2015

Prediction of discretization error using the error transport equation

Don Roscoe Parsons III

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Parsons III, Don Roscoe, "Prediction of discretization error using the error transport equation" (2015). *Graduate Theses, Dissertations, and Problem Reports*. 6382.
<https://researchrepository.wvu.edu/etd/6382>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

PREDICTION OF DISCRETIZATION ERROR
USING THE ERROR TRANSPORT EQUATION

DON ROSCOE PARSONS III

DISSERTATION SUBMITTED TO THE
COLLEGE OF ENGINEERING AND MINERAL RESOURCES
AT WEST VIRGINIA UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

Ismail Celik, Ph.D., Chair

Nigel Clark, Ph.D.

Wade Huebsch, Ph.D.

William Rogers, Ph.D.

Richard Turton, Ph.D.

MECHANICAL AND AEROSPACE ENGINEERING DEPARTMENT

Morgantown, West Virginia

October 2015

Keywords: numerical analysis, discretization error, error transport equation

Copyright 2015

Abstract

Prediction of Discretization Error Using the Error Transport Equation

Don Roscoe Parsons III

Since its conception, computational fluid dynamics (CFD) has had a role to play in both the industrial and the academic realm. Due to the availability of relatively cheap computer resources, CFD is now playing an increasingly critical role in the industry compared to experiments. It is also becoming increasingly important for the CFD analyst to have an in-depth understanding of the error inherently present in numeric and mathematical models. At a minimum, CFD simulation results should be accompanied by some analysis of the error including some type of grid convergence study and an estimation of the corresponding numerical uncertainty.

Error estimation is commonly done using a method known as Richardson Extrapolation (RE). While RE does produce good error predictions, the requirements for appropriate application of RE are often cumbersome and even prohibitive. In order to achieve an accurate estimate of the discretization error using RE, solutions on at least three grids are required. To obtain satisfactory results, all of these solutions must be in the asymptotic regime, which often requires solutions on more than three grids to verify. Solutions of practical interest in the industry are often quite complex and require a large number of grid points. In this situation, it is not always desirable to produce solutions on three significantly different grids.

This study focuses on an approach to quantify the discretization error associated with numerical solutions by solving an error transport equation (ETE). The goal is to develop a method that can be used to adequately predict the discretization error using the numerical solution on only one grid/mesh. The primary problem associated with solving the ETE is the development of the error source term which is required for the solution of the problem. In this study, a novel approach is considered which involves fitting the

numerical solution with a series of locally smooth curve fits and then blending them together with a weighted spline approach. The result is a continuously differentiable analytic expression that can be used to determine the error source term. Once the source term has been developed, the ETE can easily be solved using the same solver that is used to obtain the original numerical solution.

The new methodology is applied to increasingly complex problems to quantify the discretization error. The method is first validated with the simplistic 1-D and 2-D convection diffusion problem. For both cases the results were very promising. However, in order for this method to be of practical use in the industry, the method must be applicable to the Navier-Stokes equations which are used to solve complete flow fields. The method is extended to solution of the Navier-Stokes equations with increasing complexity. The obtained results indicate that there is much promise going forward with the newly developed source evaluation technique and the ETE.

to my children who inspire me every day in ways that they don't yet understand

Acknowledgements

I don't know if I can adequately express my gratitude to everyone that truly deserves it. I have so much to be thankful for and so many people to be appreciative of. I suppose first and foremost I would like to thank my committee members for their help, support, and time. I understand that because of my working while finishing this research the path to this point has been very bumpy. However, all of the committee members have been exceedingly understanding throughout the entire process. I truly appreciate this more than they know.

I would also like to thank all my friends in the CFD lab. While I only attended WVU full time for three years I met some amazing people that I shall remember forever. I would like to particularly thank Raju, Jaggu, Ertan, Nihan, Jose, and Fransisco. All of you were an amazing wealth of knowledge and even better friends. I truly hope life holds the most amazing gifts for you.

I also must thank my family. My entire family all the way down through aunts and uncles have been so incredibly supportive throughout this entire process. I appreciated every single word of encouragement more than they can possibly imagine. I thank my dad for being there for me. I know I can always call him with any problem I may have. Most of all I would like to thank my mom and my sister. They have absolutely represented the most consistent and amazing support system I could ever ask for in my life. It would take me another dissertation to adequately express how many ways they have helped me, especially over the last five years. I would also like to thank my boys, Declan and Colin. More than anything they are what drives me to be my best every single day. I cannot imagine my life without them in it.

I would also like to thank my girlfriend Auzar. Girlfriend seems like such an inadequate word to express what she has meant in my life. I know she will impact my life in so many ways as we grow together.

Finally and most importantly I would like to thank Dr. Celik. Without his help and guidance I would absolutely not be at this point. He has been so much more understanding throughout this process than I probably deserve. Over the past five years I have gone through some very difficult times. If I would have had to endure all of that

plus lose the opportunity to finish my PhD I would have been lost. Had I followed Dr. Celik's advice on more occasions I would probably have finished much sooner and with many less headaches. But truly, thank you for all of your support.

Thank you all for making this possible and making my life as great as it has been in a long time.

Table of Contents

Abstract.....	ii
Dedication.....	iv
Acknowledgements.....	v
Table of Contents.....	vii
List of Tables.....	x
List of Figure.....	xi
List of Symbols.....	xvii
Chapter 1 Introduction	1
1.1 Verification and Validation in CFD Applications	1
1.2 Sources of Numerical Error	2
1.3 Discretization Error Prediction	4
1.3.1 Richardson Extrapolation	4
1.3.2 Error Transport Equation	7
1.4 Review of Error Transport Equation Literature.....	10
1.5 Objectives and Motivations of This Study	13
Chapter 2 Governing Equations and Methodology	15
2.1 Steady 1-D Advection Diffusion	15
2.1.1 Steady 1-D Advection Diffusion Error Transport Equation	16
2.2 Steady 2-D Advection Diffusion	17
2.2.1 Steady 2-D Advection Diffusion Error Transport Equation	17
2.3 Navier-Stokes Equations	18
2.3.1 Error Transport Equation for 2-D Navier Stokes Equations.....	19
Chapter 3 Weighted Spline Interpolation	25
3.1 One-Dimensional Weighted Spline Method	25
3.1.1 Development of One-Dimensional Weighting Functions	29
3.2 Two-Dimensional Weighted Spline Method	30

3.2.1	Development of Two-Dimensional Weighting Functions	36
3.3	Application of Radial Basis Functions for Local Spline-Region Data Fits	38
3.3.1	Radial Basis Function Methodology	39
3.3.2	Comparison of Polynomial WSM and RFB WSM	41
Chapter 4	Applications to Steady 1-D Advection Diffusion Equation Solutions	63
4.1	Problem Description and Analytic Solution	63
4.2	Numerical Solution	64
4.2.1	Description of Domain, Solver, and Boundary Conditions.....	64
4.2.2	Solution Verification.....	65
4.2.3	Solutions	69
4.3	Solution of the Error Transport Equation	73
Chapter 5	Applications to Steady 2-D Advection Diffusion Equation Solutions	81
5.1	Problem Description and Analytic Solution	81
5.2	Numerical Solution	83
5.2.1	Description of Domain, Solver, and Boundary Conditions.....	83
5.2.2	Solutions	84
5.3	Solution of the Error Transport Equation	87
Chapter 6	Applications to Developing Flow Between Parallel Plates	93
6.1	Problem Description and Analytic Solution	93
6.2	Numerical Solution	95
6.2.1	Description of Domain, Solver, and Boundary Conditions.....	95
6.2.2	Solutions	96
6.3	Solution of the Error Transport Equation	101
6.3.1	Boundary Condition Considerations	105
Chapter 7	Applications to Lid Driven Cavity Flow	112
7.1	Problem Description and the Benchmark Solution.....	112
7.2	Numerical Solution	116

7.2.1	Description of Domain, Solver, and Boundary Conditions.....	116
7.2.2	Solution Verification.....	117
7.2.3	Solution of the Error Transport Equation	125
Chapter 8 Applications to Backward Facing Step Flow		146
8.1	Problem Description and the Benchmark Solution.....	146
8.2	Numerical Solution	151
8.2.1	Description of Domain, Solver, and Boundary Conditions.....	151
8.2.2	Solution Verification.....	153
8.2.3	Solution of the Error Transport Equation	164
Chapter 9 Applications to a Suddenly Accelerated Flat Plate		187
9.1	Problem Description and Analytic Solution	187
9.2	Numerical Solution	190
9.2.1	Description of Domain, Solver, and Boundary Conditions.....	190
9.2.2	Solution Verification.....	192
9.3	Solution of the Error Transport Equation	197
Chapter 10 Conclusions and Future Work		208
10.1	Conclusions.....	208
10.2	Future Work	213
Bibliography.....		213

List of Tables

Table 3.1: One-Dimensional Weighting Functions	30
Table 3.2: Two-Dimensional Weighting Functions	37
Table 3.3: Common Radial Basis Functions	40
Table 7.1: Benchmark Data for the U Velocity at the Geometric Center Line	115
Table 7.2: Benchmark Data for the V Velocity at the Geometric Center Line	115

List of Figures

Figure 3.1: A set of discrete data divided into sub-regions (1-4) and spline-regions (I-V)	26
Figure 3.2: Quadratic Fit of the Discrete Data Over Each of the Five Overlapping Spline Regions	27
Figure 3.3: Left and Right Weighting Functions for Each Sub-Region	28
Figure 3.4: Final fit of Discrete Data Using WSM	29
Figure 3.5: Spline-Region Overlapping for an Interior Sub-Region	32
Figure 3.6: Spline-Region Overlapping for a Side Sub-Region	33
Figure 3.7: Spline-Region Overlapping for a Corner Sub-Region	34
Figure 3.8: Contours of Weighting Functions Applied to Corresponding Spline-Regions	36
Figure 3.9: Analytic Functions Used in The Evaluation of the WSM	42
Figure 3.10: Data Fit From All Variants of the WSM with 9 Sub-Regions	44
Figure 3.11: Data Fit From All Variants of the WSM with 36 Sub-Regions	45
Figure 3.12: Data Fit From All Variants of the WSM with 100 Sub-Regions	46
Figure 3.13: Data Fit From All Variants of the WSM with 9 Sub-Regions	48
Figure 3.14: Data Fit From All Variants of the WSM with 36 Sub-Regions	49
Figure 3.15: Data Fit From All Variants of the WSM with 100 Sub-Regions	50
Figure 3.16: L2 Normal of the Error in All Variants of the WSM When Applied to $f = \sin 2\pi x^2 \cos 2\pi y^2$	52
Figure 3.17: L2 Normal of the Error in All Variants of the WSM When Applied to $f = \sin 4\pi x^2 \cos 4\pi y^2$	53
Figure 3.18: L2 Normal of the Error in the 1st X Derivative When Applied to $f = \sin 2\pi x^2 \cos 2\pi y^2$	54
Figure 3.19: L2 Normal of the Error in the 1st X Derivative When Applied to $f = \sin 4\pi x^2 \cos 4\pi y^2$	55
Figure 3.20: L2 Normal of the Error in the 2nd X Derivative When Applied to $f = \sin 2\pi x^2 \cos 2\pi y^2$	56
Figure 3.21: L2 Normal of the Error in the 2nd X Derivative When Applied to $f = \sin 4\pi x^2 \cos 4\pi y^2$	57

Figure 3.22: L2 Normal of the Error in the 1st Y Derivative When Applied to $f = \sin 2\pi x^2 \cos 2\pi y^2$	58
Figure 3.23: L2 Normal of the Error in the 1st Y Derivative When Applied to $f = \sin 4\pi x^2 \cos 4\pi y^2$	59
Figure 3.24: L2 Normal of the Error in the 2nd Y Derivative When Applied to $f = \sin 2\pi x^2 \cos 2\pi y^2$	60
Figure 3.25: L2 Normal of the Error in the 2nd Y Derivative When Applied to $f = \sin 4\pi x^2 \cos 4\pi y^2$	61
Figure 4.1: Numerical Solutions on Grids of Increasing Refinement	66
Figure 4.2: True Error on Grids of Increasing Refinement.....	67
Figure 4.3: L2 Normal of the True Error as a Function of Grid Spacing.....	68
Figure 4.4: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=10$	70
Figure 4.5: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=20$	71
Figure 4.6: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=100$	72
Figure 4.7: Error Source Term for $Pe=10$	75
Figure 4.8: Error Source Term for $Pe=20$	76
Figure 4.9: Error Source Term for $Pe=100$	77
Figure 4.10: True Error and ETE Solution for $Pe=10$	78
Figure 4.11: True Error and ETE Solution for $Pe=20$	79
Figure 4.12: True Error and ETE Solution for $Pe=100$	80
Figure 5.1: Contour Plot of the Analytic Solution for 2-D Advection Diffusion with a Point Source.....	83
Figure 5.2: Contour Plots of the Analytical and Numerical Solutions for 2-D Advection Diffusion with a Point Source.....	85
Figure 5.3: Profiles of Analytical and Numerical Solutions for 2-D Advection Diffusion With a Point Source.....	86
Figure 5.4: Error Source Term for 2-D Advection Diffusion With a Point Source.....	89
Figure 5.5: Contour Plots of the True Error and ETE Solution for 2-D Advection Diffusion With a Point Source.....	90
Figure 5.6: Profiles of True Error and ETE Solution for 2-D Advection Diffusion With a Point Source.....	91

Figure 5.7: Profiles of True Error and ETE Solution for 2-D Advection Diffusion With a Point Source.....	92
Figure 6.1: Schematic Representation of Flow Between Parallel Plates	93
Figure 6.2: U Velocity Contours for Numerical Solutions on Multiple Grids	97
Figure 6.3: Analytical and Numerical U Velocity Profile on a 40x10 Grid	98
Figure 6.4: Analytical and Numerical U Velocity Profile on a 80x20 Grid	99
Figure 6.5: Analytical and Numerical U Velocity Profile on a 160x40 Grid	100
Figure 6.6: Contour Plots of the Continuity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids.....	102
Figure 6.7: Contour Plots of the U Velocity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids.....	103
Figure 6.8: Contour Plots of the V Velocity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids.....	104
Figure 6.9: True Error and ETE Solution on 40x10 Grid	108
Figure 6.10: True Error and ETE Solution on 80x20 Grid	109
Figure 6.11: True Error and ETE Solution on 160x40 Grid.....	110
Figure 6.12: L2 Normal or the True Error and ETE Solution	111
Figure 7.1: U Velocity Profile (From Ghia et al.) at the Geometric Centerline for Re=100	114
Figure 7.2: V Velocity Profile (From Ghia et al.) at the Geometric Centerline for Re=100	114
Figure 7.3: Contour Plots of the U (left) and V (right) Velocities on the 21x21 Grid.....	118
Figure 7.4: Contour Plots of the U (left) and V (right) Velocities on the 61x61 Grid.....	119
Figure 7.5: Contour Plots of the U (left) and V (right) Velocities on the 101x101 Grid..	120
Figure 7.6: U Velocity at the Vertical Geometric Centerline of the Square Lid Driven Cavity	122
Figure 7.7: V Velocity at the Horizontal Geometric Centerline of the Square Lid Driven Cavity	123
Figure 7.8: Profiles of the U (left) and V (right) True Error Calculated Using Interpolated Ghia Data	124
Figure 7.9: Continuity Error Source Term for 21x21, 61x61, and 101x101 Grids	126
Figure 7.10: U Velocity Error Source Term for 21x21, 61x61, and 101x101 Grids.....	127
Figure 7.11: V Velocity Error Source Term for 21x21, 61x61, and 101x101 Grids	128

Figure 7.12: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 21x21 Grid.....	130
Figure 7.13: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 61x61 Grid.....	131
Figure 7.14: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 101x101 Grid.....	132
Figure 7.15: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 21x21 Grid.....	133
Figure 7.16: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 61x61 Grid.....	134
Figure 7.17: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 101x101 Grid.....	135
Figure 7.18: Variation in the L2 Normal of the U Velocity Error as a Function of Grid Spacing	137
Figure 7.19: Variation in the L2 Normal of the V Velocity Error as a Function of Grid Spacing	138
Figure 7.20: U Velocity Error Profile at X=0.25 for 21x21, 61x61, and 101x101 Grids ..	140
Figure 7.21: U Velocity Error Profile at X=0.5 for 21x21, 61x61, and 101x101 Grids ...	141
Figure 7.22: U Velocity Error Profile at X=0.75 for 21x21, 61x61, and 101x101 Grids ..	142
Figure 7.23: V Velocity Error Profile at Y=0.25 for 21x21, 61x61, and 101x101 Grids ..	143
Figure 7.24: V Velocity Error Profile at Y=0.5 for 21x21, 61x61, and 101x101 Grids ...	144
Figure 7.25: V Velocity Error Profile at Y=0.75 for 21x21, 61x61, and 101x101 Grids ..	145
Figure 8.1: Schematic of Backward Facing Step Flow	147
Figure 8.2: Variation of the Primary Recirculation Zone Length With Reynolds Number	148
Figure 8.3: Experimental and Numerical Profiles of U Velocity Obtained by Armaly et al.	149
Figure 8.4: Measured and Numerically Calculated Reattachment Length Obtained by Armaly et al.	151
Figure 8.5: U Velocity Contours at Various Values of Reynolds Number	155
Figure 8.6: U Velocity Contours at Various Values of Reynolds Number	156
Figure 8.7: V Velocity Contours at Various Values of Reynolds Number	157
Figure 8.8: V Velocity Contours at Various Values of Reynolds Number	158
Figure 8.9: U Velocity Profiles at Various X Coordinate Locations	159

Figure 8.10: U Velocity Profiles at Various X Coordinate Locations	160
Figure 8.11: Recirculation Zone Length as a Function of Reynolds Number.....	161
Figure 8.12: U Velocity Profiles at Various X Coordinate Locations on All Grids.....	163
Figure 8.13: Continuity Error Source Terms for 21x11, 61x31, and 101x51 Grids	166
Figure 8.14: U Velocity Error Source Terms for 21x11, 61x31, and 101x51 Grids	167
Figure 8.15: V Velocity Error Source Terms for 21x11, 61x31, and 101x51 Grids	168
Figure 8.16: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 21x11 Grid.....	170
Figure 8.17: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 61x31 Grid.....	171
Figure 8.18: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 101x51 Grid.....	172
Figure 8.19: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 21x11 Grid.....	173
Figure 8.20: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 61x31 Grid.....	174
Figure 8.21: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 101x51 Grid.....	175
Figure 8.22: Variation in the L2 Normal of the U Velocity Error as a Function of Grid Spacing	177
Figure 8.23: Variation in the L2 Normal of the V Velocity Error as a Function of Grid Spacing	178
Figure 8.24: U Velocity Error Profile at X=0.5 for 21x11, 61x31, and 101x51 Grids	181
Figure 8.25: U Velocity Error Profile at X=1.0 for 21x11, 61x31, and 101x51 Grids	182
Figure 8.26: U Velocity Error Profile at X=1.5 for 21x11, 61x31, and 101x51 Grids	183
Figure 8.27: U Velocity Error Profile at X=2.0 for 21x11, 61x31, and 101x51 Grids	184
Figure 8.28: V Velocity Error Profile at Y=0.06 for 21x11, 61x31, and 101x51 Grids ...	185
Figure 8.29: V Velocity Error Profile at Y=0.14 for 21x11, 61x31, and 101x51 Grids ...	186
Figure 9.1: Schematic of the Flow Above a Suddenly Accelerated Flat Plate	188
Figure 9.2: Profiles from the Analytic Solution for a Suddenly Accelerated Flat Plate at Different Values of Time	190
Figure 9.3: U Velocity Profiles at t=0.001, t=0.002, and t=0.003 on the 41x21 Grid	194
Figure 9.4: U Velocity Profiles at t=0.001, t=0.002, and t=0.003 on the 81x41 Grid	195
Figure 9.5: U Velocity Profiles at t=0.001, t=0.002, and t=0.003 on the 161x81 Grid ...	196

