, or gradient diffusion, hypothesis resulting in the following equation:

\[
\frac{1}{2} \rho \langle u'_i u'_j \rangle + \langle p'_i u'_j \rangle = -\rho \frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_j}. \quad (56)
\]
The gradient diffusion hypothesis is mathematically analogous to Fourier’s Law of Heat Conduction where the transport is down the mean scalar gradient. The scalar flux vector, \( \langle u \phi' \rangle \), gives the magnitude and direction of the turbulent transport. [6]

These equations show the relation between the Reynold’s stress tensor and turbulent kinetic energy, and through the use of the Boussinesq approximation \((\nabla \cdot u = 0)\) can be closed to form the abstract version of the RANS equations for the objective function. This again utilized the Reynolds Analogy relation, and the scales differ as can be seen by the data in section 4. Kolmogorov was the first to posit the equations for the \( \omega \) dissipation and they are based on the physical arguments (turbulent viscosity, pressure, density) and dimensional analysis. [5] The equation is as follows:

\[
\begin{align*}
\rho \frac{\partial \omega}{\partial t} + \rho \langle u_i \rangle \frac{\partial \omega}{\partial x_j} &= \alpha \frac{\omega}{k} \langle u_i u'_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} - \beta \rho \omega^2 \frac{\partial}{\partial x_j} + \left[ \rho (\nu + \sigma \nu_t) \frac{\partial \omega}{\partial x_j} \right],
\end{align*}
\]

(57)

where \( \alpha, \beta, \) and \( \sigma \) are closure coefficients. [6] Finally, the turbulent viscosity can then be described as a function of kinetic energy and it’s dissipation, specifically:

\[
\nu_t = \frac{k}{\omega}.
\]

(58)

With these calculations the inverse RANS can be described and placed into calculating software, in our case Python, to be optimized. Our utilization of the method focused on the Pr\(_t\), which lead to

\[
\left( \frac{d}{dz} \left( \alpha + \nu_t \frac{Pr_t}{} \right) \frac{d}{dz} \right) \hat{T} = -2(Pr_t - T_{Ref})
\]

(59)
as the objective function.

The objective function (59) is then minimized utilizing the steepest descent algorithm as follows:

$$\Delta \alpha = -\varepsilon \frac{dJ}{d\alpha},$$  \hspace{1cm} (60)

where $J$ is the objective function, $\varepsilon$ the step size, and $\alpha$ is the set of design variables. [10] In our case $\alpha$ would be the temperature profiles from the DNS, thus the perturbations from this scalar will alter the objective function.

### 3.3 The Inverse Function

The inverse function is the main function that is used to optimize the flow field calculations. It can be described as an adjoint, constrained, objective function as such:

$$\min J, \text{ subject to }: \nu_t \geq 0.$$  \hspace{1cm} (61)

The objective function can then be described thus:

$$J = ||u(\nu_t) - U_{DNS}||_2,$$  \hspace{1cm} (62)

where $u(\nu_t)$ represents a RANS flow field produced by a specific turbulent viscosity. This is then optimized by calculating the flow field for a new, specific, turbulent viscosity at each iteration to solve the Navier-Stokes equations. The turbulent viscosity is set as a matrix, the dimension of which is the number of nodes in the mesh, that is then stored at each iteration. With each of these iterations the sensitivity gradient, $\varepsilon$, is calculated, thus producing an optimized flow field for each $\nu_t$.

The reason the adjoint method and inverse problem must be solved is because the RANS equations cannot be solved by normal optimization procedures. It is not a
convex problem; by virtue of the $L^2$ norm it is convex [2], but not in any form that is readily used. Convex optimization must be calculated in a mathematical sentence that relates the objective function to a constrained subject, thus calculating the set of equations with relation to one variable. The RANS equations are a set of three equations with five variables, and therefore are not easily optimized. With this in mind the problem became a series of attempts to massage the math into a usable form, but ultimately it was discovered that this adjoint method could solve the inefficiencies just as easily and with less obtuse equations. With the derivation of the Wilcox $k$-$\omega$ equations shown above, the following will describe the derivation of the adjoint method.