Figure 9.6: Continuity Error Source Term τ_m at All Time Steps on All Grids	198
Figure 9.7: U Momentum Error Source Term τ_u at All Time Steps on All Grids	199
Figure 9.8: V Momentum Error Source Term τ_v at All Time Steps on All Grids.....	200
Figure 9.9: Profiles of the ETE Solution and True Error on all Grids at t=0.001	202
Figure 9.10: Profiles of the ETE Solution and True Error on all Grids at t=0.002.....	203
Figure 9.11: Profiles of the ETE Solution and True Error on all Grids at t=0.003.....	204
Figure 9.12: L2 Normal of the True Error as a Function of Grid Spacing.....	206
Figure 9.13: L2 Normal of the ETE Solution as a Function of Grid Spacing	207

List of Symbols

Φ	Generic scalar quantity in transport equations
ε	Error
n	Superscript representing numerical solutions
h	Grid spacing in Richardson Extrapolation Channel height in backward facing step
P	Order of accuracy in Richardson Extrapolation
r	Refinement factor in Richardson Extrapolation
L	Differential operator
T	Error source term
P	Fluid density
μ	Fluid dynamic viscosity
ν	Fluid kinematic viscosity
Γ	Diffusion coefficient
U	Velocity in the x coordinate direction
V	Velocity in the y coordinate direction
P	Fluid pressure
T	Time
W	Weighting function in the weighted spline method
a	Coefficient in the weighted spline method
Z	Final weighted spline function fit
λ	Radial basis function coefficient
L	Channel length

Chapter 1

Introduction

Since its conception, computational fluid dynamics (CFD) has had a role to play in both the industrial and the academic realm. While always an interesting research topic in academia, it has not been until relatively recently that CFD has truly become widely utilized in the industry. As processing speeds increase exponentially each year, the computational cost of real world CFD simulations of practical interest in the industry become more reasonable. It is this author's experience in the industry that numerical analysis, such as finite element analysis (FEA) and CFD, are becoming increasingly relied upon over experimental data. It is often the case that an unanticipated result on the first test of a new subsystem, or component, is completely unacceptable. Another unfortunate development with the utilization of CFD in the industry setting is that often times everyone (except for the CFD analyst themselves) sees the CFD results as an absolutely correct representation of the real world physics. This is due to a lack of understanding, by much of the industry, of the underlying methodology in CFD modeling.

Given the increasing role that CFD is now playing in the industrial setting, and the relative ease with which the results of the simulations are accepted, it is becoming increasingly important for the CFD analyst to have an in-depth understanding of the error inherently present in his models. At a minimum, CFD simulation results should be accompanied by some analysis of the error including some type of grid convergence study and an estimation of the numerical uncertainty. Additionally, all CFD codes should undergo verification and validation when applied to problems that involve drastically different physical phenomena.

1.1 Verification and Validation in CFD Applications

Given the ever increasing role that CFD simulations are playing in the industry, it is more important than ever that the community is diligent in formally establishing the credibility

of the CFD model. This credibility is established through the completion of model verification and validation. According to the AIAA Guide for the Verification and Validation of CFD Simulations [1], verification is defined as the process of determining if a computational simulation accurately represents the conceptual model. One must note that in this definition nothing is mentioned regarding the model's ability to accurately represent the real world physical phenomena. Conversely, validation is defined as the process of determining if a computational model accurately represents the real world. It should be noted that the AIAA guide does not establish any fixed level of credibility or required level of accuracy for a computational simulation. The required level of accuracy from the simulation is entirely dependent on the final application of the simulation result. If the simulation result is only meant to provide a qualitative description of a flow phenomenon that may not be readily observed in a system, then the acceptable level of accuracy may be quite low. If, however, the simulation results are intended to completely replace the experimental testing of an important system component then the credibility and accuracy of the model would need to be significantly greater.

1.2 Sources of Numerical Error

The primary focus of code verification is the quantification of errors present in the solution. All computational simulations are susceptible to five primary sources of error; (1) Mistakes in computer programming, (2) round off errors, (3) iterative convergence errors, (4) insufficient temporal discretization, and (5) insufficient spatial discretization.

As described by Authors such as Roy [2] and Roache [3], mistakes in computer programming fall under the category of code verification, while the four other sources of error fall under solution verification. According to the authors, code verification can be completed using the following strategies in order of increasing rigor. The least rigorous is expert judgment which entails the distribution of the code and a sample output to a subject matter expert who will provide his opinion on the quality of the simulation result. The next most rigorous method is the quantification of the error through comparison of the simulation output to either an exact analytic solution or a highly accurate benchmark numerical solution. There is no criterion on the limit of the error as it was noted earlier that the required credibility of the model is entirely subjective and dependent on the intended use of the output. Next is the completion of a consistency or convergence test. This involves the comparison of the simulation output to an exact analytic or benchmark

numerical solution on successively finer grids to prove that the discretization error shrinks as the element size is reduced. Finally, the most rigorous test for code verification is the order of accuracy. This test not only requires that the discretization error observed in the consistency test decrease with shrinking element size, but it must reduce at the expected rate based on the utilized discretization scheme.

The four remaining error sources fall under solution error and are present, to some degree, in all numerical simulations. The round off error is of the least concern. This error source was most problematic before 64-bit processing was available on nearly every computer. When computational simulations are completed with double precision, the round off error is practically negligible in most cases. Because nearly all computational simulations of practical interest require some form of iteration, the error associated with the iterative convergence must also be quantified. For the majority of cases with exact analytic solutions or highly accurate numerical solutions, the solution can be converged to a point where the iterative convergence error is negligible compared to other sources of error. However, for some complex simulations of practical interest in the industry, where schedule constraints are often problematic, a fully converged solution may not be achievable. This is especially the case in unsteady simulations where the solution must be converged at each time step. In these situations, the iterative convergence error must be accounted for.

This leaves the final and most significant sources of error in computational simulations, temporal and spatial discretization error. Given that round off error can be all but eliminated with double precision calculations, and iterative convergence error can be virtually eliminated with sufficient time to converge the solution, the discretization error remains to be the dominant error source. The discretization error is assessed by systematically reducing the grid and time step size. As the grid size is reduced, the discretization error should asymptotically approach zero. When considering a simple case for which an exact analytic solution exists the discretization error is simply calculated by subtracting the numerical solution from the exact solution. This same methodology also applies for highly accurate benchmark numerical solutions. However, because the analytic and benchmark solutions are only available for the most simplified problems, other methods have been developed for calculating the discretization error for more complex cases.

1.3 Discretization Error Prediction

Every numerical simulation suffers from some level of discretization error. As has been mentioned previously, there is no set level of acceptable error. The acceptable level of error is determined by the intended use of the solution output. The quantification of the discretization error is an important part of the verification and validation process. Additionally, it is a best practice to include a prediction of the discretization error with the final results of any computational simulation.

To this point the numerical analysis community has largely favored an a posteriori method for calculation of the discretization error, known as the Richardson Extrapolation (RE) method. Currently, RE is the most well accepted method for the prediction of discretization error. RE is capable of accurately predicting the discretization error in a wide range of numerical simulations as long as several conditions are met. While RE is the currently accepted best practice for the calculation of the discretization error, the method has enough disadvantages associated with it that significant research has taken place in the development of alternative options. One of these alternative options, known as the Error Transport Equation (ETE), will be the topic of this study, but first the current best practice, RE, will be covered in detail.

1.3.1 Richardson Extrapolation

RE is a commonly used discretization error prediction technique that is applied to the computational simulation solution a posteriori. RE is an extrapolation based discretization error estimator that utilizes solutions on multiple grids of successively finer grid spacing to calculate the extrapolated value of a dependent variable at zero grid spacing. Using RE, the discretization error can be predicted with a reasonable level of accuracy as long as certain conditions have been met. The methodology for the application of RE is similar regardless of which computational simulation was used to produce the solution. The method can be applied to finite-element, finite-volume, and finite-difference formulations. While these advantages have made RE commonplace in the numerical community, there are several disadvantages that keep RE from being the ideal solution to discretization error prediction. In the best case scenario, where the formal order of accuracy of the computational model is known, RE requires solutions on two grids of different grid sizes. If the formal order of accuracy is not known, the

observed order of accuracy must be solved for which introduces another unknown to the problem, thus requiring a numerical solution on a third grid of different grid spacing. To make matters worse, the solutions on the three different grids must be in the asymptotic regime of convergence in order to calculate a reliable value for the observed order of accuracy. For simple problems, performing multiple numerical simulations using varying grid spacing may not be an issue. However, for more complex flows of practical interest this can be time consuming. Additionally, the RE method has significant issues when oscillatory convergence is present. This shortcoming of RE has been thoroughly investigated in a paper by Celik et al. [4]. Finally, while RE works very well for smooth linear cases, there are challenges associated with cases involving nonlinearities, discontinuities, singularities, and complex physics. The advantages and disadvantages of RE extrapolation have been the topic of numerous papers in recent years [5–9] and will not be covered further in this section.

The RE method is based on the assumption that as long as the solution is in the asymptotic range (meaning that the spacing is small enough that all but a few terms in the Taylor series expansion for error can be neglected) the discretization error can be computed as

$$\varepsilon = \phi - \phi^n = ah^p \tag{1.1}$$

Where ε represents the discretization error, ϕ and ϕ^n represent the exact and numerical solutions, a is a coefficient that is independent of h , h is the grid spacing, and p is the order of accuracy. If the solution is not in the asymptotic regime the above equation contains higher order terms that are no longer negligible thus making the resulting error prediction less reliable. If two numerical solutions are generated in the asymptotic regime then the following equations can be developed and solved simultaneously to provide an estimation of the discretization error.

$$\begin{aligned}\varepsilon_1 &= \phi - \phi_1^n = ah^p \\ \varepsilon_2 &= \phi - \phi_2^n = a(rh)^p\end{aligned}\tag{1.2}$$

In the above equations, all variables with subscript 1 are derived from the fine grid, while all variables with the subscript 2 are derived from the coarse grid. Also, the variable r represents the grid refinement factor and is defined as the ratio of the coarse grid spacing to the fine grid spacing. The above equations can then be solved for the predicted discretization error in the fine grid solution resulting in the following equation.

$$\varepsilon_1 = \phi - \phi_1^n = \frac{\phi_1^n - \phi_2^n}{r^p - 1}\tag{1.3}$$

It must be noted that in the above equations the variable p represents the formal order of accuracy. The formal order of accuracy can only be used if there is an exact solution that can be used to verify that the numerical solution is in the asymptotic regime. Unfortunately, the vast majority of the time there is no exact solution for the problem under consideration and a solution on a third grid is required to calculate the observed order of accuracy. Consider a numerical simulation where solutions are available on fine, medium, and coarse grids. The following equations can be developed in a manner similar to those in Equation 1.2.

$$\begin{aligned}\varepsilon_1 &= \phi - \phi_1^n = ah^p \\ \varepsilon_2 &= \phi - \phi_2^n = a(rh)^p \\ \varepsilon_3 &= \phi - \phi_3^n = a(r^2h)^p\end{aligned}\tag{1.4}$$

In the above equations the subscripts 1, 2, and 3 represent the fine, medium, and coarse grids, respectively. These three equations can be solved to obtain an equation for the observed order of accuracy.

$$p = \frac{\ln\left(\frac{\phi_3^n - \phi_2^n}{\phi_2^n - \phi_1^n}\right)}{\ln(r)} \quad 1.5$$

This is a calculated observed order of accuracy, and as such, no assumptions regarding the order of accuracy need to be made. The caveat here is that if all of the solutions are not in the asymptotic regime, the calculated order or accuracy will not result in an accurate extrapolation of the dependent variable to zero grid spacing and thus the predicted error will not compare favorably to the true error. Further information on RE is available in a book by Roache [3] that reviews solution verification procedures.

1.3.2 Error Transport Equation

Due to the ever increasing roll that CFD is playing in engineering analysis today, much time and effort have been spent developing methods for discretization error prediction. Given that RE becomes less reliable and significantly more time consuming for the complex problems that are of practical interest today, more time is being invested in the development of alternative methods for discretization error prediction. One such method is the alternative equation method, or the error transport equation (ETE), as it will be referred to here.

This method is referred to as the error transport equation because the governing equations are manipulated in such a way as to develop a new conservation equation that can be solved in the same manner as the original conservation equations for the error. Consider an operator L that represents a partial differential equation (PDE) and the exact solution to that PDE ϕ .

$$L(\phi) = 0 \quad 1.6$$

Now allow L^n to represent another operator that is the discretized PDE. The numerical solution to that discretized equation is ϕ^n . For the time being, assume that the solution was allowed to iterate for such a time that the solution is fully converged resulting in a negligible residual.

$$L^n(\phi^n) = 0 \quad 1.7$$

From this point the derivation can proceed in two ways. The first would be to substitute the exact solution Φ into the discretized equation, L . Due to the discretization error, the right hand side of this equation would not be zero but some finite residual.

$$L^n(\phi) = \tau(\phi) \quad 1.8$$

Now subtracting Equation 1.7 from Equation 1.8 and defining the discretization error as $\varepsilon = \Phi - \phi^n$ yields the following error transport equation, provided that L^n is a linear operator.

$$L^n(\varepsilon) = \tau(\phi) \quad 1.9$$

Here the left hand side of the above equation represents the time rate of change, convection, and diffusion of the error, while the right hand side represents the source term for the error. There are numerous benefits to this method, including the potential of developing a general algorithm that is applicable to a wide range of problems. The problem with this method and all other methods involving the ETE is the development of

the error source term. Equation 1.8 defines the error source term in terms of the exact solution which is never known in problems of practical interest. The full methodology for the application of the above equations is discussed extensively in a paper by Celik et al. [10].

Backtracking slightly, the second derivation of the error transport equation is initiated by substitution of the numerical solution, $\widetilde{\phi}^n$, that has been mapped to a continuous solution, ϕ^n , into the original partial differential equation.

$$L(\phi^n) = \tau(\phi^n) \tag{1.10}$$

Again, because there are errors associated with the numerical solution, the original partial differential equation will not be satisfied and there will be a finite residual on the right hand side of the equation. Now if Equation 1.10 is subtracted from Equation 1.6, and the discretization error is defined as $\varepsilon = \phi - \phi^n$, the following PDE for the discretization error will result.

$$L(\varepsilon) = -\tau(\phi^n) \tag{1.11}$$

From here the PDE can be discretized using the same scheme that was used to solve the original problem. Again, as with the first derivation, the problem is the development of the error source term τ . In this case, the error source term is developed by evaluating the original partial differential equation with the numerical solution which is a discrete rather than a continuous variable.

The ETE method has several advantages over the more commonly used RE method. The most significant of these advantages is that the ETE method has the potential to accurately predict the error using the solution from only one grid. This represents a significant time saving over RE where calculations on a minimum of three grids are required. Often even more solutions are necessary to ensure that all solutions are in the

asymptotic regime. Additionally, if there is an interest in grid adaptation, refining the grid in the areas of the computational domain, where the error is actually generated rather than where it is finally transported to, could be advantageous [11]. The primary disadvantage associated with the ETE method is the difficulty in calculating the error source term. The calculation of the error source term is the topic of a great deal of research and the topic of the study presented here. An overview of the available literature on this topic will be presented in the following section.

1.4 Review of Error Transport Equation Literature

As the previous section should have made apparent, the general methodology behind the ETE method is quite straight forward. The conservation equation representing the transport of discretization error can easily be derived from the governing equation of interest, regardless of the discretization scheme. A very nice summary of several common discretization error estimators, including the ETE method, is presented in a book by Roache [12].

While the methodology behind the ETE equation is trivial, the details of the equation are where the problem lies, specifically, in the calculation of the error source term. There are difficulties associated with the development of the error source term regardless of whether the continuous or discrete ETE is derived. Recall from Equation 1.8 that the error source term can be calculated by substituting the exact solution into the discretized equation. If the exact solution is known, the source term is easily calculated. However, if the exact solution is known then why bother solving the ETE. In the typical case where the exact solution is not known, some type of approximation must be made. Several approaches have been developed to complete this task, many of which are discussed by Roache [12] and Celik [10].

Now recall from Equation 1.10 that the error source term can also be calculated by evaluating the original continuous PDE with the numerical solution. The problem here is that some approximations must be made in order to evaluate a continuous PDE with a discrete variable. It is this very problem that is the topic of this study. Numerous ideas for handling the calculation of the error source term for the continuous ETE have been developed with varying levels of success. Several of these studies will be discussed in

the remainder of this section to provide the reader with an understanding of the current research.

In a paper presented by Van Straalen et al. [13] the authors derive the error transport equation from first principles for the one- and two-dimensional advection diffusion equations. The ETE was then solved, and the predicted error was compared to true error calculations developed using exact analytic solutions to the advection diffusion equations. Similar to the methods developed in this study, the authors define the error source term as the residual of the original PDE when evaluated with the numerical solution. To accomplish this task, the numerical solution was mapped onto a continuous distribution using a piecewise linear function that corresponds to central differencing. The predicted error showed satisfactory qualitative and quantitative agreement for both first and second order numerical solutions. Van Straalen later extended upon this work by applying his method to the Navier-Stokes equations for backward facing step flow. While Van Straalen does show analytically that the method can be extended to the coupled, non-linear, conservation equations the resulting predicted error did not compare favorably to the true error. Van Straalen concludes that the poor agreement is likely attributed to the poor performance of the central differencing method employed as a residual estimator.

In a paper published by Zhang et al. [14], the authors utilize the continuous ETE to estimate the error in a one-dimensional, non-linear system of hyperbolic conservation laws. Here the error source term is again defined as the residual of the original PDE when evaluated using the numerical solution. However, here the residual is estimated using a modified equation analysis and accounting for the flux difference across cell interfaces. The discretization error was predicted for four sample cases possessing exact analytical solutions: (1) a linear advection equation, (2) a non-linear Burgers equation, (3) the unsteady Euler equations, and (4) the steady Euler equations with variable cross-section. The authors conclude that the solutions of the ETE offer a good approximation of the true error in all cases considered. Zhang also makes the observation that the error transport equation can be particularly useful in grid adaptation. Typically, the grid is adapted based on some prediction of the discretization error. This method only accounts for where the errors reside in the final solution, but does not take into account the transport of the error from its original source.

In a paper by Ilinca et al. [15], the authors compare the performance of three different discretization error estimators on several incompressible flow problems. Discretization error was predicted first using the RE method and solutions on three grids so that the observed order of accuracy could be used. Discretization error was also estimated by calculating the difference between the numerical solution and a higher order reconstruction of the exact solution using a least squares method. Finally, the discretization error is predicted by applying the methodology developed by Zhang et al. using the modified equation analysis to estimate the error source term. Discretization error was predicted using the three described methods for subsonic radial flow, supersonic vortex flow, and supersonic confluence flow. In all cases, the error predictions from the higher order reconstruction of the exact solution using least squares were in the poorest agreement with the true error. For the first two test cases (subsonic radial flow and supersonic vortex flow), the RE methods provided the best prediction of the discretization error. However, the results of the ETE were in good agreement as well. For the third case (supersonic confluence flow), RE did not perform as well, likely because of the discontinuities present in the flow. The ETE method performed well when the confluence flow was solved with a first order discretization scheme, but performed worse when the flow was solved with a second order scheme. The authors hypothesize that it may be beneficial to ensure that the ETE is solved with a higher order discretization than the original flow problem.

In a later paper by Celik and Hu [16], the authors again utilize the error transport equation to predict the discretization error in one-dimensional steady and transient convection diffusion problems, as well as two-dimensional steady and unsteady convection diffusion problems. Here Celik and Hu also make use of the modified equation, applied to the discretized equations, to estimate the error source term. The modified equation is used in conjunction with a Taylor series expansion of each node in the discretized equation about the central node. The resulting evaluation of the error source term requires the user to have access to the influence coefficients of the original discretized equation. The method presented here results in good approximations of the true error. The authors conclude that, for the non-linear cases examined in the study, it is not necessary to solve the ETE equation with a higher order discretization scheme than the original flow.

In the paper by Hay and Visonneau [17], the continuous ETE equation is utilized, in a manner similar to the methodology presented here, to predict the discretization error in several test cases including two-dimensional laminar flow over a circular cylinder. Hay utilizes a higher-order reconstruction of the numerical solution to evaluate the original PDE to obtain an estimate of the error source term. A bicubic polynomial fit in a least squares sense is used to facilitate the reconstruction. Hay notes that care must be taken near the boundaries to ensure that the original boundary conditions are satisfied in the reconstruction. The resulting discretization error predictions are in very good agreement with the true error. The authors also note that while the ETE method is quite efficient with reference to CPU time for uncoupled conservation equations there is some computation cost when the method is applied to coupled equations such as the Navier-Stokes equations.

In a paper by Cavallo and Sinha [18], the methods employed by Zhang et al. (discussed above) were extended to three-dimensional external aerodynamic flows. As in the study by Zhang, the error source term is approximated by using a modified equation analysis and accounting for the flux difference across cell interfaces. The authors compare the computational results to experimental data by plotting the numerical solution with error bars calculated using the ETE method. The results appear to be favorable for all cases considered as the experimental data provided does fall within the established error bars. The authors imply that viscous diffusion of the error was not included in the ETE. It is also noted that for future work the diffusion terms should be included.

1.5 Objectives and Motivations of This Study

As has been discussed at length thus far in this study, and in the previous studies referenced in the literature review, the derivation and solution of the ETE is straight forward. The error transport equation is easily derived from first principles using the original PDE. Once discretized, the error transport equation can be solved using the same solver that was used in the solution of the original partial differential equation. The real problem with the implementation of the ETE method for the prediction of discretization error is the development of the error source term.

The problem of developing the error source term has been examined by numerous researchers from numerous angles. A comprehensive summary of these methods has

been discussed in the literature review. The most commonly implemented method for calculating the error source term is by evaluating the original partial differential equation (or sets of partial differential equations) with the numerical solution. To facilitate this method the numerical solution must be mapped from the discretized domain to the continuous domain. Numerous authors did have a reasonable amount of success with this method in the prediction of discretization error for one- and two-dimensional linear equations. The results were seldom as favorable when predicting error for coupled non-linear equations, such as the Navier-Stokes equations. When the predicted discretization error was not in satisfactory agreement with the true error, the most commonly cited concern was the insufficient quality of the method used to map the numerical solution to the continuous domain.

A method was developed by Junkins et al. [19] to model irregular surfaces. The original intent of the method was to model large sets of topographical data. The method uses a weighting function to blend together a series of localized fits. The weighting functions are implemented in such a way as to guarantee C^n continuity (continuity of the original function and the first n derivatives) at all localized boundaries. Going forward, this author shall refer to this method as the weighted spline method (WSM). The WSM has numerous beneficial properties. The localized fits that are blended together are not constrained in anyway. The local data could be fit with any scheme desired. Because of this, the quality of the fit can be altered based on application. The level of continuity can also be varied based on need. This method was later used by Roy and Sinclair [20] in what the authors term the method of nearby problems. Their method utilized the WSM to generate realistic manufactured solutions for use in verification and validation as well as discretization error prediction. To the best of the author's knowledge, this study is the first to utilize the WSM for the calculation of discretization error.

The presentation of this study in using the WSM in conjunction with the ETE method for the prediction of discretization error will proceed in the following manor. First a brief overview of the verification problems used in this study will be presented, followed by a brief description of the WSM, and a more in-depth review of the error transport equation. Finally, data from a wide range of sample cases completed for verification and validation of this method will be presented. Conclusion will be drawn and recommendations for future work will be made in order to facilitate the further development of this topic.

Chapter 2

Governing Equations and Methodology

Before the computational error of a numerical model can be assessed, a solution must be calculated. Numerical solutions are calculated through the discretization of the corresponding governing conservation equations. In this study, the discretization error of increasingly complex numerical models was assessed through the use of the ETE.

The study begins with the solution and error prediction of the 1-D advection diffusion equation. Once it became apparent that the solution of the ETE yielded satisfactory results for this case, the same analysis was applied to the 2-D advection diffusion equations. Finally, a similar method was applied to the incompressible Navier-Stokes equations. What follows in each sub-section is a brief summary of the governing conservation equations that apply to each of the fluid flows considered in this study. Also presented is the corresponding error transport equation for each of the governing equations.

2.1 Steady 1-D Advection Diffusion

The steady 1-D advection diffusion conservation equation is the least complex problem considered in this study, and serves as an excellent starting point for the evaluation of the error transport equation methodology. The 1-D advection diffusion equation takes the following form.

$$\frac{d}{dx}(\rho u \phi) - \frac{d}{dx}\left(\Gamma \frac{d\phi}{dx}\right) = 0 \quad 2.1$$

In the above equation, ρ represents the density, u represents the velocity, Γ represents the diffusivity, and ϕ a general property that is being transported. For the purposes of

this study, the density, velocity, and diffusivity will all be held constant, resulting in the following governing conservation equation.

$$u \frac{d\phi}{dx} - \Gamma \frac{d^2\phi}{dx^2} = 0 \quad 2.2$$

In the above equation, the constant density has been included into the diffusivity term.

2.1.1 Steady 1-D Advection Diffusion Error Transport Equation

The methodology for deriving the error transport equation is similar for all governing conservation equations. However, all conservation equations include different variables, and thus, the final error transport equation is different depending on what conservation equation is considered.

If the governing differential equation provided in equation 2.2 were to be solved exactly for ϕ , then the right hand side of the equation would remain zero if the solution ϕ were substituted back into equation 2.2. Now assume a numerical solution, namely ϕ^n , were solved for using a discretized equation instead of the original differential equation. If the solution ϕ^n was substituted into the original differential equation, then the right hand side of equation 2.2 would no longer be zero, but would be some finite number representing some residual. This residual is a result of the discretization error inherent in any numerical model. In equation form this would be described as

$$u \frac{d\phi^n}{dx} - \Gamma \frac{d^2\phi^n}{dx^2} = \tau \quad 2.3$$

The variable τ in the above equation represents the error transport source term. Now if equation 2.3 is subtracted from equation 2.2, and the quantity $\phi - \phi^n$ is defined as the

solution error, and designated as ε , the error transport equation results and is given by the following.

$$u \frac{d\varepsilon}{dx} - \Gamma \frac{d^2\varepsilon}{dx^2} = -\tau \quad 2.4$$

At this point, equation 2.4 can be discretized and solved on the same grid as the original conservation equation. The error transport equation source term has been calculated in many different ways in multiple studies, and will be discussed in great detail in a later section.

2.2 Steady 2-D Advection Diffusion

To extend the evaluation of the error transport equation methodology considered in this study, the error transport equation was applied to the prediction of error for a steady 2-D advection diffusion equation. The 2-D governing conservation equation is analogous to the corresponding 1-D form and is provided below.

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} - \Gamma \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) = 0 \quad 2.5$$

While the 2-D advection diffusion equation does not appear to be significantly more complicated than its 1-D counterpart, the analytic solution of the equation is made significantly more difficult by the additional variables. The analytic solution considered in this study will be presented in the results section of this document.

2.2.1 Steady 2-D Advection Diffusion Error Transport Equation

The error transport equation for the steady 2-D advection diffusion equation is derived in a manner similar to the 1-D advection diffusion equation. First, the numerical solution is

substituted into the governing conservation equation resulting in a residual on the right hand side. The result of this substitution is provided below.

$$u \frac{\partial \phi^n}{\partial x} + v \frac{\partial \phi^n}{\partial y} - \Gamma \left(\frac{\partial^2 \phi^n}{\partial x^2} + \frac{\partial^2 \phi^n}{\partial y^2} \right) = \tau \quad 2.6$$

Next, equation 2.6 is subtracted from equation 2.5 resulting in the following ETE for the steady 2-D advection diffusion equation.

$$u \frac{\partial \varepsilon}{\partial x} + v \frac{\partial \varepsilon}{\partial y} - \Gamma \left(\frac{\partial^2 \varepsilon}{\partial x^2} + \frac{\partial^2 \varepsilon}{\partial y^2} \right) = -\tau \quad 2.7$$

Again, this error transport equation can be solved on the same grid as the corresponding numerical solution.

2.3 Navier-Stokes Equations

The two cases presented above serve as excellent starting points for the evaluation of the computational error prediction methods presented in this work. However, the prediction of error in the modeling of 1-D and 2-D advection diffusion equations is of little practical use. Most practical applications of CFD today require the resolution of the velocity in all directions, as well as the pressure and any other scalar quantities that may be required by the model under consideration (temperatures, species concentrations, etc.). This requires the solution of the coupled, non-linear, set of PDEs, namely the continuity equation, momentum equation, and the energy equation.

For the purposes of this work, the energy equation will not be considered as all fluids will be treated as isothermal, incompressible fluids with constant viscosity, thus uncoupling the energy equation from the continuity and momentum equations. Thus, the following 2D coupled equations will be discretized and solved to obtain a numerical solution.

$$\text{Continuity Equation:} \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad 2.8$$

$$\text{X-Momentum Equation:} \quad \rho \left[\frac{\partial u}{\partial t} + \frac{\partial(uu)}{\partial x} + \frac{\partial(vu)}{\partial y} \right] + \frac{\partial P}{\partial x} - \mu \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] = 0 \quad 2.9$$

$$\text{Y-Momentum Equation:} \quad \rho \left[\frac{\partial v}{\partial t} + \frac{\partial(uv)}{\partial x} + \frac{\partial(vv)}{\partial y} \right] + \frac{\partial P}{\partial y} - \mu \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right] = 0 \quad 2.10$$

In the above equations ρ , represents the fluid density, u and v represent the fluid velocity in the x coordinate direction and y coordinate direction, respectively, P represents the pressure, and μ represents the fluid dynamic viscosity. Very few analytic solutions exist for this equation set, and the ones that do are for a very specific set of initial and boundary conditions.

2.3.1 Error Transport Equation for 2-D Navier Stokes Equations

While the method from which the Navier-Stokes error transport equation shall be derived is the same as that for the 1-D and 2-D advection diffusion equations, the final result is significantly more complicated. Because the Navier-Stokes equations are coupled and non-linear, the resulting error transport equations will also be coupled and non-linear. However, as with the advection diffusion equations, this is not a complication computationally as the form of the derived error transport equations is quite similar to their governing conservation equations.

The easiest of the three error transport equations to derive is the continuity error transport equation due to its linearity. If the numerical solutions for the u and v velocities are substituted into the governing differential equation, the right hand side of equation 2.8 will no longer be zero. The following equation results.

$$\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y} = \tau_m \quad 2.11$$

In the above equation, u^n and v^n represent the numerical solutions of the velocity in the x and y coordinate directions, respectively, and τ_m represents the residual that results from the substitution. This residual will ultimately become the source term in the continuity error transport equation. Now, if equation 2.11 is subtracted from equation 2.8, and the variable ε is allowed to represent the solution error as $(\varepsilon_u, \varepsilon_v) = (u, v) - (u_n, v_n)$, the continuity error transport equation results as follows.

$$\frac{\partial \varepsilon_u}{\partial x} + \frac{\partial \varepsilon_v}{\partial y} = -\tau_m \quad 2.12$$

The method for deriving the x-momentum error transport equation is very similar to the method used to derive the continuity error transport equation. However, the non-linear nature of the momentum equation introduces some additional terms into the derivation that must be treated with care.

As expected, one must first substitute to numerical solutions for the u velocity, v velocity, and pressure into the governing conservation equation 2.9, which will result in a similar residual term on the right hand side of the equation.

$$\rho \left[\frac{\partial u^n}{\partial t} + \frac{\partial (u^n u^n)}{\partial x} + \frac{\partial (v^n u^n)}{\partial y} \right] + \frac{\partial P^n}{\partial x} - \mu \left[\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2} \right] = \tau_u \quad 2.13$$

In the above equation, u^n and v^n represent the numerical solutions for the velocities in the x and y coordinate directions, respectively, P^n represents the numerical solution for pressure, and τ_u represents the resulting residual. Again, this residual will ultimately become the source term in the x-momentum error transport equation.

The derivation continues by subtracting equation 2.13 from the original governing conservation equation 2.9, resulting in the following equation.

$$\rho \left[\frac{\partial(u - u^n)}{\partial t} + \frac{\partial(uu - u^n u^n)}{\partial x} + \frac{\partial(vu - v^n u^n)}{\partial y} \right] + \frac{\partial(P - P^n)}{\partial x} - \mu \left[\frac{\partial^2(u - u^n)}{\partial x^2} + \frac{\partial^2(u - u^n)}{\partial y^2} \right] = -\tau_u \quad 2.14$$

Now if the solution error is defined as $(\varepsilon_u, \varepsilon_v, \varepsilon_P) = (u, v, P) - (u^n, v^n, P^n)$, and equation 2.14 is rearranged, the result is the complete x-momentum error transport equation.

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + \frac{\partial \left(2u^n \varepsilon_u + \overset{A}{\varepsilon_u^2} \right)}{\partial x} + \frac{\partial \left(v^n \varepsilon_u + u^n \varepsilon_v + \overset{B}{\varepsilon_u \varepsilon_v} \right)}{\partial y} \right] + \frac{\partial(\varepsilon_P)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] = -\tau_u \quad 2.15$$

Equation 2.15 represents the full x-momentum error transport equation. The above equation can be simplified by eliminating the two higher order error terms noted as A and B. This is a reasonable simplification as the higher order terms will not typically contribute significantly to the final solution. After some rearranging, the error transport equation takes on a form very similar to the x-momentum equation.

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + 2 \frac{\partial(u^n \varepsilon_u)}{\partial x} + \frac{\partial(v^n \varepsilon_u)}{\partial y} \right] + \frac{\partial(\varepsilon_P)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] + \rho \frac{\overset{A}{\partial(u^n \varepsilon_v)}}{\partial y} = -\tau_u \quad 2.16$$

A brief inspection of the above equation reveals that the error transport equation is very similar in form to the x-momentum equation. All of the terms are identical with three exceptions. The x derivative in the convection term is multiplied by a factor of two. This arises because of the non-linearity of the u velocity in the x-momentum equation. There is also an additional term on the left hand side of the equation noted as A in equation 2.16. This term arises from the y derivative in the convection term of the x-momentum equation, and serves to couple the x-momentum and y-momentum error transport equations. Finally, there is a source term on the right hand side of the equation which arises from the substitution of the numerical solution into the original governing conservation equation. Because the form of the error transport equation is so similar to the x-momentum equation, it can be solved with the same numerical methods as the x-momentum equation with minimal modifications.

The method for deriving the y-momentum error transport equation is identical to the method used to derive the x-momentum error transport equation. As a result, the terms in the final error transport are also analogous. The full derivation of the y-momentum error transport equation will be covered as well for completeness.

As expected, one must first substitute numerical solutions for the u velocity, v velocity, and pressure into the governing conservation equation 2.10, which will result in the familiar residual term on the right hand side of the equation.

$$\rho \left[\frac{\partial v^n}{\partial t} + \frac{\partial(u^n v^n)}{\partial x} + \frac{\partial(v^n v^n)}{\partial y} \right] + \frac{\partial P^n}{\partial y} - \mu \left[\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2} \right] = \tau_v \quad 2.17$$

In the above equation, u^n and v^n represent the numerical solutions for the velocities in the x and y coordinate directions, respectively, P^n represents the numerical solution for pressure, and τ_v represents the resulting residual. Again, this residual will ultimately become the source term in the y-momentum error transport equation.

The derivation continues by subtracting equation 2.17 from the original governing conservation equation 2.10 resulting in the following equation.

$$\rho \left[\frac{\partial(v - v^n)}{\partial t} + \frac{\partial(uv - u^n v^n)}{\partial x} + \frac{\partial(vv - v^n v^n)}{\partial y} \right] + \frac{\partial(P - P^n)}{\partial y} - \mu \left[\frac{\partial^2(v - v^n)}{\partial x^2} + \frac{\partial^2(v - v^n)}{\partial y^2} \right] = -\tau_v \quad 2.18$$

Now if the solution error is defined as $(\varepsilon_u, \varepsilon_v, \varepsilon_P) = (u, v, P) - (u^n, v^n, P^n)$ and equation 2.18 is rearranged, the result is the complete y-momentum error transport equation.

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial \left(u^n \varepsilon_v + v^n \varepsilon_u + \overbrace{\varepsilon_u \varepsilon_v}^A \right)}{\partial x} + \frac{\partial \left(2v^n \varepsilon_v + \overbrace{\varepsilon_v^2}^B \right)}{\partial x} + \frac{\partial(\varepsilon_P)}{\partial y} \right] - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] = -\tau_v \quad 2.19$$

Equation 2.19 represents the full y-momentum error transport equation. The above equation can be simplified by eliminating the two higher order error terms noted as A and B. This is a reasonable simplification for the same reasons that applied to the x-momentum error transport equation. After some rearranging, the error transport equation takes on a form very similar to the y-momentum equation.

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial(u^n \varepsilon_v)}{\partial x} + 2 \frac{\partial(v^n \varepsilon_v)}{\partial y} \right] + \frac{\partial(\varepsilon_P)}{\partial y} - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] + \overbrace{\rho \frac{\partial(v^n \varepsilon_u)}{\partial x}}^A = -\tau_v \quad 2.20$$

A brief inspection of the above equation reveals that the error transport equation is very similar in form to the y-momentum equation. All of the terms are identical with three

exceptions. The y derivative in the convection term is multiplied by a factor of two. This arises because of the non-linearity of the v velocity in the y -momentum equation. There is also an additional term on the left hand side of the equation noted as A in equation 2.20. This term arises from the x derivative in the convection term of the y -momentum equation and serves to couple the x -momentum and y -momentum error transport equations. Finally, there is a source term on the right hand side of the equation which arises from the substitution of the numerical solution into the original governing conservation equation. Because the form of the error transport equation is so similar to the y -momentum equation, the discretized error transport equation can be solved with the same numerical methods as the y -momentum equation with minimal modifications.