3.3.1 Derivation of Inverse Equations

The objective function is directly influenced by the flow solution, and therefore by the turbulent viscosity field, marking it as the optimization constraint. It is expressed as:

$$J = J(w, \nu_t),$$ \hspace{1cm} (63)

where ‘$w$’ is the flow solution. From here we must determine a cost function to be used as the sensitivity function. The first iteration is as follows:

$$\partial J = \frac{\partial J^I}{\partial w} \partial w + \frac{\partial J^I}{\partial \nu_t} \partial \nu_t.$$ \hspace{1cm} (64)

The RANS equations are then described as such for their abstract representation:

$$R(w, \nu_t) = 0.$$ \hspace{1cm} (65)
The “tangent” equation is then used to expand upon the abstract RANS equations:

\[
\left[ \frac{\partial R}{\partial w} \right] \partial w + \left[ \frac{\partial R}{\partial v_i} \right] \partial v_i = 0. \tag{66}
\]

From here one method is to include the adjoint state with the Lagrange Multiplier, \( \psi \), which is introduced and the mean flow field can be set as a constraint. This produces the equation:

\[
\partial J = \frac{\partial J^T}{\partial w} \partial w + \frac{\partial J^T}{\partial v_i} - \psi^T \left( \left[ \frac{\partial R}{\partial w} \right] \partial w + \left[ \frac{\partial R}{\partial v_i} \right] \partial v_i \right),
\]

\[
= \left( \frac{\partial J^T}{\partial w} - \psi^T \left[ \frac{\partial R}{\partial w} \right] \right) \partial w + \left( \frac{\partial J^T}{\partial v_i} - \psi^T \left[ \frac{\partial R}{\partial v_i} \right] \right) \partial v_i. \tag{67}
\]

The adjoint state is then chosen to satisfy:

\[
\left[ \frac{\partial R}{\partial w} \right]^T \psi = \frac{\partial J}{\partial w}, \tag{68}
\]

With the \( w \rightarrow 0 \) due to \( J \) being a function of \( v_i \). Therefore, the second derivative of the flow solution term becomes zero. And thus the sensitivity gradient becomes:

\[
\partial J = \left( \frac{\partial J^T}{\partial v_i} - \psi^T \left[ \frac{\partial R}{\partial v_i} \right] \right) \partial v_i. \tag{69}
\]
These equations make up the alternative Lagrange viewpoint for duality in the adjoint method [10], which can then be used to find optimum turbulent viscosity fields, $\nu_t$, and solve the Navier-Stokes equations for the given flow. [6]

This is the form that will be used within the code to optimize the Prandtl number via the temperature profile. This shows that the objective function that is now inherently convex as all versions of the $L_2$ norm are convex functions, and that the deciding factor in optimization is the scaling of the turbulent Prandtl number. In other words the turbulent Prandtl number is the minimization constraint.

### 3.4 The Tridiagonal Matrix Algorithm

Typical optimization of the RANS equations would be a highly taxing program in terms of processing power. The amount of nodes in the mesh required for a smooth result and the amount of computations ($N_{\text{nodes}}+1$) of the turbulent viscosity field for the sensitivity gradient would be far greater than those necessary to complete the adjoint calculations. This is partially to blame for the slow improvement upon CFD software in the RANS field analysis. The adjoint calculations through the inverse method are less taxing because each of the sensitivity calculations are completed in terms of the mean velocity and viscosity fields rather than full RANS equations. Our system utilizes the Thomas Algorithm, or the tridiagonal matrix algorithm, to solve the system of equations.

The Gaussian elimination is the basis of the Thomas Algorithm, which utilizes backward substitution to solve the diagonally dominant matrix of equations. This, again, takes advantage of the lower cost of processing as it decreases the number of
variables. Through basic row manipulation the matrix of equations can be transformed into an upper triangular matrix. From here the elimination of variables to zero creates a diagonal matrix, which is the form of which all the vectors need to be solved. The method of the Thomas Algorithm is to sweep through the coefficients and eliminate them, then backwards substitute these into the equations to find solutions. The form is shown below for how a system of $n$ unknowns is solved.