Chapter 3

Weighted Spline Interpolation

In this chapter, the weighted spline method (WSM) for modeling and interpolation is introduced. The WSM serves as a means to fit a discrete set of data with a series of analytic equations over multiple spline regions. The WSM was originally developed to model topographic surfaces [19]. When provided with a set of discrete data, the domain is divided into multiple sub regions. The sub regions are joined together to form individual spline zones. These spline zones are fit with polynomials and blended together using weighting functions (derivation of the weighting functions will be provided later). The final analytical equations are differentiable with C^n order continuity (the function is continuous as well as the n th derivative). The level of continuity is determined by the development of the weighting function. In what follows, the methodology will be provided for the 1-D WSM followed by an example. After the 1-D WSM is presented the 2-D WSM will conclude the section.

3.1 One-Dimensional Weighted Spline Method

When provided with a set of discrete data points, the WSM can be used to fit the data with a series of polynomials blended together with weighting functions (will be defined in the following section) such that C^n continuity (continuity of the original function and n^{th} derivative at the spline boundaries) is achieved. The order of continuity is determined by the derivation of the weighting function.

The first step in the implementation of the WSM is to divide the computational domain into a set of sub-regions. These sub-regions are then combined to form a series of overlapping spline-regions. Each interior spline-region is composed of two sub-regions while the boundary spline-regions are composed of only one sub-region. It is these spline regions that will be fit with polynomials and blended together with the weighting functions. Some care needs to be taken in the selection of the number of sub regions. The accuracy of the final fit is improved with increasing number of sub regions. However,

each spline-region must contain enough data points such that the desired polynomial order may be achieved in each region. The limiting spline-regions are often the boundary regions as they are only composed of one sub-region, and thus, typically contain the fewest data points. Figure 3.1 shows an example of a discrete set of data with the domain divided into sub-regions and then combined to form the spline-regions. Sub-regions are provided as numbers, and spline regions are provided as roman numerals.

Figure 3.1: A set of discrete data divided into sub-regions (1-4) and spline-regions (I-V)

At this point, each spline region is fit with a polynomial. The polynomial can be of any desired order as long as there is enough data in each spline region to apply the fit. For the purposes of this example, a quadratic has been fit to each of the overlapping spline regions. The results are plotted in Figure 3.2.

Figure 3.2: Quadratic Fit of the Discrete Data Over Each of the Five Overlapping Spline Regions

Now that the data in each spline-region has been fit with a polynomial, the final fit over each sub-region can be determined. This is accomplished through the use of weighting functions and equation 3.1.

$$Z_m(x) = P_{left}(x)W_{left}(\bar{x}) + P_{right}(x)W_{right}(\bar{x}) \quad 3.1$$

Where Z_m is the final analytic equation fit over sub-region m . P_{left} and P_{right} represent the polynomial fit in the left and right overlapping spline-regions, respectively. W_{left} is the weighting function applied to the left polynomial. W_{left} varies from 1 on the left side of sub-region m to 0 on the right side of sub-region m . W_{right} is the weighting function applied to the right polynomial. W_{right} varies from 0 on the left side of sub-region m to 1

on the right side of sub-region m . The overbars on x in the weighting function indicate that the independent variables are normalized in each sub-region. The weighting functions are derived in such a way that W_{left} is simply a transformation of W_{right} . This is shown in equation 3.2. Plots of the weighting functions for each sub-region are provided in Figure 3.3 below (weighting functions from the paper by Junkins [19]).

$$W_{left}(\bar{x}) = W_{right}(1 - \bar{x}) \quad 3.2$$

Figure 3.3: Left and Right Weighting Functions for Each Sub-Region

After proper application of equation 3.1, each sub-region is fit with a continuous function. These functions are continuous at the boundaries of the sub-regions in value, as well as the derivatives. The level of continuity of the function is determined by the selected weighting function. The weighting functions can be formulated to provide any level of continuity. Additionally, as each spline region is fit with a simple polynomial and, as will be shown in the next section, the weighting functions are rather simple expressions. The final fit in each sub-region is easily differentiated analytically to determine higher order

derivatives. The use of numerical differencing schemes is not required. The final fit of the data in this example is plotted in Figure 3.4.

Figure 3.4: Final fit of Discrete Data Using WSM

3.1.1 Development of One-Dimensional Weighting Functions

One of the primary benefits of the WSM is that the final fit function is C^n continuous at the sub-region boundaries. The level of continuity at the sub-region boundaries is determined by the particular form of the weighting functions used (W in equation 3.1 and equation 3.2). The weighting function takes the general form provided in equation 3.3.

$$W_{right} = x^{n+1} \left(\sum_{i=0}^n a_i x^i \right) \quad 3.3$$

Notice that the general form of the weighting function automatically satisfies the constraint that the function and its n^{th} derivatives must equal zero at $x = 0$. The following

additional constraints apply to the weighting function over each sub-region after the independent variable in the sub-region has been normalized.

- W_{right} must equal *unity* at $x = 1$
- The n^{th} derivatives of W_{right} must equal *zero* at $x = 1$

It is important to note that in the derivation of the weighting functions, not all levels of desired continuity result in a unique solution of the weighting function. However, some of the lower order polynomials do have unique solutions [19]. There is no need to select a weighting function with a level of continuity higher than what is desired for the problem at hand, as it has no influence on the accuracy of the final fit. A list of derived weighting functions, up through a C^3 level of continuity, is provided below in Table 3.1. It is only a simple matter of solving a series of linear simultaneous equations to determine a weighting function of higher order, if needed.

Table 3.1: One-Dimensional Weighting Functions

Continuity Level	Weighting Function
C^1	$x^2(3-2x)$
C^2	$x^3(10-15x+6x^2)$
C^3	$x^4(35-84x+70x^2-20x^3)$

3.2 Two-Dimensional Weighted Spline Method

With the one-dimensional WSM fully explained in section 3.1, it is not difficult to extend the methodology to a two-dimensional space. The overarching methodology is exactly the same in two dimensions as in one dimension. However, the specific details of the process are slightly more complicated (fitting 2-D polynomials, development of 2-D weighting functions, more complicated spline-region overlapping, etc.). All of this will be covered in the following sections.

The two-dimensional domain is still divided into a user specified number of sub-regions. As with the one-dimensional method, the accuracy of the final fit is improved with an increasing number of sub regions. The tradeoff for this increased accuracy is increased computational time, which obviously becomes more of an issue when considering a 2-D space. These sub regions are again combined to form overlapping spline-regions in which data will be fit with a two-dimensional polynomial, in a least squares sense. Each interior spline-region is composed of four sub-regions. The boundary spline-regions are composed of two sub-regions, while the corner spline-regions each contain only one sub-region. Because of this, the corner spline-regions will be a limiting factor when selecting the number of sub-regions and the order of the polynomial fit. The corner spline-regions will contain the least number of points, and thus, care must be taken to ensure that each one has enough points to adequately define the selected polynomial.

The final analytic fit in each sub-region is generated by blending the polynomial fit in four different spline-regions using weighting functions similar to the one-dimensional methodology. The size and shape of the four spline regions is dependent on the location of the sub-region under consideration. Interior sub-regions make use of four spline-regions which contain four sub-regions each. Side sub-regions make use of two spline-regions which are composed of four sub-regions, and two spline-regions which are composed of two sub-regions. Corner sub-regions make use of one spline-region composed of four sub-regions, two spline-regions composed of two sub-regions, and one spline-region which contains only one sub-region. For simplicity, these spline regions are always numbered I - IV beginning in the north-east corner and progressing counter clockwise around the sub-region under consideration. The above description is represented visually in Figure 3.5 through Figure 3.7. Each figure contains an image of the entire sample domain. The highlighted sub-region is the one over which the final fit will be valid. The other four images in the figure represent the four overlapping spline-regions whose polynomial fits will be blended together with the weighting function.

Figure 3.5: Spline-Region Overlapping for an Interior Sub-Region

	II		I
	III		IV

II	

	I

III	

	IV

Figure 3.6: Spline-Region Overlapping for a Side Sub-Region

II	I		
III	IV		

II

	I

III

	IV
--	----

Figure 3.7: Spline-Region Overlapping for a Corner Sub-Region

Using the above overlapping scheme for each sub-region, a polynomial (of user specified order) is fit to each of the spline-regions. These polynomials are blended using weighting functions to form the final overall fit for each sub-region. The equation for the final fit is provided in equation 3.4.

$$Z_m(x, y) = P_I(x, y)W_I(\bar{x}, \bar{y}) + P_{II}(x, y)W_{II}(\bar{x}, \bar{y}) \\ + P_{III}(x, y)W_{III}(\bar{x}, \bar{y}) + P_{IV}(x, y)W_{IV}(\bar{x}, \bar{y}) \quad 3.4$$

Where Z_m is the final analytic equation fit over sub-region m , P_{I-IV} represents the polynomial fit in the corresponding overlapping spline-regions. W_{I-IV} is the weighting function applied to the corresponding polynomials. W_I varies from the value of 0 for all $x=0$ and all $y=0$ to the value of 1 at $x=1$ and $y=1$. The overbars on x and y in the weighting functions indicate that the independent variables are normalized within each sub-region. The weighting functions are derived in such a way that W_{II-IV} are simply transformations of W_I . This is shown in equation set 3.5. Plots of the weighting functions for each sub-region are provided in Figure 3.8.

$$W_{II}(\bar{x}, \bar{y}) = W_I(1 - \bar{x}, \bar{y})$$

$$W_{III}(\bar{x}, \bar{y}) = W_I(1 - \bar{x}, 1 - \bar{y}) \quad 3.5$$

$$W_{IV}(\bar{x}, \bar{y}) = W_I(\bar{x}, 1 - \bar{y})$$

After proper application of equation 3.4, each sub-region is fit with a continuous function. These functions are continuous at the boundaries of the sub-regions in value, as well as derivative. The level of continuity of the function is determined by the derivation of the weighting function. The weighting functions can be derived to provide any level of continuity. Because each spline region is fit with a simple polynomial and, as will be shown in the next section, the weighting functions are rather simple expressions, the

final fit in each sub-region is easily differentiated analytically to determine higher order derivatives. The use of numerical differencing schemes is not required.

Figure 3.8: Contours of Weighting Functions Applied to Corresponding Spline-Regions

3.2.1 Development of Two-Dimensional Weighting Functions

One of the primary benefits of the WSM is that the final fit function is C^n continuous at the sub-region boundaries. The level of continuity at the sub-region boundaries is determined by the form of the weighting functions (W_{I-IV} in equation 3.4 and equation 3.5). The weighting function takes the general form given in equation 3.6.

$$W_I = x^{n+1}y^{n+1} \left(\sum_{i,j=0}^n a_{i,j} x^i y^j \right) \quad 3.6$$

Notice that the general form of the weighting function automatically satisfies the constraint that the function and its n^{th} derivatives must equal *zero* along the lines $x = 0$ and $y = 0$. The following additional constraints apply to the weighting function over each sub-region after the independent variables in the sub-region have been normalized.

- W_I must equal *unity* at the point $(x = 1, y = 1)$
- The n^{th} derivatives with respect to x and y of W_I must equal *zero* at the point $(x = 1, y = 1)$

It is important to note that, in the derivation of the weighting functions, not all levels of desired continuity result in a unique solution of the weighting function. However, some of the lower order polynomials do have unique solutions. There is no need to select a weighting function with a level of continuity higher than what is desired for the problem at hand as it has no influence on the accuracy of the final fit. Two derived weighting functions are provided below in Table 3.2 up through a C^3 level of continuity. It is only a simple matter of solving a series of linear simultaneous equations to determine a weighting function of higher order if needed.

Table 3.2: Two-Dimensional Weighting Functions

Continuity Level	Weighting Function
C^1	$x^2y^2(9 - 6x - 6y + 4xy)$
C^3	$x^4y^4(1225 - 2940y + 2450y^2 - 700y^3 - 2940x + 7056xy - 5880xy^2 + 1680xy^3 + 2450x^2 - 5880x^2y + 4900x^2y^2 - 1400x^2y^3 - 700x^3 + 1680x^3y - 1400x^3y^2 + 400x^3y^3)$

3.3 Application of Radial Basis Functions for Local Spline-Region Data Fits

The implementation of the WSM, while fitting the data in the local spline-regions with two-dimensional polynomials in a least squares sense, produces good results in most cases. However, there are several shortcomings that could potentially be improved upon. First of all, the level of discretion left to the user regarding the order of the polynomial fit in the spline-regions is significant. This also factors into the selection of the number of sub-regions to divide the original computational domain into. As a rule, the more sub-regions that the computational domain is divided into, the better the final fit. However, it can also be shown that using a higher order polynomial to fit the spline region data results in a better overall fit as well. However, the higher the order of the polynomial the more points that are required to calculate the coefficients. If the set of discrete data is on a very coarse mesh, it can be difficult to find a balance between the two options. This is not desirable as the ultimate goal with a discretization error prediction method is to have a general application that could be implemented into industrial CFD calculations.

Another issue with using polynomial fits in the spline regions is that the polynomial is fit in the least squares sense, thus implying that the data from the fit does not necessarily have to agree with the original data set. While the error is usually relatively small, this could become problematic when calculating the error source term near the boundaries, as it is important that the mapped data still satisfies the boundary conditions. Because of this, some additional function is required to force the value of the boundary conditions. This methodology was required in the work completed by Hay and Visonneau [17] in their implementation of the bicubic polynomial.

Because of the implementation of the WSM, there is no reason that the data in the local spline-regions must be fit with a polynomial in the least squares sense. Many of the issues addressed above can be resolved by implementing a radial basis function (RBF) for the data fit in the local spline-regions. RBF methods allow for a set of scattered data to be fit with an analytic function. The smooth analytic function also allows for interpolation between the scattered data points. These methods were first studied by Roland Hardy [21] as a means to study topographical surfaces. However, these methods have more recently been used to represent very complex 3D surfaces [22].

3.3.1 Radial Basis Function Methodology

The methodology behind the RBF method is fairly straight forward and is just as easily implemented in three dimensions as in one. This is because the final RBF fit is a function of only one variable, r , no matter what number of dimensions the method is applied to. The single variable r corresponds to the Euclidian normal of the difference between two points. Therefore, given a set of N scattered data points at locations $\mathbf{x}$ (which can be in one, two, or three dimensional space) with values $\mathbf{y}$ the final RBF fit takes the following form.

$$s(x_j) = \sum_{i=1}^N \lambda_i \phi(\|x_j - x_i\|) \quad 3.7$$

Where in the above equation λ_i are influence coefficients associated with each data point. The Euclidian normal is represented by $\|x_j - x_i\| = \sqrt{(x_{j,1} - x_{i,1})^2 + (x_{j,2} - x_{i,2})^2}$. The continuous function ϕ is known as the basis function. This function could take any desired form but, there are several common forms that are often used. A few of these are included in Table 3.3 below for reference purposes. The basis function controls the distance from which each point exerts its influence. Some basis functions, such as the thin plate, are known as global functions and allow each point to be influenced by all others. Other basis functions, such as the Gaussian, are local functions and only allow each point to be influenced by neighboring points. Most local basis functions have a free parameter designated as σ which allows the user to adjust the radius of influence. For the purposes of this study the thin plate was selected as it does not require any attention to the free parameter. The optimum value of the free parameter is loosely a function of grid spacing and is not always straight forward to set. Additionally, the global basis function tends to result in a smoother overall fit as each point in the final fit is being influenced by all points in the domain.

Table 3.3: Common Radial Basis Functions

Form	Function (ϕ)
Linear	r
Cubic	r^3
Thinplate	$r^2 \ln(r+1)$
Gaussian	$\exp\left(-\frac{r^2}{2\sigma^2}\right)$
Multiquadrics	$\sqrt{1 + \frac{r^2}{\sigma^2}}$

The RBF coefficients, λ_i , are solved for using the provided scattered data and ensuring that $s(x_j) = y_j$. This leads to the following linear system of equations that can be solved for the RBF coefficients, λ_i .

$$B\lambda = y^c_i \quad 3.8$$

Here, B is an $N \times N$ matrix with entries that take the following form.

$$b_{ij} = \phi(\|x_i - x_j\|) \quad 3.9$$

The resulting set of linear equations to be solved simultaneously takes the following form.

$$\begin{aligned}
y_1 &= \lambda_1 \phi(\|x_1 - x_1\|) + \lambda_2 \phi(\|x_2 - x_1\|) + \dots + \lambda_N \phi(\|x_N - x_1\|) \\
y_2 &= \lambda_1 \phi(\|x_1 - x_2\|) + \lambda_2 \phi(\|x_2 - x_2\|) + \dots + \lambda_N \phi(\|x_N - x_2\|) \\
y_N &= \lambda_1 \phi(\|x_1 - x_N\|) + \lambda_2 \phi(\|x_2 - x_N\|) + \dots + \lambda_N \phi(\|x_N - x_N\|)
\end{aligned} \tag{3.10}$$

Once the coefficients are determined using the above system of linear equations the value of the smooth function can be determined anywhere in the domain using equation 3.7.

3.3.2 Comparison of Polynomial WSM and RFB WSM

In order to assess the performance both the polynomial and RBF variants of the WSM, the presented methodologies were applied to two analytic functions. Both functions are the same trigonometric function with a constant that was varied to change the resulting gradients. These functions are provided in the equation below.

$$\begin{aligned}
f &= \sin(2\pi x^2) \cos(2\pi y^2) \\
f &= \sin(4\pi x^2) \cos(4\pi y^2)
\end{aligned} \tag{3.11}$$

By changing 2π to 4π the resulting solution has much larger gradients and is thus much more difficult to fit with a smooth function. These analytic functions were evaluated over a computation domain that included $0 < x < 1$ and $0 < y < 1$. A mesh plot of the resulting functions is provided in Figure 3.9.

$$f = \sin(2\pi x^2)\cos(2\pi y^2)$$

$$f = \sin(4\pi x^2)\cos(4\pi y^2)$$

Figure 3.9: Analytic Functions Used in The Evaluation of the WSM

Both methodologies were first applied to the 2π function as the smaller gradients were easier to fit. Both variants of the WSM were applied to the analytic function several times using an increasing number of sub-regions. For the polynomial variant, consideration was given to 3rd, 4th, and 5th order polynomials. Because the number of points required in each sub-region increases with the order of the polynomial the maximum number of sub-regions varied. The RBF variant has no limitation on the number of sub-regions (a significant advantage over the polynomial variant of the WSM) so the maximum number of sub-regions was limited to the maximum number allowed for the 3rd order polynomial.

In order to qualitatively evaluate the each fit, mesh plots have been provided in Figure 3.10 through Figure 3.12. Plots are provided for all variants of the WSM using 9, 36, and 100 sub-regions. A mesh plot of the analytic function is provided in each figure for comparison purposes. It is readily apparent when using only 9 sub-regions that there are noticeable errors in all of the polynomial variants. However, the RBF variant appears to perform exceedingly well. As the number of sub-regions is increased, all of the polynomial variants begin to perform better. When using 100 sub-regions, the qualitative error in all of the polynomial variants is no longer noticeable. The most important takeaway in these plots is the steady performance of the RBF variant. While the polynomial variants of the WSM do perform well with larger number of sub-regions, the RBF variant performs well regardless of the number of sub-regions. This is a very nice feature as it requires less user modification on a case by case basis.