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i,$$

where $a_1 = 0$ and $c_n = 0$.

$$\begin{bmatrix}
  b_1 & c_1 & 0 \\
  a_2 & b_2 & c_2 \\
  a_3 & b_3 & \ddots \\
  \vdots & \vdots & \ddots & \ddots \\
  0 & \cdots & & a_{n-1} & b_n
\end{bmatrix} \begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  \vdots \\
  x_n
\end{bmatrix} = \begin{bmatrix}
  d_1 \\
  d_2 \\
  d_3 \\
  \vdots \\
  d_n
\end{bmatrix}.$$

**Figure 1: TDMA Style Matrix Equation**
CHAPTER 4: FINDINGS

4.1 Initial Attempts

Dr. Dubief first proposed this project with an idea that ten percent error in variables such as the Prandtl Number, turbulent viscosity, bulk velocity, and kinetic energy was highly unacceptable with the ability of current programs and processors. A sensitivity analysis was performed on a linear, turbulent flow with a Re$_\tau$ of 137 to determine the significance of the the variables in the k-$\varepsilon$ model. From this sensitivity analysis the significant variables were chosen and an attempt was made to optimize them using CVXOPT.

It was decided that the most important mitigating factors of flow are the Prandtl number, turbulent viscosity, and thermal diffusivity, which unsurprisingly is where most models in use fail. With these specific variables called out we looked into the use of the various methods described above, including the k-$\varepsilon$, k-$\omega$, $v^2$-$f$, and Spallart-Almaras models, and various forms of optimization to improve upon existing flow simulation methods. The optimization of any of these would improve upon the use of said methods so industries and scientists would be able to more accurately represent the model in question. Mitigating the opportunities in the most used method, $k - \varepsilon$, would help propel the computational fluid community toward more accurate, more efficient models.

Convex optimization, specifically the Stanford CVXOPT program, was the first iteration of code utilized to improve the k-$\varepsilon$ model. Since convex optimization requires such stringent mathematical procedures and sets of variables, it was inherently difficult
to take an already high-level, multivariate problem and express it so it was convex and again a set of equations that could be constrained by one variable. It was during this research that Dr. Mads Almassalkhi advised that either linear regression or another form of statistical analysis may be a better option due to the nature of the mathematics. The convex problems were no longer a tractable solution and would be dropped for another method.

It was from this suggestion that the inverse adjoint method was adopted from Q. Wang’s work at MIT [5] and Giles and Pierce’s work [9], thus the project took the form of optimizing the turbulent Prandtl number through the adjoint method. This technique was then verified using the Nelder-Mead method and tested against the solutions from that process.

4.2 Coding

All computations for this project were completed in Python provided as an open-source system. Dr. Yves Dubief supplied most of the basis for the systems and calculations, as well as the direct numerical simulation data. The temperature data for the main code was created previously to be used for several different experiments including this one.

4.2.1 Setup

The work began by creating a system of canonical equations based on the RANS equations for turbulent flow. Pre-existing Python libraries were used to call in functions as necessary. These included libraries such as SciPy, NumPy, and IPython to call in various mathematical operators and graphing techniques. The main calculations
were done with the TDMA Solver from Numpy to determine the constitutive tridiagonal matrix. The tridiagonal matrix, or Thomas Matrix, is used to solve for the diagonally dominant set of variables as a Gaussian elimination method. From here the variables can be determined for the ‘CSR’ matrix and the right and left hand sides of the temperature equation. Each piece of the equation in question is treated as a simplified coefficient and then calculated for each eigenvalue in the matrix.

The system of equations is then stored as the matrix values to be used as data for the iterations of the flow regime. This is done by calling back the TDMA solver with the right and left hand side variables of the equations as the four inputs to the solver. This returns the diagonal solution and thus the resultant matrix of temperature profiles. These two sets of equations are also their own functions as the CSR matrix and diffusion matrix must be solved separately in order to determine the coefficients of the temperature equation.

```python
In [2]: import numpy as np
def TDMA_solver(a, b, c, d):
    ...:
    TDMA solver, a b c d can be NumPy array type or Python list type.
    refer to https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm
    ...
    nf = len(a)  # number of equations
    ac, bc, cc, dc = map(np.array, (a, b, c, d))  # copy the array
    for it in range(1, nf):
        xc = ac[it]/bc[it-1]
        bc[it] = bc[it] - cc[it-1]*xc
        dc[it] = dc[it] - cc[it-1]*xc
    xc = ac
    xc[-1] = dc[-1]/bc[-1]
    for il in range(nf-2, -1, -1):
        xc[il] = (dc[il] - cc[il]*xc[il+1])/bc[il]
    del bc, cc, dc  # delete variables from memory
    return xc
```

**Figure 2: TDMA Solver for temperature equations**
Much of the preliminary coding was used as validation for the optimization and analysis of the experimental data. As with all computational fluid dynamic models, the empirical models must be compared to the analytical models in order to prove their effectiveness. This was accomplished by first analyzing the provided canonical simulation of fully developed flow through a channel and graphically showing each profile (velocity, temperature, and viscosity) in the channel. These can be seen below.

![Velocity Profile from simulated data](image.png)

**Figure 3: Velocity Profile from simulated data**
Figure 4: Temperature Profile from simulated data

Figure 5: Viscosity Profile from simulated data
Once a baseline was established via this proof of easily tractable data, the solutions to the temperature and turbulent Prandtl number equations needed to be verified. This consisted of solving each equation for the known temperatures given from the data; in this way the thermal diffusivity, $\alpha$, turbulent Prandtl number, $Pr_t$, and Nusselt number $Nu_t$ could be established in the traditional method as a constant. Below are the profiles for the temperature profiles compared with a constant $\alpha$ from calculations.

As is seen in figure 6, the calculated thermal diffusivity results in very nearly the same temperature profile as the experimental data, but the analysis is noisy in
comparison, as well as off by a small factor away from the walls. The equation for the
temperature profile is as follows:

\[
\frac{T_j - T_{j-1}}{2(\Delta y)^2} \alpha_{T,j-1} - \frac{T_{j-1} - 2T_j + T_{j+1}}{2(\Delta y)^2} \alpha_{T,j} - \frac{T_{j+1} - T_j}{2(\Delta y)^2} \alpha_{T,j+1} = \alpha \frac{T_{j-1} - 2T_j + T_{j+1}}{2(\Delta y)^2}
\]

\[= \alpha \frac{T_{j-1} - 2T_j + T_{j+1}}{(\Delta y)^2}\] (70)

With \(\alpha\) a function of the turbulent viscosity as shown:

\[\nu T' = \alpha \frac{\partial T}{\partial y}.\] (71)

From these two equations and a matrix solver, the profile in figure 5 was calculated. This is the general use of CFD and the accepted error within the calculations. To this end the graph of the singular valued thermal diffusivity profile itself (Figure 6) proves that optimization is necessary to improve our computation.

The next step is to solve the system altogether. This involved the implementation of each of the previous functions in a loop of equations in order to solve the system for each set of \(A^+\), the distance from the wall.

4.2.2 Optimization Coding

Once the system of temperature values have been determined the optimization and solution to the RANS equations can begin in earnest. In this section the proof of the failing of the Reynolds Analogy will be most apparent. Due to the varying scales, the optimized scalars will drastically effect the outcome of the RANS values, thus adhering more and more closely to the DNS data analysis. Since the near-wall analysis of the
Figure 7: Temperature optimization coding example

data has such a digression from the DNS it becomes quickly apparent that the scalar and momentum do not, in fact, act on the same scales.

This is where the adjoint method is used and the equations take the inverse form in order to be solved in the desired method. Several steps were taken in between to ensure the temperature, diffusivity, and various matrices were of the correct form. Such steps included printing a time-lapse graph of the temperature profile within the flow and the matrix of all the temperature solutions to show the user what values were calculated thus far. Figure 7 above shows the optimization coding that was used; this is a Python adaptation of the equations described previously.