Figure 3.10: Data Fit From All Variants of the WSM with 9 Sub-Regions

$$f = \sin(2\pi x^2)\cos(2\pi y^2)$$

Figure 3.11: Data Fit From All Variants of the WSM with 36 Sub-Regions

$$f = \sin(2\pi x^2)\cos(2\pi y^2)$$

Figure 3.12: Data Fit From All Variants of the WSM with 100 Sub-Regions

$$f = \sin(2\pi x^2)\cos(2\pi y^2)$$

Application of the same methodology to the 4π function yields similar results with a few exceptions. Mesh plots of the resulting data fits have been provided in Figure 3.13 through Figure 3.15. The primary difference is that more sub-regions are required before any of the polynomial variants of the WSM perform reasonably well. The resulting data fits from the polynomial variants when using only 9 sub-regions are almost entirely unusable. However, the RBF variant of the WSM again performs exceedingly well even with only 9 sub-regions. Even when utilizing 36 sub-regions there are still obvious levels of qualitative error in all of the polynomial variants. Once the number of sub-regions is increased to 100, the 4th order and 5th order polynomial variants no longer have a noticeable level of qualitative error. However, the error in the 3rd order variant is certainly still noticeable. Again, the RBF variant of the WSM produces a fit with no apparent qualitative error regardless of the number of sub-regions. The value of this cannot be overstated as the level of simplicity afforded to the user is an important consideration during practical application.

Figure 3.13: Data Fit From All Variants of the WSM with 9 Sub-Regions

$$f = \sin(4\pi x^2)\cos(4\pi y^2)$$

Figure 3.14: Data Fit From All Variants of the WSM with 36 Sub-Regions

$$f = \sin(4\pi x^2)\cos(4\pi y^2)$$

Figure 3.15: Data Fit From All Variants of the WSM with 100 Sub-Regions

$$f = \sin(4\pi x^2) \cos(4\pi y^2)$$

In order to provide a more quantitative analysis of the performance of all variants of the WSM, the L2 normal of the error between the analytic function and the WSM was calculated. The resulting plots are provided in Figure 3.16 and Figure 3.17 for the 2π and 4π analytic functions respectively. After a brief inspection of these plots it is readily apparent that the RBF variant performs much better than the polynomial variant of the WSM. Regardless of what order polynomial is used and how many sub-regions are used the RBF variant of the WSM results in significantly smaller errors in the fit of the original function. In the polynomial variant, much attention must be paid to the order of the polynomial selected and the required number of sub-regions to perform the fit. The 5th order polynomial requires more data points so fewer sub-regions are allowed in this case than in the 3rd order polynomial. However, the 5th order polynomial performs better than the 3rd order polynomial despite not being able to use as many sub-regions. The RBF variant of the WSM outperforms all other variants by a significant margin. Regardless of the number of sub-regions used in the fit, the RBF variant outperforms the polynomial variants. This confirms the qualitative observations made earlier in this section.

Because the implementation of the continuous error transport equation requires the calculation of up to second derivatives of the numerical solution, it is important to evaluate each method's ability to predict the first and second derivatives. Plots of the L2 normal of the error in the first and second derivative with respect to the x coordinate direction and the first and second derivative with respect to the y coordinate direction are provided in Figure 3.18 through Figure 3.25 for both analytic functions respectively. After an inspection of the L2 normal plots several observations could be made. Most importantly, as with the original function fit, the calculated derivatives from the RBF variant of the WSM are in much better agreement with the analytic derivatives than the polynomial variants. As anticipated, the polynomial variants' derivative estimations appear to improve as the number of sub-regions increase. However, it is interesting to note that the derivative predictions from the RBF variant seem to get slightly worse as the number of sub-regions increase. At this time it is believed that this phenomenon is due to the slightly decreased performance of the RBF near the edges of the domain being fit. As you add more sub-regions there are more locations where the edge effects are present in the overall fit. Ultimately however, the RBF variant outperforms the polynomial variants regardless of the number of sub-regions utilized.

Figure 3.16: L2 Normal of the Error in All Variants of the WSM When Applied to $f = \sin(2\pi x^2)\cos(2\pi y^2)$

Figure 3.17: L2 Normal of the Error in All Variants of the WSM When Applied to $f = \sin(4\pi x^2)\cos(4\pi y^2)$

Figure 3.18: L2 Normal of the Error in the 1st X Derivative When Applied to $f = \sin(2\pi x^2)\cos(2\pi y^2)$

Figure 3.19: L2 Normal of the Error in the 1st X Derivative When Applied to $f = \sin(4\pi x^2)\cos(4\pi y^2)$

Figure 3.20: L2 Normal of the Error in the 2nd X Derivative When Applied to $f = \sin(2\pi x^2)\cos(2\pi y^2)$

Figure 3.21: L2 Normal of the Error in the 2nd X Derivative When Applied to $f = \sin(4\pi x^2)\cos(4\pi y^2)$

Figure 3.22: L2 Normal of the Error in the 1st Y Derivative When Applied to $f = \sin(2\pi x^2)\cos(2\pi y^2)$

Figure 3.23: L2 Normal of the Error in the 1st Y Derivative When Applied to $f = \sin(4\pi x^2)\cos(4\pi y^2)$

Figure 3.24: L2 Normal of the Error in the 2nd Y Derivative When Applied to $f = \sin(2\pi x^2)\cos(2\pi y^2)$

Figure 3.25: L2 Normal of the Error in the 2nd Y Derivative When Applied to $f = \sin(4\pi x^2)\cos(4\pi y^2)$

The implementation of several variants of the WSM have been explained and compared in this section. Based on the results it appears that the RBF variant has many advantages over the polynomial variant. First of all and most importantly, the RBF variant produces a more accurate fit of the original data and all of its derivatives. In many cases, the improvement in the accuracy of the fit is several orders of magnitude. Other benefits of the RBF variant of the WSM include its robustness. The RBF variant performs well over a wide range of conditions and does not require the user to spend time balancing the order of the polynomial and the number of sub-regions. Also, because of the methodology of the RBF variant, the values of the numerical solution at the boundaries are preserved exceedingly well. This is not the case with the polynomial variant which would require some additional function be applied in order to meet this criteria. In many ways the RBF variant of the WSM is superior to the polynomial variants.

Chapter 4

Applications to Steady 1-D Advection

Diffusion Equation Solutions

A method has been proposed to predict numerical discretization error using the continuous error transport equation. The WSM will be used to map the numerical solution to the continuous domain and calculate the error transport equation source term. In order to demonstrate the ability of this method to accurately predict the numerical discretization error, the methodology shall be applied to several cases with either known analytical solutions, or highly accurate benchmark solutions. The first, and simplest, of these is the case of the 1-D advection diffusion equation which will be discussed in this chapter.

4.1 Problem Description and Analytic Solution

An ideal starting place for the validation of the methods presented here is the case of steady 1-D Advection diffusion. This problem governs with the transport of some scalar quantity (species concentration as an example) by means of both fluid motion and diffusion through the medium. The transport processes in the steady 1-D advection fusion problem are governed by the following conservation equation.

$$u \frac{d\phi}{dx} - \Gamma \frac{d^2\phi}{dx^2} = 0 \tag{4.1}$$

Where u represents the velocity and Γ is the diffusivity.

If one considers the case of a fluid, with both a constant velocity and diffusivity, there exists an exact analytic solution. This analytic solution can be used here to calculate the

exact numerical discretization error and evaluate the error transport equation solution. If the following boundary conditions are applied,

$$\begin{aligned}\Phi(0) &= \Phi_0 \\ \Phi(L) &= \Phi_L\end{aligned}\tag{4.2}$$

then the resulting analytic solution is

$$\phi(x) = \left(\frac{\phi_L - \phi_0}{e^{Pe} - 1} \right) \left[e^{(Pe \frac{x}{L})} - 1 \right] + \phi_0\tag{4.3}$$

Where Φ_0 is the scalar value at the west boundary, Φ_L is the scalar value at the right boundary, L represents the total domain length, Pe is the Peclet number which is defined as $Pe=uL/\Gamma$.

4.2 Numerical Solution

4.2.1 Description of Domain, Solver, and Boundary Conditions

For the current study, the steady 1-D advection diffusion problem was solved in a domain that ranged from 0 to 1 in the x-coordinate direction. The computational domain was discretized using 41 points spaced equally across the domain. This number of grid points was selected such that enough data points were available to provide a reasonable fit from the WSM, while also resulting in a reasonably large magnitude of discretization error.

The steady 1-D advection diffusion equation was solved using an in-house model written in MATLAB. The solver utilizes the finite volume method with 2nd order central

differencing for the diffusion term and a 1st order upwind differencing scheme for the convection term.

The boundary conditions were applied such that the scalar value at the west boundary was held fixed at $\Phi_0=1.0$, while the scalar value at the east boundary was held fixed at $\Phi_L=0.0$. For the case considered in this study, the diffusivity was held fixed at a value of 0.1.

The final discretized equations were solved using the Tridiagonal Matrix Algorithm (TDMA) and, as such, required no residual monitoring.

4.2.2 Solution Verification

In order to verify the current solver, a brief grid dependence study was conducted to ensure that the discretization error was reducing at the expected rate as the grid spacing was refined. To facilitate this study, the velocity was set such that the Peclet number had a value of 20. Solutions were calculated on grids with 21, 41, and 81 grid points. As anticipated, the numerical solution began to more accurately represent the analytic solution as the grid was refined. The numerical solution from each grid is proved in Figure 4.1 along with the analytic solution for a Peclet number of 20. The profile of the true error for each solution is provided in Figure 4.2. Notice that the numerical error does indeed decrease as the grid spacing is refined. The L2 normal of the true error was also calculated for each grid and plotted in Figure 4.3. Notice that the L2 normal of the true error reduces proportionally to the grid spacing raised to a power of 1.3552. As anticipated, this exponent is approximately equal to the order of the numerical model.

Figure 4.1: Numerical Solutions on Grids of Increasing Refinement

Figure 4.2: True Error on Grids of Increasing Refinement

Figure 4.3: L2 Normal of the True Error as a Function of Grid Spacing

4.2.3 Solutions

Using the grid composed of 41 points, solutions were obtained for Peclet numbers of 10, 20, and 100. These Peclet numbers correspond to velocities of 1.0, 2.0, and 10.0. As the Peclet number is increased, the flow becomes more and more convection dominated. The result is that the value of Φ at the west boundary is swept further downstream, resulting in a more extreme gradient at the east boundary.

The numerical solutions for each Peclet number are plotted below in Figure 4.4 through Figure 4.6, along with the corresponding analytic solution. The numerical solution is usually in very good agreement with the analytic solution close to the west boundary and, slowly begins to diverge from the analytic solution as the east boundary is approached. The maximum magnitude of the true error (occurring near the east boundary) tends to increase with Peclet number due to the larger gradients present at the east boundary.

Figure 4.4: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=10$

Figure 4.5: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=20$

Figure 4.6: Analytic and Numerical Solution to the Steady 1-D Advection Diffusion Problem for $Pe=100$

4.3 Solution of the Error Transport Equation

Using first principles, and the original governing partial differential equation, the ETE is derived, taking the following form.

$$u \frac{d\varepsilon}{dx} - \Gamma \frac{d^2\varepsilon}{dx^2} = -\tau \quad 4.4$$

In equation 4.4, τ is calculated by mapping the numerical solution to the continuous domain using the WSM, and evaluating the original conservation equation 4.1 with the mapped continuous numerical solution. The resulting error source terms, for all Peclet numbers considered, are plotted in Figure 4.7 through Figure 4.9. An exceedingly interesting observation can be made by a brief inspection of the error source term profiles. Given that the numerical error is known to increase with increasing Peclet number, it would be reasonable to hypothesize that the error source would also increase with increasing Peclet number. However, this is not the case in the calculated profiles. As the Peclet number increases, the magnitude of the error source term actually decreases. This raises a very interesting point regarding the difference between the source of the error and the error itself. The error that is ultimately reported is actually where the error ends up after the error sources are transported according to the error transport equation. In this case, while the magnitude of the error source actually decreases with increasing Peclet numbers that smaller error is actually convected downstream into a smaller region. This is because of the substantially increased convection in the cases with higher Peclet numbers. This conclusion would certainly come into play when considering how to refine a grid based on an error analysis. Ideally, the analyst would want to refine the grid in regions of high error source, not necessarily where the error ultimately ends up.

The resulting ETE is solved using the same grid and the same discretization scheme as was used to solve the original PDE. The resulting predicted discretization errors are plotted in Figure 4.10 through Figure 4.12, along with the corresponding true error.

An inspection of the resulting plots of the discretization error yields several interesting observations. The primary observation is that, for all Peclet numbers considered in this study, the predicted discretization error is in very good qualitative and quantitative agreement with the true error. The shape of the predicted discretization error is an excellent representation of the true error. As such, the solution of the ETE accurately captures the location of the maximum error. The magnitude of the maximum error is also accurately represented. As anticipated, the location of the maximum error shifts further downstream as the convection becomes more dominant with increasing Peclet number. Additionally, the magnitude of the maximum error increases with increasing Peclet number. Both of these trends are accurately captured in the ETE solution. Overall, the method considered in this study performed very well for the steady 1-D advection diffusion case.

Figure 4.7: Error Source Term for $Pe=10$

Figure 4.8: Error Source Term for $Pe=20$

Figure 4.9: Error Source Term for $Pe=100$

Figure 4.10: True Error and ETE Solution for $Pe=10$

Figure 4.11: True Error and ETE Solution for $Pe=20$

Figure 4.12: True Error and ETE Solution for $Pe=100$

Chapter 5

Applications to Steady 2-D Advection

Diffusion Equation Solutions

Given the success of the ETE methodology when applied to the steady 1-D advection diffusion equation, the next logical step was the application of the same methodology to the steady 2-D advection diffusion equation. The governing conservation equation solved in this section is simply the 2-D extension of the advection diffusion equation solved in the previous section. The applied methodology is essentially identical with the exception that the scalar will, and the resulting error will, be solved in two dimensions rather than just one. The primary difference in the methodology is that the development of the error source term will now require the use of the 2-D weighted spline method.

5.1 Problem Description and Analytic Solution

This section focuses on the extension of the 1-D advection diffusion equation to a 2-D space. Again, the problem deals with the transport of some scalar quantity (species concentration as an example) by means of both fluid motion and diffusion through the medium, now in both the x and y directions. The transport processes in the steady 2-D advection fusion problem are governed by the following conservation equation.

$$u \frac{d\phi}{dx} + v \frac{d\phi}{dy} - \Gamma \frac{d^2\phi}{dx^2} - \Gamma \frac{d^2\phi}{dy^2} = 0 \quad 5.1$$

Where u and v represent the velocity in the x and y coordinate directions, respectively, and Γ is the diffusivity.

One unfortunate issue with the extension of the advection diffusion equation to two dimensions is that analytic solutions become significantly more difficult to develop. Few analytic solutions exist, and the ones that do are for a very specific set of conditions. One such solutions exists for the case of a unit point source at the origin (0,0) subjected to flow in the x coordinate direction alone. The derivation of the closed form analytic solution to this problem is rather lengthy, and as such, will not be presented here. For a complete derivation of the solution, the reader can reference the work by Van Straalen [13]. The resulting analytic solution is given by the following expression.

$$\phi(x, y) = e^{\frac{Ux}{2\Gamma}} K_0 \left[\frac{U}{2\Gamma} (x^2 + y^2)^{0.5} \right] \quad 5.2$$

Where K_0 is a zero degree modified Bessel function of the second kind.

Because of the use of the modified Bessel function, the exact solution is not defined at the origin. This will be an important consideration when generating the computation grid, as the domain will be shifted a small amount in the positive x direction. In this study, the u velocity will be set to a value of 0.15, and the diffusion coefficient will be set to 0.01. Substituting these variables into the analytic expression results in the distribution of the scalar quantity provided in the contour plot in Figure 5.1.

Figure 5.1: Contour Plot of the Analytic Solution for 2-D Advection Diffusion with a Point Source

5.2 Numerical Solution

5.2.1 Description of Domain, Solver, and Boundary Conditions

For the solution of the steady 2-D advection diffusion problem, a computational domain was created such that $0.025 < x < 2.0$ and $-0.5 < y < 0.5$. As was mentioned previously, the computational domain was shifted 0.025 in the positive x direction because the analytic solution, which utilizes a zero degree modified Bessel function of the second kind, is not valid at the origin.

The steady 2-D advection diffusion equation was solved using an in-house code written in MATLAB. The solver utilizes the finite volume method with 2nd order central differencing for the diffusion term and a 1st order upwind differencing scheme for the convection term.

Because the analytic solution is known at all x and y locations, dirichlet boundary conditions are applied at all boundaries. The value of the scalar is specified by the value of the analytic solution at all boundaries. As was mentioned previously, the u velocity was set to a constant value of 0.15, and the diffusion coefficient was set to a value of 0.01.

Because there are now two dimensions to consider, the final discretized equation cannot be solved directly with TDMA, as was the case with the 1-D problem. Here, the final discretized equation is solved using the TDMA line by line iteration method. The solver alternates sweeping direction from iteration to iteration to allow the boundary conditions to propagate through the solution as efficiently as possible.

5.2.2 Solutions

As this section is considering a simple 2-D extension of the problem presented in the previous section, a numerical solution was only calculated on one grid. The computational domain was discretized into 41 grid points in both the x and y coordinate directions. This results in a grid spacing of $dx=0.049375$ in the x direction (the odd value of grid spacing is the result of the domain beginning at 0.025) and a grid spacing of $dy=0.025$ in the y direction.

The resulting solution is shown in the contour plot provided in Figure 5.2. For comparison purposes, a contour plot of the analytic solution is also provided in the same figure. In order to provide a more complete presentation of the solution, profiles of both the numerical and analytical solutions have been provided at several different x locations in Figure 5.3. An inspection of the contours plots gives a good representation of the flow field. The largest scalar values are near the origin where the point source is located. The scalar is transported through the domain as would be anticipated through convection and diffusion. The profiles indicate that the numerical solution is a very close approximation of the analytic solution. The boundary condition at the west face matches the analytic solution exactly as it should. The largest error seems to occur at the center plan of the computational domain at all axial locations.

Analytic Solution

Numerical Solution

Figure 5.2: Contour Plots of the Analytical and Numerical Solutions for 2-D Advection Diffusion with a Point Source

$X=0.025$

$X=0.51875$

$X=1.0125$

$X=1.5062$

Figure 5.3: Profiles of Analytical and Numerical Solutions for 2-D Advection Diffusion With a Point Source

5.3 Solution of the Error Transport Equation

Using first principles, and the original governing partial differential equation, the ETE is derived, taking the following form.

$$u \frac{d\varepsilon}{dx} + v \frac{d\varepsilon}{dy} - \Gamma \frac{d^2\varepsilon}{dx^2} - \Gamma \frac{d^2\varepsilon}{dy^2} = -\tau \quad 5.3$$

In equation 5.3, τ is calculated by mapping the numerical solution to the continuous domain using the WSM, and evaluating the original conservation equation 5.1 with the mapped continuous numerical solution. A contour plot of the resulting error source term is provided in Figure 5.4. While it is difficult to select a scale which will cover the maximum and minimum values of the source term, as well as provide good resolution across the entire domain, it is still obvious that the magnitude of the error source term is highest close to the point source. This is anticipated as locations with large gradients tend to result in the largest error.

The resulting ETE is solved using the same grid and the same discretization scheme as was used to solve the original PDE. Contour plots of the true error and the ETE solution are provided in Figure 5.5. A comparison of the contour plots indicates a strong qualitative and quantitative agreement between the true error and the ETE solution. The general shape of the ETE solution is in very good agreement with the true error.

A more complete picture of the error distribution can be provided through the inspection of error profiles rather than a contour plot. Profiles of the true error, and the ETE solution, have been provided at several different x locations in Figure 5.6 and Figure 5.7. Again, the profiles yield several interesting observations. Paying careful attention to the errors at the boundaries, it is evident that the boundary conditions of the ETE have been accounted for appropriately. Because the value of the scalar is implemented directly at the boundary in the numerical solution, there is zero error at all boundary locations. The profiles, as with the contour plot, indicate that the ETE solution matches the shape of the true error exceedingly well. The locations of maximum and minimum error are matched almost exactly. The magnitude of the true error is also predicted very closely in the ETE

solution. Another interesting observation is that the ETE appears to more accurately predict the true error in regions where the error is larger. This is a trend that will frequently be encountered throughout this study. Ultimately, the performance of this methodology in regions of large error is what is truly important.

The ETE methodology has been applied to the 2-D advection diffusion equation, and the solution has been compared to the calculated true error. The predicted error is in excellent agreement with the true error in both shape and magnitude. Having shown that the methodology works well with 1-D and 2-D scalar conservation equations, the next step will be the application of the method when a system of coupled, non-linear, PDEs are solved to calculate the entire flow field.

Figure 5.4: Error Source Term for 2-D Advection Diffusion With a Point Source

True Error

ETE Solution

Figure 5.5: Contour Plots of the True Error and ETE Solution for 2-D Advection Diffusion With a Point Source

$X=0.025$

$X=0.32125$

$X=0.6175$

$X=0.91375$

Figure 5.6: Profiles of True Error and ETE Solution for 2-D Advection Diffusion With a Point Source

$X=1.21$

$X=1.5063$

$X=1.7531$

$X=2.0$

Figure 5.7: Profiles of True Error and ETE Solution for 2-D Advection Diffusion With a Point Source

Chapter 6

Applications to Developing Flow Between Parallel Plates

Thus far, it has been demonstrated that the methods presented in this study yield accurate predictions of the numerical error in the case of the steady advection diffusion problem in both 1-D and 2-D. The simple solutions presented to this point have all assumed a known and constant velocity. This has served to demonstrate the feasibility of the method. Ultimately, for this method to be of practical use, it must be demonstrated that the methodology also works when solving for the complete flow field, in addition to any scalar quantities. To account for all of the flow variables, including velocity in all directions, as well as the pressure, one must solve the set of coupled, non-linear PDEs known as the Navier-Stokes equations.

6.1 Problem Description and Analytic Solution

There are relatively few exact analytical solutions to the Navier-Stokes equations. The exact solutions that do exist are based on idealized flows with numerous simplifications. One such solution is for fully developed flow between parallel plates. The problem is presented schematically in Figure 6.1.

Figure 6.1: Schematic Representation of Flow Between Parallel Plates

The fully developed assumption implies an infinitely long channel ($L \gg h$) in which the flow is no longer varying in the x direction, and there is no flow in the y direction. These assumptions allow the full continuity and momentum equations to be reduced to the following simplified forms.

$$\begin{aligned}\frac{\partial u}{\partial x} &= 0 \\ 0 &= -\frac{\partial P}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2}\end{aligned}\tag{6.1}$$

In the above equation, u represents the velocity in the x direction, P represents the pressure, and μ represents the fluid viscosity. Assuming that the flow has fully developed, and that the fluid viscosity is constant, an exact analytical solution exists. Application of the no-slip condition at the south and north walls results in the following boundary conditions for the U velocity.

$$\begin{aligned}U(0) &= 0 \\ U(h) &= 0\end{aligned}\tag{6.2}$$

The analytical solution for the U velocity distribution between two parallel plates is

$$u(y) = (y^2 - hy) \left(\frac{1}{2\mu} \frac{\partial P}{\partial x} \right)\tag{6.3}$$

6.2 Numerical Solution

6.2.1 Description of Domain, Solver, and Boundary Conditions

Rather than implement the known u velocity distribution at the inlet of the domain, it was decided to allow the solver to develop the flow and solve the error everywhere in the computational domain. While the error will be solved for everywhere in the domain, the solution to the error transport equation will only be compared to the true error in the fully developed region where an exact analytic solution exists. The computational domain for this numerical solution was chosen such that the channel height was set to $h=1.0$ and the channel length was set to $L=20.0$. The large channel length-to-height ratio ensures that the flow is fully developed at the channel exit.

The flow field between the parallel plates was solved using an in-house developed code written in MATLAB. The solver utilizes the finite volume method with 2nd order central differencing used to discretize the diffusion terms and a hybrid 2nd order scheme on the convection terms which applies upwind or central differencing based on the cell Peclet number. The pressure-velocity coupling was handled using the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm.

The boundary conditions were applied based on the desire to solve for developing flow. At the west boundary (channel inlet), the u velocity was set to a uniform velocity profile with a value of 1.0. The east boundary (channel outlet) condition was set to an outflow (or fully developed) boundary condition assuming zero u velocity gradient in the x direction. No slip boundary conditions were applied at the south and north walls by setting the u velocity equal to zero at both boundaries. The v velocity was set equal to zero at all boundaries, per the conditions of developing flow between parallel plates.

The final discretized equations were solved using the TDMA line by line iteration method. The solver alternates sweeping direction from iteration to iteration to allow the boundary conditions to propagate through the flow field as efficiently as possible.

6.2.2 Solutions

For purposes of completeness, and to serve as a means of validation of the in-house flow solver, solutions were generated on three increasingly refined computational grids. The coarse grid solution was solved on a 40x10 (x and y coordinates, respectively) grid. The computation grid was refined by a factor of two for each successive solution. This resulted in medium and fine grids with 80x20 and 160x40 points, respectively.

This study is only considering solutions in the laminar regimes. As such, a Reynolds number of 100 was selected. This low Reynolds number flow both ensures that the flow is laminar, as well as resulting in a shorter development length, thus allowing for a shorter computational domain to achieve fully developed flow. The density was set to a value of 1.0, while the viscosity was set to a value of 0.01. Given the inlet velocity of 1.0, the resulting Reynolds number is 100. This is based on the Reynolds number being defined as $Re = \rho u h / \mu$. The velocity component in this definition of Reynolds number is assumed to be the average velocity, which is equal to 2/3 of the maximum velocity in fully developed flow between parallel plates.

Solutions were generated on all grids. No limit was placed on the number of iterations in the SIMPLE algorithm, thus the solver was allowed to iterate until fully converged. This ensures that the residuals do not need to be taken into account when solving the error transport equation.

As this is a fully developed flow, and the v velocity is zero everywhere, no plots of the v velocity will be presented. Contour plots of the u velocity on each grid are provided in Figure 6.2. Fully developed u velocity profiles compared to the analytic solution for each grid are provided in Figure 6.3 through Figure 6.5. As anticipated, the u velocity profiles near the channel outlet are fully developed, and match the exact analytic solution well. Also as anticipated, as the grid spacing is reduced, the numerical solution approaches the exact solution indicating that the numerical solution will converge to the analytic solution in the limit of zero grid spacing.

40x10

80x20

160x40

Figure 6.2: U Velocity Contours for Numerical Solutions on Multiple Grids

Figure 6.3: Analytical and Numerical U Velocity Profile on a 40x10 Grid

Figure 6.4: Analytical and Numerical U Velocity Profile on a 80x20 Grid

Figure 6.5: Analytical and Numerical U Velocity Profile on a 160x40 Grid

6.3 Solution of the Error Transport Equation

Using first principles, and the original governing partial differential equations, the ETE is derived. Unlike the solution of the one-dimensional advection diffusion equation, there are three coupled partial differential equations that must be solved.

Continuity Error Transport Equation

$$\frac{\partial \varepsilon_u}{\partial x} + \frac{\partial \varepsilon_v}{\partial y} = -\tau_m \quad 6.4$$

U Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + 2 \frac{\partial(u^n \varepsilon_u)}{\partial x} + \frac{\partial(v^n \varepsilon_u)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] + \rho \frac{\partial(u^n \varepsilon_v)}{\partial y} = -\tau_u \quad 6.5$$

V Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial(u^n \varepsilon_v)}{\partial x} + 2 \frac{\partial(v^n \varepsilon_v)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial y} - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] + \rho \frac{\partial(v^n \varepsilon_u)}{\partial x} = -\tau_v \quad 6.6$$

Where τ_m , τ_u , and τ_v are calculated by mapping the numerical solution to the continuous domain using the WSM and evaluating the original conservation equations.

Contour plots of the continuity error transport source term have been provided for all grids in Figure 6.6. Contour plots of the u and v velocity error transport source terms have been provided for all grids in Figure 6.7 and Figure 6.8, respectively. It is difficult to select an appropriate scale for the contour plots, as all of the error source terms take on very small values (1e-6 or less) in the vast majority of the computational domain. However, all of the error source terms take on comparatively larger values near the inlet. This intuitively makes sense as it would be reasonable to assume that the largest source of error would be present in the developing region of the flow near the channel inlet rather than in the fully developed region near the outlet.

Figure 6.6: Contour Plots of the Continuity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids

40x10

80x20

160x40

Figure 6.7: Contour Plots of the U Velocity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids

40x10

80x20

160x40

Figure 6.8: Contour Plots of the V Velocity Error Transport Source Term on 40x10, 80x20, and 160x40 Computational Grids

With the error source terms calculated, the resulting ETE is solved using the same grid and discretization scheme as was used to solve the original PDE. The resulting solution for numerical error in the u velocity agrees exceedingly well with the calculated true error. Given that the analytic solution being utilized here is only valid in the fully developed region of the flow, the true error can only be calculated very near the outlet. The true error is calculated at $x=19.5$, which is the first coordinate inside the east boundary on the 40×10 grid. The true error profiles are plotted along with the error transport equation solution in Figure 6.9 through Figure 6.11.

An inspection of the error profiles yields several interesting observations. The first, and most apparent, observation is the strong qualitative and quantitative agreement of the ETE solution with the true error. The predicted error profile is in nearly perfect agreement with the true error profile. Also, while implied by the previous observation, it is worth noting that the solution to the ETE does scale down as anticipated with decreasing grid size. In order to better illustrate this point, the L2 normal of the true error and ETE solution is calculated, and plotted, versus the grid spacing in the y direction in Figure 6.12. As anticipated, based on the strong agreement of the error profiles, the L2 normal of the true error, and the ETE solution, agree almost perfectly. A second important observation is that the L2 normal of both the true error and the ETE solution decrease at the same rate as the grid spacing is reduced. The true error reduces proportionally to the grid spacing raised to a power of 2.5347, and the ETE solution reduces proportionally to the grid spacing raised to a power of 2.5289. In addition to confirming the strong agreement between the true error and the ETE solution, this also serves as a validation of the flow solver as one would expect the true error to reduce proportionally to the order of the model. The solver utilized for this problem makes use of 2nd order discretization on both the diffusion and convection terms. Thus, the true error decreases consistently with what would be anticipated from this solver.

6.3.1 Boundary Condition Considerations

Another important observation regarding boundary conditions was made while solving the ETE for the flow between parallel plates. While applying the ETE methodology to the case of the 1-D and 2-D advection diffusion, the only variable being solved for was a scalar quantity. The flow field variables, such as velocity, were assumed to be known, thus, only requiring a solution on a single grid. When actually solving the Navier-Stokes

equations to resolve the flow field variables, such as velocity and pressure, unrealistic solutions can result through the use of a single grid and standard discretization schemes. One possible way to counteract these unrealistic solutions is to solve the problem using three grids that are staggered from each other. One grid is used for the velocity in the x direction, one grid is used for the velocity in the y direction, and a third grid is used to store all of the scalar variables (pressure, density, viscosity, species concentrations, etc.). Conveniently, if staggered grids are used to solve the problem, the standard discretization schemes can be used without issue. The problem is that some care must be taken when developing the boundary conditions on the staggered grids.

Because the boundary conditions of the flow field are being defined by the user, it would intuitively make sense that the error at all boundaries would be equal to zero. This was absolutely the case in the advection diffusion examples because all of the all variables were only stored on one grid. However, a brief inspection of the error profiles for flow between parallel plates shows that the error at the south and north boundaries is not zero. This issue arises because of the way the grids are staggered when solving for flow field variables. As a result of the staggered grids, it is not possible to have grid points on all of the boundaries. For example, in the code utilized in this study, the grid on which the u velocity data is stored has grid points on the west and east boundaries, but no grid points on the south and north boundaries. Therefore, when applying boundary conditions for the solution of the u velocity, the velocity can be defined explicitly at the west and east boundaries. This does indeed result in zero error at those boundaries. However, boundary conditions controlling the u-velocity at the south and north boundaries (no slip condition in the case of flow between parallel plates) are enforced through the use of a ghost point and central differencing. Because of this utilization of central differencing, there is some finite error present at the south and north boundaries. Fortunately, the error associated with central differencing is easily quantified using the following equation.

$$\text{boundary error} = \frac{\Delta y^2}{8} \frac{\partial^2 u}{\partial y^2} \quad 6.7$$

In Equation 6.7 (which represents the error in the u velocity at the south and north boundaries), it is shown that the error at the boundaries without grid points is proportional to the product of the square of the grid spacing perpendicular to the boundary and the second derivative of the variable under consideration.

Ultimately, the application of the ETE methodology to the problem of flow between parallel plates was quite successful. The model utilized in this study was able to adequately capture the flow field under consideration. This was verified through the inspection of the true error profiles which indicate that the true error was decreasing proportionally according to the known order of the discretization schemes. The ETE was solved for the entire domain, and profiles were compared to the true error in the fully developed region of the flow. The ETE solution was in nearly perfect agreement with the true error. The ETE solution also decreased proportionally with the grid size at the expected rate based on the order of the discretization scheme.

Figure 6.9: True Error and ETE Solution on 40x10 Grid

Figure 6.10: True Error and ETE Solution on 80x20 Grid

Figure 6.11: True Error and ETE Solution on 160x40 Grid

Figure 6.12: L2 Normal or the True Error and ETE Solution

Chapter 7

Applications to Lid Driven Cavity Flow

In the previous section, it was demonstrated that the ETE methodology is capable of producing accurate discretization error predictions when applied to the system of coupled, non-linear, PDEs that govern fluid flow. The previously demonstrated case of developing flow between parallel plates represents one of the simplest solutions to the Navier-Stokes equations. At this point, the ETE methodology should be applied to more complicated flows of more practical interest. The problem, when considering these more complicated flows, is that there are no closed form solutions as there was for the case of fully developed flow between parallel plates. In order to facilitate the calculation of the true error, which is required to adequately assess the performance of the ETE methodology, it is acceptable to substitute a well-accepted benchmark solution on a fine grid for the exact solution. One such standard benchmark flow that is often used in verification and validation is that of the square lid driven cavity.

7.1 Problem Description and the Benchmark Solution

The lid driven cavity problem is an often cited benchmark solution that is frequently used in the verification and validation of CFD models. The lid driven cavity is often represented by a square computational domain. Unlike the flow between parallel plates, this problem does not offer any simplifications to the Navier-Stokes equation, and requires a numerical simulation to solve the flow field.

The lid driven cavity is typically represented by a square computational domain with four walls of equal length. All of the walls are treated with the no slip boundary conditions. The west, south, and east walls are assumed to be stationary walls with both the u and v velocities set equal to zero at all locations. The north wall is assumed to be moving at a constant velocity in the positive x coordinate direction. Thus, the u velocity at the north wall is set equal to some constant value, U_0 , and the v velocity at the north wall is set

equal to zero. These boundary conditions are represented in equation form below for a square cavity with walls all of length 1.0.

$$\left. \begin{aligned} U(0,Y) &= 0 \\ V(0,Y) &= 0 \end{aligned} \right\} \textit{West Boundary}$$

$$\left. \begin{aligned} U(X,0) &= 0 \\ V(X,0) &= 0 \end{aligned} \right\} \textit{South Boundary}$$

7.1

$$\left. \begin{aligned} U(1,Y) &= 0 \\ V(1,Y) &= 0 \end{aligned} \right\} \textit{East Boundary}$$

$$\left. \begin{aligned} U(X,1) &= U_0 \\ V(X,1) &= 0 \end{aligned} \right\} \textit{North Boundary}$$

While there is no closed form analytic solution for this problem there are several well documented, high accuracy benchmark solutions on fine grids that can be used to approximate the exact solution. The most frequently cited benchmark solution is that of Ghia et al. [23]. In this paper Ghia, publishes the u velocity and v velocity profiles, but only at the center of the square cavity. While this is obviously not sufficient to calculate the true error over the entire domain, it will serve to validate the numerical solution.

The square lid driven cavity data presented by Ghia was solved using the vorticity and stream functions equations. The computational domain, which Ghia defines by $0 < x < 1$, and $0 < y < 1$, was discretized into a uniform mesh consisting of 129 grid points in both the x and y coordinate directions. The governing partial differential equations were discretized using 2nd order accurate central finite-differencing approximations for all second order derivatives, and 1st order accurate upwind differencing schemes for all of the convective terms. The discretized equations were solved using the multigrid method that was the focus of Ghia's paper.

Solutions were generated for Reynolds numbers ranging from 100 to 10000. Here, the Reynolds number is defined as $Re = \rho U_0 L / \mu$, where ρ is the fluid density, μ is the fluid viscosity, L is the wall length, and U_0 is the velocity of the north wall. As the primary concern of this study has been limited to the performance of the ETE methodology for

laminar cases, the $Re=100$ solution will be the primary focus. Ghia provided both u and v velocity data for all Reynolds numbers at the vertical and horizontal geometric center lines, respectively. Profiles of the u and v velocities at the geometric center lines for $Re=100$ are provided in Figure 7.1 and Figure 7.2. All center line u velocity data provided by Ghia is shown in Table 7.1 while the v velocity data is shown in Table 7.2.