The optimization is simpler due to the implementation of the norm function from the linear algebra library of Numpy, as well as the use of the while loop. The
function breaks down as a cost function for the turbulent Prandtl number ($Pr_t$) which is the norm of the adjoint equation:

$$J = || T(Pr_t) - T_{DNS} ||_{L_2}. \quad (72)$$

From this function the derivative can be calculated and used as the looped equation for the convergence iterations. The following equation is a mathematical representation of the adjoint derivative:

$$\partial J = J(Pr_t^0 + \partial Pr_t) - J(Pr_t^0 - \partial Pr_t). \quad (73)$$

This indicates that the change in the adjoint equation will move toward the local minimum. This is achieved by the while loop below the partial adjoint equation; the new value of the Prandtl number is assigned by adding the $\partial Pr$ if the $\partial J$ is negative, meaning the value is moving toward the lowest point or local minimum. If the $\partial J$ is positive the equation has calculated too far past the minimum and must move back (subtract $\partial Pr_t$) toward it. Once the work cell has calculated an acceptable minimum, by determining that the sign of the current value of the $\partial J$ is the same sign as the previous $\partial J$, the system will recalculate again by decreasing the change in Prandtl number. In this way the system will reach its convergence threshold for a more exact solution, set previously as $10^{-6}$. The results from this calculation can be seen below as the set of values called out to print.
Figure 8: Tabulated Values of Turbulent Prandtl Number

The list of numbers in figure 8 are the outcome of the cost function calculations. The final number is the most important, as it is the value determined by the code that represents the \( \text{Pr}_t \). This value is less than expected, as the turbulent Prandtl number is generally around 0.85 for most turbulent flow, but it does show that the optimization coding of the \( \text{Pr}_t \) can be achieved in eight iterations to some degree of accuracy.

4.2.3 General Solutions

This is the first iteration of optimizing the turbulent Prandtl number. The value given at the end of the set of calculations will be the optimized turbulent Prandtl number for the given set of values and boundary conditions. While effective, the ultimate goal is to optimize the Prandtl number to each iteration of the flow in space. The adjoint method has succeeded in simplifying the equations and the time taken to complete a full set of calculations has been significantly reduced from attempting to use CVXOPT. The system must then be adapted to solve the set of equations as a matrix incorporating the iterations in space, then again for the remaining variables.
From the temperature and thermal diffusivity solutions, the $Pr_t$ was solved as a matrix for each iteration in space at a step of 0.1. As shown in the figure below, the turbulent Prandtl number ranges in accordance to the distance from the wall.

![Figure 9: Prandtl number per position in channel](image)

Figure 9 shows the turbulent Prandtl number varies with respect to its distance from the wall. Approaching the boundaries, the $Pr_t$ rises, while it jumps in the center of the flow as it cannot be divided by zero. This is in accordance to simulated data, as in the transition layers the $Pr_t$ ranges at an inverse relationship to $y$. Since the turbulent Prandtl number is a ratio between the eddy diffusivities of the momentum and heat transfer equations, $Pr_t = \frac{\nu_T}{\alpha_T}$, this inverse relationship is to be expected. The thermal diffusivity near the wall approaches zero, hence the $Pr_t$ approaches infinity, and the semi-asymptotic nature around the center of the flow shows that the momentum reaches zero in the bulk velocity.
With this range of Pr in mind, the next set of code was created to adapt to the change in value with respect to $A^\tau$. In this way, the Prandtl number can be shown to adapt to the flow and thus the temperature profile will be more accurate to the DNS data. The first graphical representation of the optimized Pr is shown below in Figure 8. This represents the value of 0.625 determined by the initial optimization loop and does not account for the variation across the height.

As seen in the figure above, the optimized temperature profile more closely relates to the DNS profile. The “R” profile is the singular-valued residual data calculation with an initial guess of one and a calculated value around 0.96. There is most likely an error here, however, as this calculated value is much too far away from
an expected value. \( \text{Prt} = 0.624 \) does not represent the final calculated value, and further verification is needed to ensure a more accurate solution.