Figure 7.1: U Velocity Profile (From Ghia et al.) at the Geometric Centerline for $Re=100$

Figure 7.2: V Velocity Profile (From Ghia et al.) at the Geometric Centerline for $Re=100$

Table 7.1: Benchmark Data for the U Velocity at the Geometric Center Line

129- grid pt. no.	y	Re						
		100	400	1000	3200	5000	7500	10,000
129	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
126	0.9766	0.94123	0.75837	0.63998	0.33236	0.48223	0.47244	0.47221
125	0.9688	0.78871	0.68439	0.57492	0.48296	0.46120	0.47048	0.47783
124	0.9609	0.73722	0.61756	0.51117	0.46547	0.45992	0.47323	0.48070
123	0.9531	0.68717	0.55892	0.46604	0.46101	0.46036	0.47167	0.47804
120	0.9318	0.23131	0.29083	0.33394	0.24882	0.33336	0.34238	0.34633
83	0.7544	0.00832	0.18236	0.38719	0.39791	0.30087	0.20591	0.20673
80	0.6173	-0.13643	0.02125	0.08792	0.07136	0.08183	0.08342	0.08344
65	0.3000	-0.20361	-0.11477	-0.08089	-0.04372	-0.00099	-0.03800	0.03111
39	0.4381	-0.21000	-0.17119	-0.10648	-0.06436	-0.07404	-0.07308	-0.07340
37	0.3813	-0.13662	-0.32726	-0.37806	-0.34427	-0.32855	-0.33176	-0.33186
23	0.1719	-0.10130	-0.34399	-0.38289	-0.34929	-0.33050	-0.32999	-0.32709
14	0.1016	-0.06434	-0.14612	-0.29730	-0.41933	-0.40435	-0.38324	-0.38000
10	0.0703	-0.04775	-0.10338	-0.22220	-0.37827	-0.43643	-0.43025	-0.41657
9	0.0625	-0.04192	-0.09266	-0.20196	-0.35344	-0.42901	-0.43590	-0.42537
8	0.0547	-0.03717	-0.08186	-0.18109	-0.32407	-0.41165	-0.43154	-0.42735
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 7.2: Benchmark Data for the V Velocity at the Geometric Center Line

129- grid pt. no.	x	Re						
		100	400	1000	3200	5000	7500	10,000
129	1.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
123	0.9688	-0.05906	-0.12146	-0.21388	-0.39017	-0.49774	-0.53858	-0.54302
124	0.9609	-0.07391	-0.15663	-0.27669	-0.47425	-0.55069	-0.55216	-0.52987
123	0.9531	-0.08864	-0.19254	-0.33714	-0.52357	-0.55408	-0.52347	-0.49099
122	0.9453	-0.10313	-0.22847	-0.39188	-0.54053	-0.52876	-0.48590	-0.45863
117	0.9063	-0.16914	-0.23827	-0.51550	-0.44307	-0.41442	-0.41050	-0.41496
111	0.8594	-0.22445	-0.44993	-0.42665	-0.37401	-0.36214	-0.36213	-0.36737
104	0.8047	-0.24533	-0.38598	-0.31966	-0.31184	-0.30018	-0.30448	-0.30719
65	0.5000	0.03434	0.03186	0.02326	0.00999	0.00945	0.00824	0.00831
31	0.2344	0.17527	0.30174	0.32235	0.28188	0.27280	0.27348	0.27224
30	0.2266	0.17907	0.30203	0.33075	0.29030	0.28066	0.28117	0.28003
21	0.1563	0.16077	0.28124	0.37095	0.37119	0.33368	0.33060	0.35070
13	0.0938	0.12317	0.22965	0.32627	0.42768	0.42951	0.41824	0.41487
11	0.0781	0.10890	0.20920	0.30353	0.41906	0.42648	0.43564	0.43124
10	0.0703	0.10091	0.19713	0.29012	0.40917	0.43329	0.44030	0.43733
9	0.0625	0.09233	0.18360	0.27485	0.39560	0.42447	0.43979	0.43983
1	0.0000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

7.2 Numerical Solution

7.2.1 Description of Domain, Solver, and Boundary Conditions

To remain consistent with the data presented by Ghia, a computational domain including $0 < x < 1$ and $0 < y < 1$ is utilized in this study. Solutions were completed on three different uniform grids with decreasing grid spacing. The 21×21 grid results in $\Delta x = \Delta y = 0.05$, the 61×61 grid results in $\Delta x = \Delta y = 0.0167$, and the 101×101 grid results in the finest grid spacing where $\Delta x = \Delta y = 0.01$. Typically, grid refinement is completed by factors of 2, thus it would have been more traditional to use grids spacing to include 21×21 , 41×41 , and 81×81 . However, because of the use of the staggered grid in this solution, it is convenient to use odd multiples of the coarsest grid. This allows for comparison of data from grid to grid without interpolation. This will end up being quite important for reasons discussed in the next section.

The square lid driven cavity problem was solved using an in-house developed code written in MATLAB. The solver utilizes the finite volume method with 2^{nd} order central differencing used to discretize the diffusion terms, and a hybrid 2^{nd} order scheme on the convection terms which applies upwind or central differencing based on the cell Peclet number. The pressure velocity coupling was handled using the SIMPLE algorithm.

The boundary conditions for the square lid driven cavity are simply implemented at the walls using the no slip condition. Both the u and v velocity at the west, south, and east walls was set to zero. The v velocity was also set equal to zero at the north boundary while the u velocity was set equal to 1.0 (again to remain consistent the the data presented by Ghia). See equation 7.1 for a straight forward representation of the boundary conditions. As was mentioned previously, the primary interest of this study is the application of the ETE methodology in the laminar flow regime. As a result, the Reynolds number will be set such that $Re = \rho U_0 L / \mu = 100$. Given that the wall length was set to $L = 1.0$, and the lid velocity was set to 1.0, the remaining variables are set to force the desired condition. The density will be set such that $\rho = 1.0$, while the viscosity will be set such that $\mu = 0.01$.

The final discretized equations were solved using the TDMA line by line iteration method. The solver alternates sweeping direction from iteration to iteration to allow the boundary conditions to propagate through the flow field as efficiently as possible. No limit was placed on the number of iterations in the SIMPLE algorithm, thus the solver was allowed to iterate until fully converged. This ensures that the residuals do not need to be taken into account when solving the error transport equation.

7.2.2 Solution Verification

After solving the square lid driven cavity problem using the numerical procedures outlined in the previous section, it is prudent to validate the solutions on all grids by making some comparisons to the established benchmark solution. Because Ghia only provides u and v velocity at the horizontal and vertical geometric centers, it is not possible to make a qualitative comparison of the entire flow field. However, it is still important to inspect the contours of the u and v velocity for all grids. The primary concern is to verify that the flow is behaving as anticipated. Contour plots of the u and v velocity for each of the three grid densities considered in this study are provided in Figure 7.3 through Figure 7.5. A brief qualitative comparison of the contour plots indicates that the solutions on all of the grids are in very good agreement. This would seem to indicate that even the solution on the 21×21 mesh is of high quality. This is likely due to the use of the 2nd order discretization schemes. From a verification perspective, the flow field on all meshes seems to be as anticipated. The large primary vortex is very evident in all of the contour plots. More difficult to see are the extremely small counter rotating vortices in the south east and south west corners of the computational domain. These counter rotating vortices grow as the Reynolds number is increased, but are extremely small for the low $Re=100$ case considered here.

U Velocity

V Velocity

Figure 7.3: Contour Plots of the U (left) and V (right) Velocities on the 21x21 Grid

Figure 7.4: Contour Plots of the U (left) and V (right) Velocities on the 61x61 Grid

U Velocity

V Velocity

Figure 7.5: Contour Plots of the U (left) and V (right) Velocities on the 101x101 Grid

It is also beneficial to examine how the u and v velocities at the centerlines of the computational domain compare to the data present by Ghia. Profiles of the u velocity across the vertical centerline are provided in Figure 7.6, while profiles of the v velocity across the horizontal centerline are provided in Figure 7.7. Several interesting observations can be made from these velocity profiles. The first, and most important, observation is that all of the solutions calculated in this study are in agreement with the solutions generated by Ghia. This, in itself, suggests that the solver is performing as anticipated. A second interesting observation is that there appears to be little improvement in the solution after the grid is refined to the 61x61 case. This would indicate that the 61x61 mesh produces sufficient accuracy for most applications of this data.

Ultimately, the most important verification that can be completed is an inspection of the true error produced by the current model. Because a closed form analytic solution is not available for the square lid driven cavity, the benchmark solution by Ghia could be substituted for the calculation of the true error. Unfortunately, two problems arise with this method. Firstly, the centerline velocities reported by Ghia only include a select few data points that adequately capture the general shape of the profile, as well as all local maxima and minima. Secondly, even if this data was available, the data points would not match up as Ghia utilized a collocated grid while this study makes use of a staggered grid. Because of both of these issues, the use of the velocity data reported by Ghia to calculate the true error would require interpolation. This gives rise to the greatest issue. If the data reported by Ghia is interpolated to the computational grids used in this study, and the error is calculated, it is seen that the interpolation error (even when using a 3rd order spline) is on the same order, and in some cases larger, than the discretization error. This issue is quite apparent in the u and v error profiles provided in Figure 7.8.

Because of the issues associated with the interpolation of the data presented by Ghia, it was decided to utilize an ultra-fine mesh solution from the same solver used in this study to compute the solutions on the 21x21, 61x61, and 101x101. In order for the grid points to line up exactly, and require no interpolation, a 301x301 grid is required. The solution on the 301x301 grid is absolutely acceptable given that the solution is within 0.2% of the lid velocity.

Figure 7.6: U Velocity at the Vertical Geometric Centerline of the Square Lid Driven Cavity

Figure 7.7: V Velocity at the Horizontal Geometric Centerline of the Square Lid Driven Cavity

U Velocity

V Velocity

Figure 7.8: Profiles of the U (left) and V (right) True Error Calculated Using Interpolated Ghia Data

7.2.3 Solution of the Error Transport Equation

The ETE derived from first principles here is identical to what was derived for use in the case of flow between two parallel plates. The equations are repeated below for reference purposes only.

Continuity Error Transport Equation

$$\frac{\partial \varepsilon_u}{\partial x} + \frac{\partial \varepsilon_v}{\partial y} = -\tau_m \quad 7.2$$

U Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + 2 \frac{\partial(u^n \varepsilon_u)}{\partial x} + \frac{\partial(v^n \varepsilon_u)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] + \rho \frac{\partial(u^n \varepsilon_v)}{\partial y} = -\tau_u \quad 7.3$$

V Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial(u^n \varepsilon_v)}{\partial x} + 2 \frac{\partial(v^n \varepsilon_v)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial y} - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] + \rho \frac{\partial(v^n \varepsilon_u)}{\partial x} = -\tau_v \quad 7.4$$

Where τ_m , τ_u , and τ_v are calculated by mapping the numerical solution to the continuous domain using the WSM and evaluating the original conservation equations.

Contour plots of the continuity error transport source term have been provided for all grids in Figure 7.9. Contour plots of the u and v velocity error transport source terms have been provided for all grids in Figure 7.10 and Figure 7.11, respectively. Again, it is difficult to select an appropriate scale for the contour plots as all the minimum and maximum magnitudes of the error source terms differ by several orders of magnitude. The primary observation that should be made from the provided contour plots is the location of the maximum error source. No matter which source term is being considered, the largest value of the source term is found at the north west and north east corners of the computational domain. This intuitively makes sense as this location of discontinuity is a source or error in this type of problem.

Figure 7.9: Continuity Error Source Term for 21x21, 61x61, and 101x101 Grids

Figure 7.10: U Velocity Error Source Term for 21x21, 61x61, and 101x101 Grids

Figure 7.11: V Velocity Error Source Term for 21x21, 61x61, and 101x101 Grids

Here the boundary conditions were applied in a manner similar to the case of flow between parallel plates. Because both the u and v velocities are explicitly defined at all boundaries using dirichlet boundary conditions, the error at all boundaries should be zero. However, because of the use of a staggered grid this is not the case. For the x momentum equation, the u velocity can be defined explicitly at the west and east boundaries (resulting in a zero error boundary condition for the ETE), but the u velocity at the south and north boundaries must be defined using linear interpolation. This ultimately results in the same error at the south and north boundaries that was discussed in the case of flow between parallel plates. This situation is similar, but reversed in the case of the y momentum equation. Here the v velocity is defined explicitly at the south and north boundaries (again resulting in a zero error boundary condition for the ETE), while the v velocity at the west and east boundaries is defined using linear interpolation.

Using the calculated error source terms and the described boundary conditions, the ETE is solved on the three meshes considered in this study. The same discretization scheme that was applied to the governing PDEs was also applied to the ETE. The solution was allowed to fully converge so that no iterative convergence error needed to be taken into account. The true error was approximated through the use of the solution on the 301×301 mesh. Contour plots of the predicted discretization error for the u and v velocity on all grids are provided in Figure 7.12 through Figure 7.17. Each figure contains a contour plot of the predicted error, as well as a contour plot of the corresponding true error.

An inspection of the error contour plots serves as a good qualitative assessment of the error predicted by the ETE solution. A comparison of the predicted error to the true error on all of the considered grids shows quite reasonable agreement between the shape of the error, as well as the magnitude. The locations of regions of large error are very much accurately captured by the ETE solution. The scale of the contour plots was kept constant from grid to grid. This results in a loss of resolution when showing the smaller error on the finer grids. However, this serves to show how the error reduces as grid spacing is reduced. This is one of the most important discriminators of an error predictions method which seems to be very well captured in this study.

Figure 7.12: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 21x21 Grid

True Error

ETE Solution

Figure 7.13: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 61x61 Grid

True Error

ETE Solution

Figure 7.14: Contour Plots of the U Velocity True Error (left) and ETE Solution (right) on the 101x101 Grid

True Error

ETE Solution

Figure 7.15: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 21x21 Grid

True Error

ETE Solution

Figure 7.16: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 61x61 Grid

True Error

ETE Solution

Figure 7.17: Contour Plots of the V Velocity True Error (left) and ETE Solution (right) on the 101x101 Grid

In order to further illustrate that the predicted error is reducing as it should with grid spacing, the L2 normal of the u and v velocity ETE solution is calculated on each grid. The resulting values are plotted as a function of the representative grid spacing and plotted. The same analysis was completed on the true error. These results were also included in the plots for comparison purposes. The L2 normal of the u velocity error is plotted in Figure 7.18, while the L2 normal of the v velocity error is plotted in Figure 7.19. An inspection of these plots shows an exceedingly strong agreement between the L2 normal of the ETE solution and the L2 normal of the true error. The u velocity error shows almost perfect agreement between the ETE solution and the true error. The v velocity error does not have quite as strong of an agreement between the ETE solution and the true error, but it is still quite acceptable.

Additionally, it should be noted that the L2 normal of the u velocity true error is shown to be proportional to the grid spacing to the power of 2.26 while the L2 normal of the v velocity true error is shown to be proportional to the grid spacing to the power of 2.35. This is an important observation as it shows that the error is decreasing as anticipated, based on the 2nd order discretization schemes utilized in the numerical model. This is an important step in the validation of the numerical solutions. Most importantly, the same dependence on grid spacing is shown the ETE solutions. The L2 normal of the u velocity ETE solution is shown to be proportional to the grid spacing to the power of 2.28, while the L2 normal of the v velocity ETE solution is shown to be proportional to the grid spacing to the power of 2.33. This represents an extremely strong agreement with the true error. Ultimately, this indicates that the ETE methodology presented here is performing as desired. The overall error in the domain is in very good agreement with the true error. Most importantly, the ETE solution seems to be accurately representing the variation of error with grid spacing.

Figure 7.18: Variation in the L2 Normal of the U Velocity Error as a Function of Grid Spacing

Figure 7.19: Variation in the L2 Normal of the V Velocity Error as a Function of Grid Spacing

A final evaluation of the ETE methodology is completed by comparing profiles of the ETE solution with the true error at several different locations in the computational domain. Additionally, the error predicted using the Richardson Extrapolation method is also included for comparison purposes. Profiles of the u velocity error on all of the grids are plotted at three different x coordinate locations ($x=0.25$, $x=0.50$, and $x=0.75$) are plotted in Figure 7.20 through Figure 7.22. Profiles of the v velocity error on all of the grids plotted at three different y coordinate locations ($y=0.25$, $y=0.50$, and $y=0.75$) are plotted in Figure 7.23 through Figure 7.25.

Many interesting observations can be made from the plotted error profiles. Overall, the ETE solution provides quality estimates of the true error for both the u and v velocities. The shape of the ETE solutions matches the shape of the true error very well in most locations. The locations of the minimum and maximum error match quite well. The only exception is the ETE solution on the 21×21 grid where it appears that the locations of the minimum and maximum error are shifted slightly. In the majority of the domain, it seems that the error is slightly over predicted by the ETE solution.

The error predictions generated using the Richardson Extrapolation method also performed exceedingly well. The Richardson Extrapolation results are in better agreement with the true error than the ETE solution in the vast majority of the profiles. This is not entirely unexpected as the Richardson Extrapolation methods make use of solutions on three grids to perform error predictions. Ultimately, the goal of this study is to provide a reliable error prediction with only one grid, thus saving a huge amount of time. The generation of solutions on three grids is not difficult in this simple case, but in some problems of practical interest in the industry, computational grids can contain hundreds of thousands of data points. Generating additional, unnecessary solutions on such a grid would prove to be prohibitively time consuming.

Overall, the ETE methodology presented here performed exceedingly well in the case of square lid driven cavity flow. The predicted error throughout the domain appears to be in good qualitative and quantitative agreement with the true error. The ETE solution for both the u and v velocity error also appears to decay at the anticipated rate. While the Richardson Extrapolation method does appear to provide a better estimate of the error in many cases, the time savings associated with the ETE methodology should make it an attractive option.

Figure 7.20: U Velocity Error Profile at X=0.25 for 21x21, 61x61, and 101x101 Grids

Figure 7.21: U Velocity Error Profile at X=0.5 for 21x21, 61x61, and 101x101 Grids

Figure 7.22: U Velocity Error Profile at X=0.75 for 21x21, 61x61, and 101x101 Grids

Figure 7.23: V Velocity Error Profile at Y=0.25 for 21x21, 61x61, and 101x101 Grids

Figure 7.24: V Velocity Error Profile at Y=0.5 for 21x21, 61x61, and 101x101 Grids

Figure 7.25: V Velocity Error Profile at Y=0.75 for 21x21, 61x61, and 101x101 Grids

Chapter 8

Applications to Backward Facing Step Flow

In the previous section, it was demonstrated that the ETE methodology performs acceptably, even when applied to more complicated flows such as a square lid driven cavity. Based on the results from the square lid driven cavity, it cannot be said that the ETE methodology is capable of predicting discretization error more accurately than the Richardson Extrapolation method, but the results are quite good and require a solution on only one grid as opposed to three. As a second test of the ETE methodology, flow over a backward facing step will be solved and the error will be predicted. This portion of the study will be completed in exactly the same manner as that of the lid driven cavity flow. Again, there is no closed form analytic solution to this problem. However, there are well accepted benchmark solutions on fine meshes. Ultimately, as with the lid driven cavity problem, the true error in this case will be approximated using a solution on a very fine mesh from the same solver that has been used throughout this study. This will eliminate the need for interpolation between the grids and provide a better analysis of the predicted error.

8.1 Problem Description and the Benchmark Solution

The backward facing step problem is another often cited benchmark solution that is frequently used in the verification and validation of CFD models. The backward facing step problem is characterized by a small channel the empties into a larger channel. The sudden expansion results in a recirculation zone just after the expansion step. This recirculation zone is referred to as the primary recirculation zone. Backward facing step flow has been the topic of numerous experimental and numerical studies for quite some time [24–27]. This flow is better described schematically as shown in Figure 8.1.

Figure 8.1: Schematic of Backward Facing Step Flow

A comprehensive study of backward facing step flow was presented by Armaly et al. [27]. This study included a summary of the extensive experimental measurements taken by the group, as well as the results of a numerical analysis that was also completed. The experimental setup featured an inlet channel with a height of 0.52 cm and a length of 20 cm to ensure fully developed flow at the backward facing step. The channel downstream of the step had a height of 1.01 cm, resulting in an expansion ratio of approximately 2. The flow velocity measurements were taken using a laser-Doppler anemometer.