The above solutions provided a foundation for the height-differentiable optimized turbulent Prandtl number. The code was adapted to account for the variation in distance from the wall, as represented by \( A^+ \). The code can be seen in figure 9 below.

```python
def cost_function_damping_function(x):
    global T_DNS
    return la.norm(temperature_solver_damping_function(x[0], x[1]) - T_DNS, 2)

def temperature_solver_damping_function(Prt,A):
    global N, dy, T_ml, T_N, alpha, y
    a = np.zeros(N)
    b = np.zeros(N)
    c = np.zeros(N)
    d = np.zeros(N)
    beta = np.ones(N)

    # beta /= (1. - np.exp(-(1. - np.abs(y))/A)) * Prt
    beta /= (1 + np.power((1.0 - np.abs(y))/A, 2)) * Prt
    T_ml = 1.
    T_N = -1
    alpha_star = alpha*ones(N)
    #alpha_star += nut*beta_array[np.argmin(R_array)] #alphatnu_t/Fr_t
    alpha_star += nut*beta #alphatnu_t/Fr_t

    #alpha_star += alpha_t
    a[0] = alpha_star[0] + alpha
    a -= 2*dy**2
    b[0] = alpha + 2*alpha_star[0] + alpha_star[1]
    b -= 2*dy**2
    c[0] = 2*dy**2
    d[0] = a[0]*T_ml
    d[1] = c[1]*T_N
    T = TUMASolver(a, b, c, d)
    return T

x = optimize.minimize(cost_function_damping_function, [1., 0.01], method='Nelder-Mead')
print(x)
```

Figure 11: Optimization of \( \text{Pr}_t \) With Respect to Wall

The result of this was to show the dampened function as a temperature profile that varies the \( \text{Pr}_t \) to its distance from the wall. Once this was established the calculated value was set by the Nelder-Mead optimization method. The solution is shown below in figure 12.
The solution is the “x: array([ 0.998, 0.044 ])” entry. The first value in the array is the calculated, optimized, turbulent Prandtl number, the second value is the A⁺ solution, or the distance from the wall. This distance is still within the boundary layer in terms of the flow.

The Nelder-Mead optimization shows a calculated value much closer to unity. Though it does not match the previously calculated value of Prt = 0.625, the profile more closely relates to the DNS data. This can be seen below in figure 13.
The further optimized profile approaches the DNS profile more closely than expected. Since the values are significantly different, approximately 0.37, the profile was not originally expected to correlate so closely. However, the Nelder-Mead method is a well established optimization method and the simplex model converges quickly to an optimum with a function of few parameters.

4.2.3 Oscillating Flow Setup

As a continuation of previous experiments, analysis was carried out on an oscillating flow. This flow was created to simulate gas flow in an automobile engine, showing the change in flow of a piston cylinder. The analysis of said flow was carried out for thirty-two phases in order to show the oscillating nature of the pulsatile
mechanics. Each calculation is carried out in one of the phases and shown as its own profile in the graphs presented below. In this way the flow can be analyzed for the difference in Reynolds number, Prandtl number, or other desired scalars or variables. Below figure 14 shows the velocity profiles as they appear in the pulsatile flow in accordance to the distance from the wall.

![Figure 14: Pulsatile mean velocity profiles](image)

Only the first sixteen phases are shown as the next sixteen will repeat the same profiles by nature. The velocity of the flow is shown with respect to the distance from the wall, as most of the variables are shown. In this way the flow is normalized by $y^+$ in order to provide reference. Likewise, the below temperature profiles are seen in the pulsatile flow as normalized scalars to the distance from the wall.
Again as a point of reference, figure 16 shows the sensitivity analysis of turbulent Prandtl number profiles (Pr) by phase.
From this setup the next two figures will show the turbulent Prandtl number normalized to the wall as was shown in the initial flow. Note that the spikes in $\text{Pr}_t$ close to the wall are much higher than that of the initial flow. This is mostly due to phase eight, as will be discussed below.
Figure 17: Pr$_t$ Normalized - Macro View

Figure 18: Pr$_t$ Normalized: Zoom View
Figure 17 is a macro view of the initial turbulent Prandtl number values of the pulsatile flow. This shows the highest values of Pr, reach approximately sixty-five and happens at phase 8. Figure 18 shows the Pr profiles in a classic view, ignoring this important spikes. These are the focus of this research, as the Reynolds Analogy assumption that the thermal and momentum scales occur at the same scale is thus disproven. With the focus on these spikes, the further analysis will utilize a damping function to recognize this. Shown below is the plot used as a graphic for the combination of the damping functions. These functions mimic the asymptotic nature of the Pr and the plateau within the fully developed flow.

The combination of these two functions are used to normalize the Pr function by A and best fit the data. In future work, experimentation on the proper damping function and value of A will be explored. With the optimization of these two variables, A and n (the value of the power law), the analytical data can again be better fit to the
DNS. In the next section the solutions from the optimization of the functions for the heat flux and turbulent Prandtl number will be discussed.

4.2.4 Oscillating Flow Solutions

As a second test to assess the validity of optimizing the Prt, we utilized this oscillating flow simulation. From the setup above the same equation for the turbulent Prandtl number,

\[
Pr_t = \frac{u'v' d\theta/dy}{v'^2 d\pi dy},
\]

was used to determine the turbulent Prandtl number for each node in the phase. Initially, the flow is solved with the k-\(\varepsilon\) equation, as it is the standard. The graphs can be seen below of the kinetic energy and diffusion by phase.

![Momentum profiles by phase normalized to wall](image)

*Figure 20: Momentum profiles by phase normalized to wall*
These graphs are important, as the momentum and diffusion of which are tantamount to the Reynolds Analogy. Though not on the same scale as the turbulent Prandtl number, it is important to note the spikes near the wall. Interestingly, the phases at which these spikes reach a maximum do not align with the maximum in the $Pr_t$; the $k$ and $\varepsilon$ graphs maximize at phase 12, whereas the $Pr_t$ maximizes at phase 8. This could be due to the lag time in energy mixing from the turbulence, but would require further research that this project is not focused on.

The next step is to optimize the $Pr_t$ equation, as well as the heat flux, to show the significance of the value of the turbulent Prandtl number. Figure 22 shows the heat flux as function of a constant $Pr_t$ and the lack of alignment to the DNS data.
As can be seen in figure 22, the initial guess of a Pr$_t$ of 0.9 does not convey a proper fit to the direct numerical simulation data. In this way the Reynolds Analogy fails, and most industry standard tests miss the mark. There is a phase shift of approximately three, the DNS maximizes around phase fifteen and the analytical data maximizes around phase twelve. There is also a significant magnitude loss from the DNS to the simulated data, again a value near to three.

With optimization and the addition of the damping function the analytical data can be shown to much more closely relate, even in pulsatile flow. Again the graphs of the heat flux were analyzed with optimized turbulent Prandtl numbers as shown in figures 23 and 24.
As can be seen in these figures, the phase alignment can be easily rectified, though the amplitude and magnitude are more difficult to achieve. Both functions utilize the same optimization method, but the power law of the damping function was
tested to discover its effect on the simulation. The optimization was performed separately and then graphed similarly to show this variation.

The damping function and optimization have been successful in proving that a varied turbulent Prandtl number has great effect on the heat flux matching the DNS data. The success of these findings not only opens up research into finding a better Reynolds Analogy, but also raises the question of why the momentum diffusion is so much greater near the wall than the thermal diffusion.

4.3 Verification

In order to verify the accuracy of the optimized variables the system must be tested against known solutions and the error calculated. For this reason more than one optimization method was used. With the basic adjoint method and the Nelder-Mead method resulting in two results it follows that though successful, the method in question must be further analyzed to discover its faults and then corrected. This project was successful in creating a skeleton for optimization of RANS models, but since the error is still too high from the initial attempt, 37%, more work must be done to improve it.

It is important to note that the Nelder-Mead method is effective at low variable functionality, but fails rapidly when more than ten variables are passed to the simplex. This is because the simplex method will find a local optimum, but not always a global optimum, due to its nature of quickly converging on a minimum based solely on the initial guess. The adjoint method will be more effective in finding solutions when the flow is analyzed for each node, as it can handle a much larger set of parameters. This is why it was used as the test and the Nelder-Mead only used as verification.
Since the resultant solutions were shown to have significant impact, however, this project was successful in proving the need to examine the diffusion more closely. The verification of this comes from an examination of the graphical representation of the temperature profiles. As seen in figures 6, 9, and 13, the changes in diffusion ratio change the profile drastically, even with a constant Pr. Examination of other data will help prove this is not just a case for the given set, and examination of different types of flows will further expand the depth of impact the Pr has on a flow.

This can be seen in the analysis of the pulsatile flow in section 4.2.4. With this analysis a more complex flow was simulated and analyzed, with a focus on the heat flux. This further verification again shows the need for optimized variables and a better Reynolds Analogy equation.

4.4 Results and Future Work

With the results shown above, our research shows that the existing model can be further improved and can simulate flows that have a previously complicating nature such as scalar transport or oscillations. The utilization of the adjoint method and optimization of the scalar functions has led to a series of profiles that more closely relate to the DNS analysis. The RANS equations were linearized in order to be used in the adjoint objective form and the DNS temperature profiles were analyzed to formulate a foundation. From this beginning, the turbulent Prandtl number was optimized to show that the dissipation varies within the flow based on distance from the wall and now more closely matches the DNS data. Importantly the spike in Pr near the wall was found to illustrate the failings of the Reynolds Analogy as the momentum and
thermal diffusivity do not act at the same scale as assumed. Near the wall the turbulent viscosity (momentum diffusion) is far greater than the thermal diffusivity. This shows by some action of the flow the thermal dissipation is incredibly small compared to the energy of the fluid, thus the flow is far more energetic than assumed. By this discovery we recommend the Reynolds Analogy is not robust enough to make CFD models with appropriate accuracy. The next steps would be to create a set of equations with the adjoint method to linearize and optimize various flow variables. The adjoint method is one of the easiest ways to implement the iterative process needed to optimize RANS scalars; the method described in this manuscript found this to be true for the turbulent Prandtl number.

Moving forward, the other variables in question such as the Nusselt number (heat flux), thermal diffusivity, and other scalar transports, can be optimized to drastically improve the model in use. Then the data can be simulated for flows with a highly complex geometry, a chemical transport, or high temperature flow. This will take further analysis in the adjoint method and optimization of these models, but this was a large first step. If more variables can be optimized in accordance with the given models the industry may yet take another leap into finer-tuned models and more complicated flows. For example, biomedical research on capillary flows and blood pressure conservation through junctions between capillaries and venules would benefit from more specific models taking into account the ‘duct geometry’ or the oscillatory nature of the flow. Another beneficiary would be the aeronautics industry as a closer look at the geometry of an airfoil in high velocity or high temperature flow could be achieved
through the specific optimization of the turbulent viscosity at each point in the flow field. Since this research was conducted in term of distance from the wall, $y^+$, and not a specific height designation, each of these calculations is scaled.

A significant learning from this project was the effect that is often ignored in these flows: the strong influence the diffusion has on the flow. Since the equations are averaged, and since the diffusion ratio ($Pr_t$) is used as a constant, the minute changes in the layers of a flow are lost. For example, even in a simple, classical model of flow over a heated, flat plate, the turbulent Prandtl number had significant effect on the boundary layer. This in turn has great effect on the transition layer, the formation of turbulent and laminar flows, and the bulk flow above the boundary. Since the diffusion has such great effect on the flow, it came as a surprise that this problem has been overlooked for so long. There are numerous advantages to modeling the effect diffusion has on a flow such as more precisely predicting the boundary layer on an airfoil. Since the Prandtl number is a ratio of the momentum diffusivity to thermal diffusivity it follows that when the ratio is unity, the boundary layer is in equilibrium. If the $Pr_t > 1$ the momentum diffusivity is greater than the thermal diffusivity and the boundary layer will separate at a higher $A^+$, thus effecting the wall shear. Conversely, if the $Pr_t < 1$ the thermal diffusivity is strong, and again the wall shear will be effected as the boundary layer separates closer to the wall.

The most notable conclusion is that there is a way to utilize existing models and improve them to acquiesce to the necessary complications in a given flow. Dimensional analysis is a great basis for determining a large area, or averaged flow, field velocity,
but given changes in the flow the models fail. With the use of the optimization methods and inverse model solutions, each of these methods can be used to simulate a flow that more closely relates to the direct numerical simulation data. Further research into broader use of the adjoint method would allow the expansion of understanding in turbulent models and could refine analysis in industry. As the current standards incur so much error, the findings of this research are recommended to expand upon the adjoint optimization of Reynolds Average Navier-Stokes models.
Bibliography