The primary recirculation zone was of the greatest interest in the study presented by Armaly and serves as an important basis for CFD code verification and validation. The primary recirculation zone is caused by the sudden expansion of the flow, and is always attached to the downstream side of the backward facing step. The length of the primary recirculation zone was shown to be a function of the flow Reynolds number and the height of the step. Here the Reynolds number is defined as $Re = \rho U D_h / \mu$, where ρ is the fluid density, U is the average inlet velocity, D_h is the hydraulic diameter (2x the inlet channel height), and μ is the fluid viscosity. As Re is increased up to 1200, the reattachment length increases with Re . At $Re=1200$, the flow begins to transition from laminar to turbulent flow, and the length of the primary recirculation zone actually begins to shrink as Re is further increased. This trend continues to approximately $Re=6600$, at which point the flow is turbulent and the length of the primary recirculation zone remains

constant. The variation of the primary recirculation zone is demonstrated in the plot provided in Figure 8.2, which is taken directly from Armaly et al.

Figure 8.2: Variation of the Primary Recirculation Zone Length With Reynolds Number

It was also reported by Armaly et al. that the backward facing step flow is only truly two dimensional when $Re < 400$ and when $Re > 6000$. Inside this range of Reynolds numbers, the flow was shown to be strongly three dimensional. This three dimensionality was apparent in the experimental data, but was also confirmed by the numerical analysis as the length of the primary recirculation zone could not be accurately predicted inside this range. Profiles of the u velocity obtained from experiment and numerical analysis at three values of Reynolds number are provided in Figure 8.3. This plot is taken directly from the work by Armaly et al. and is an excellent illustration of the three dimensionality of the flow when $Re > 400$. As the interest of this study remains the performance of the ETE methodology in laminar flow, the numerical analysis performed in this study will focus on $Re < 400$.

Figure 8.3: Experimental and Numerical Profiles of U Velocity Obtained by Armaly et al.

The numerical solutions presented by Armaly et al. were obtained on a computation domain that was very similar in dimension to their experimental setup. The inlet channel had a height of 0.5, while the larger downstream channel had a height of 1.0, resulting in an expansion ratio of 2.0. The length of the domain was set such that the outflow boundary was at least four primary recirculation zone lengths away from the inlet. This channel length was shown to be sufficient to ensure fully developed flow. The velocity at the inlet channel (west boundary) was defined using the velocity measured in the corresponding experiments. Both the u and v velocity was set equal to zero at the south and north wall boundaries, as well as the downstream side of the backward facing step. The west boundary (outlet) was set using the outflow boundary condition which assumes that the flow at the boundary is fully developed. These boundary conditions are represented more clearly in equation below.

$$U(0,y) = \text{measured}$$

$$V(0,y) = 0$$

$$U(x,0) = V(x,0) = 0 \tag{8.1}$$

$$U(x,1) = V(x,1) = 0$$

$$\frac{\partial U}{\partial x}(L,y) = \frac{\partial V}{\partial x}(L,y) = 0$$

The computational domain was discretized using 45 grid points in both the x and y coordinate direction. Armaly et al. found this to be sufficient to determine the reattachment length. The Navier-Stokes equations are solved using a finite difference solver of unspecified order. The resulting variation of recirculation zone length is provided in Figure 8.4 which is a plot taken straight from Armaly's presentation of the data. From this plot, it is quite apparent that beginning at $Re=400$ the two dimensional numerical model is no longer capable of accurately predicting the recirculation zone length.

Figure 8.4: Measured and Numerically Calculated Reattachment Length Obtained by Armaly et al.

8.2 Numerical Solution

8.2.1 Description of Domain, Solver, and Boundary Conditions

Because all of the data presented in Armaly et al. is normalized by the step height, it was not necessary to generate a computational domain that was identical to that used in the numerical analysis conducted by Armaly. The backward facing step flow in this study was solved on a computational domain that includes $0 < x < 2.5$ and $0 < y < 0.2$. This domain length far exceeds the requirement for fully developed flow established by Armaly et al. (4x the recirculation zone length).

As with the lid driven cavity problem, there is obviously no closed form analytic solution that can be used to calculate the true error. Use of experimental or numerical data obtain by Armaly et al. is not a viable substitution for the exact solution as interpolation would be required, which has been shown to significantly alter the estimate of true error. The solution here is the same as it was for the lid driven cavity problem. The exact solution will be approximated by a solution on an extremely fine grid. This method worked exceedingly well on the lid driven cavity case, and should prove just as successful with the backward facing step. Solutions were completed on three different uniform grids with decreasing grid spacing. The 21x11 grid results in $\Delta x=0.1250$ and $\Delta y=0.0200$, the 61x31 grid results in $\Delta x=0.0417$ and $\Delta y=0.0067$, and the 101x51 grid results in the finest grid spacing where $\Delta x=0.0250$ and $\Delta y=0.0040$. The finest grid solution that is used to approximate the exact solution was solved on a 301x151 grid resulting in $\Delta x=0.0083$ and $\Delta y=0.0013$. Again, because of the use of the staggered grid in this solution, it is convenient to use odd multiples of the coarsest grid for the grid refinement. This allows for comparison of data from grid to grid without interpolation.

The backward facing step problem was solved using an in-house developed code written in MATLAB. The solver utilizes the finite volume method with 2nd order central differencing used to discretize the diffusion terms, and a hybrid 2nd order scheme on the convection terms which applies upwind or central differencing based on the cell Peclet number. The pressure-velocity coupling was handled using the SIMPLE algorithm.

The boundary conditions for the backward facing step are applied exactly in the same manner as in the numerical analysis presented by Armaly et al. (equation x). The only exception is that the inlet velocity profile was calculated using the analytic solution for flow between parallel plates rather than measure velocity profiles. Given that the experimental setup presented by Armaly et al. was developed to ensure fully developed flow at the channel inlet, this assumption is reasonable. As was mentioned previously, the primary interest of this study is the application of the ETE methodology in the laminar flow regime. As a result, the Reynolds number is set such that $Re=100$ for the error analysis portion of this section. Solutions are also presented for several Reynolds numbers below 400 in the verification and validation section. To obtain the various

values of Reynolds number, the value of the kinematic viscosity was maintained at a constant value of $1\text{E-}6$, while the velocity was varied.

The final discretized equations were solved using the TDMA line by line iteration method. The solver alternates sweeping direction from iteration to iteration to allow the boundary conditions to propagate through the flow field as efficiently as possible. No limit was placed on the number of iterations in the SIMPLE algorithm, thus the solver was allowed to iterate until fully converged. This ensures that the residuals do not need to be taken into account when solving the error transport equation.

8.2.2 Solution Verification

As was discussed in the previous section, the backward facing step problem was solved for $Re=100$ on three different grids of decreasing grid spacing. Additionally, for the purposes of verification and validation of the solution, the backward facing step problem was also solved at $Re=50, 100, 150, 200, 250, 300, 350$, and 400 on the 60×30 grid. No solutions were obtained for $Re > 400$ as it was shown by Armaly et al. that the three dimensional effects of the flow become important, and the two dimensional solver does not adequately represent reality. Initially, for the backward facing step, it is important to inspect contour plots of the u velocity to ensure that the flow is behaving qualitatively as anticipated. U velocity contour plots for all Reynolds numbers considered in this study are provided in Figure 8.5 and Figure 8.6. Several important observations can be made from an inspection of these contour plots. The first is that the primary circulation zone is very evident, even in the case of $Re=50$. As anticipated, the primary recirculation zone increases in length as Reynolds number is increased. Additionally, the flow does appear to be fully developed, especially at the lower values of Reynolds number, which is important for the purposes of the error analysis. For completeness, contour plots of the v velocity at all values of Reynolds number are provided in Figure 8.7 and Figure 8.8.

Another beneficial qualitative assessment is to inspect the shape of the u velocity profiles at various x coordinate locations. These profiles are provided for all values of Reynolds numbers in Figure 8.9 and Figure 8.10. It is important to note that these plots simply provide a view of the shape of the profile. The apparent value of the u velocity in these profiles has absolutely no meaning. What is apparent, again, is the increasing length of the recirculation zone with increasing Reynolds number. Also, inspection of the

u velocity profiles at high values of Reynolds number shows the beginning of the development of the secondary recirculation zone at the top of the channel.

Figure 8.5: U Velocity Contours at Various Values of Reynolds Number

Figure 8.6: U Velocity Contours at Various Values of Reynolds Number

Figure 8.7: V Velocity Contours at Various Values of Reynolds Number

Figure 8.8: V Velocity Contours at Various Values of Reynolds Number

Figure 8.9: U Velocity Profiles at Various X Coordinate Locations

Figure 8.10: U Velocity Profiles at Various X Coordinate Locations

For the verification and validation of a backward facing step solution to be complete, it is important to plot the variation of the recirculation zone length as a function of the Reynolds number. This profile will be compared to the experimental data obtained by Armaly et al. The length of the recirculation zone is defined as the distance from the inlet channel to the location where the u velocity next to the bottom channel wall is no longer negative. This profile is plotted in Figure 8.11 below. As can be seen in the profile, the numerically calculated recirculation zone length is in very good agreement with the experimental data obtained by Armaly et al. This indicates that the solver is performing as anticipated and providing reasonable solutions on the 61x31 grid.

Figure 8.11: Recirculation Zone Length as a Function of Reynolds Number

In addition to inspecting the solution on a single grid at various values of Reynolds number, it is also important to consider the solution on all of the grids at a single Reynolds number. This analysis is carried out for $Re=100$. This solution provides a reasonably large recirculation zone, while remaining well inside the laminar regime.

Profiles of the u velocity have been provided at various x coordinate locations on the same plot in Figure 8.12. Again, the magnitude of the u velocity in the plots is not as important as the qualitative assessment of how the velocity profiles compare. Several interesting observations can be made here. The first is that while the solution on the 21×11 grid is quite different, the solutions on the 61×31 and 101×51 grids nearly lie on top of one another. This indicates that the numerical solution is not improving significantly after the grid spacing is further reduced from the 61×31 grid. This is why the 61×31 grid was utilized in the calculations completed at varying values of Reynolds number. Another valuable observation is that all of the grids produce extremely similar solutions near the outlet of the computational domain. This is largely because of the simple flow structure at the outlet. Here the velocity corresponds to fully developed flow between parallel plates. Given the relatively small grid spacing on even the coarsest grid, all solutions produce minimal error at the outlet.

Ultimately, it appears that the solver used in this study is perfectly capable of providing quality solutions to the backward facing step problem. The flow appears to be behaving as anticipated. The variation of the recirculation zone length with Reynolds number is in excellent agreement with the experimental data. An analysis of the velocity profiles at various grid resolutions indicates that the solution on the 301×151 grid should be more than sufficient to approximate the exact solution. These results should provide a reliable analysis of the ETE methodology.

Figure 8.12: U Velocity Profiles at Various X Coordinate Locations on All Grids

8.2.3 Solution of the Error Transport Equation

The ETE derived from first principles here is identical to what was derived for use in the cases of flow between two parallel plates and lid driven cavity flow. The equations are repeated below for reference purposes only.

Continuity Error Transport Equation

$$\frac{\partial \varepsilon_u}{\partial x} + \frac{\partial \varepsilon_v}{\partial y} = -\tau_m \quad 8.2$$

U Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + 2 \frac{\partial(u^n \varepsilon_u)}{\partial x} + \frac{\partial(v^n \varepsilon_u)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] + \rho \frac{\partial(u^n \varepsilon_v)}{\partial y} = -\tau_u \quad 8.3$$

V Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial(u^n \varepsilon_v)}{\partial x} + 2 \frac{\partial(v^n \varepsilon_v)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial y} - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] + \rho \frac{\partial(v^n \varepsilon_u)}{\partial x} = -\tau_v \quad 8.4$$

Where τ_m , τ_u , and τ_v are calculated by mapping the numerical solution to the continuous domain using the WSM and evaluating the original conservation equations.

Contour plots of the continuity error transport source term have been provided for all grids in Figure 8.13. Contour plots of the u and v velocity error transport source terms have been provided for all grids in Figure 8.14 and Figure 8.15, respectively. Again it is difficult to select an appropriate scale for the contour plots as all the minimum and maximum magnitudes of the error source terms differ by several orders of magnitude. The primary observation that should be made from the provided contour plots is the location of the maximum error source. No matter which source term is being considered, the largest value of the source term is found at the corner of the expansion step. This is likely a significant source of error due to the difficulties associated with applying the

boundary conditions at the step. Because no grid point falls exactly on the desired corner, there is generally some small fluid velocity below the step which is inaccurate.

Figure 8.13: Continuity Error Source Terms for 21x11, 61x31, and 101x51 Grids

Figure 8.14: U Velocity Error Source Terms for 21x11, 61x31, and 101x51 Grids

Figure 8.15: V Velocity Error Source Terms for 21x11, 61x31, and 101x51 Grids

Here the boundary conditions were applied in a manner similar to the case of flow between parallel plates. Because both the u and v velocities are explicitly defined at the west, south, and north boundaries using dirichlet boundary conditions, the error at all boundaries should be zero. However, because of the use of a staggered grid, this is not the case. For the x momentum equation, the u velocity can be defined explicitly at the west boundary (resulting in a zero error boundary condition for the ETE), but the u velocity at the south and north boundaries must be defined using linear interpolation. This ultimately results in the same error at the south and north boundaries that was discussed in the case of flow between parallel plates. This situation is similar, but reversed in the case of the y momentum equation. Here the v velocity is defined explicitly at the south and north boundaries (again resulting in a zero error boundary condition for the ETE), while the v velocity at the west and east boundaries is defined using linear interpolation. The east boundary condition for the u momentum equation is defined assuming fully developed flow such that $\partial U/\partial x=0$. This results in a similar boundary condition for the error where $\partial \epsilon_u/\partial x=0$.

Using the calculated error source terms and the described boundary conditions, the ETE is solved on the three meshes considered in this study. The same discretization scheme that was applied to the governing PDEs was also applied to the ETE. The solution was allowed to fully converge so that no iterative convergence error needed to be taken into account. The true error was approximated through the use of the solution on the 301x301 mesh. Contour plots of the predicted discretization error for the u and v velocity on all grids are provided in Figure 8.16 through Figure 8.21. Each figure contains a contour plot of the predicted error, as well as a contour plot of the corresponding true error.

An inspection of the error contour plots serves as a good qualitative assessment of the error predicted by the ETE solution. A comparison of the predicted error to the true error on all of the considered grids shows quite reasonable agreement between that shape of the error, as well as the magnitude. The locations of regions of large error are very much accurately captured by the ETE solution. The scale of the contour plots was kept constant from grid to grid. This results in a loss of resolution when showing the smaller error on the finer grids. However, this serves to show how the error reduces as grid spacing is reduced. This is one of the most important discriminators of an error predictions method, which seems to be very well captured in this study.

Figure 8.16: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 21x11 Grid

Figure 8.17: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 61x31 Grid

Figure 8.18: Contour Plots of the U Velocity True Error (top) and ETE Solution (bottom) on the 101x51 Grid

Figure 8.19: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 21x11 Grid

Figure 8.20: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 61x31 Grid

Figure 8.21: Contour Plots of the V Velocity True Error (top) and ETE Solution (bottom) on the 101x51 Grid

In order to further illustrate that the predicted error is reducing as it should with grid spacing, the L2 normal of the u and v velocity ETE solution is calculated on each grid. The resulting values are plotted as a function of the representative grid spacing. The same analysis was completed on the true error. These results were also included in the plots for comparison purposes. The L2 normal of the u velocity error is plotted in Figure 8.22 while the L2 normal of the v velocity error is plotted in Figure 8.23. An inspection of these plots shows reasonable agreement between the L2 normal of the ETE solution and the L2 normal of the true error. The agreement here is not as good as was observed in the lid driven cavity problem. One possible cause of the less impressive agreement is that the error in the backward facing step case is quite small even on the coarsest grid. Here the L2 normal of the u velocity true error on the coarsest grid is 1.95E-6, while the L2 normal of the u velocity true error on the coarsest grid in the lid driven cavity case was 5.79E-4. This represents a two orders of magnitude difference in the error. Recall from the derivation of the x and y momentum error transport, that two higher order terms were eliminated from the equation based on the reasoning that they were insignificant compared to the lower order terms. This is usually a reasonable assumption; however, given how small the error is in the case of backward facing step flow, this assumption may no longer be valid here.

Additionally, it should be noted that the L2 normal of the u velocity true error is shown to be proportional to the grid spacing to the power of 2.09, while the L2 normal of the v velocity true error is shown to be proportional to the grid spacing to the power of 1.99. This is an important observation as it shows that the error is decreasing as anticipated based on the hybrid 1st / 2nd order discretization schemes utilized in the numerical model. This is an important step in the validation of the numerical solutions. Most importantly, a similar dependence on grid spacing is shown the ETE solutions. The L2 normal of the u velocity ETE solution is shown to be proportional to the grid spacing to the power of 1.69, while the L2 normal of the v velocity ETE solution is shown to be proportional to the grid spacing to the power of 1.49. Again, the agreement here is not as strong as in the case of lid driven cavity flow, however, it is quite reasonable. Given the use of the hybrid scheme, one would anticipate the order to be somewhere between 1 and 2. The overall error in the domain is in very good agreement with the true error. Most importantly, the ETE solution seems to be accurately representing the variation of error with grid spacing.

Figure 8.22: Variation in the L2 Normal of the U Velocity Error as a Function of Grid Spacing

Figure 8.23: Variation in the L2 Normal of the V Velocity Error as a Function of Grid Spacing

A final evaluation of the ETE methodology is completed by comparing profiles of the ETE solution with the true error at several different locations in the computational domain. Additionally, the error predicted using the Richardson Extrapolation method is also included for comparison purposes. Profiles of the u velocity error on all of the grids are plotted at four different x coordinate locations ($x=0.5$, $x=1.0$, $x=1.5$, and $x=2.0$) are plotted in Figure 8.24 through Figure 8.27. Profiles of the v velocity error on all of the grids plotted at three different y coordinate locations ($y=0.25$, $y=0.50$, and $y=0.75$) are plotted in Figure 8.28 and Figure 8.29.

Many interesting observations can be made from the plotted error profiles. Overall, the ETE solution provides reasonably quality estimates of the true error for both the u and v velocities. The shape of the ETE solutions matches the shape of the true error quite well in most locations. The locations of the minimum and maximum error match quite well. There is, unfortunately, one significant exception here. At large x coordinate locations, the shape of the ETE solution for the u velocity error seems to begin diverging from the true error. This issue is quite evident at $x=1.5$ and $x=2.0$ in the provided plots. The boundary conditions at the south and north boundaries appear to be in good agreement with the true error, but the shape of the profile diverges rapidly. Several possible causes do come to mind. First, as was mentioned previously, is that as a result of the small magnitude of the error in this case, the higher order terms that were left out of the error transport equation have become significant factors. This is further supported by the fact that the ETE solution seems to be in much better agreement with the true error in regions of high error ($x=0.5$ and $x=1.0$). The magnitude of the ETE solution seems to be in reasonable agreement with the true error even at larger values of x , but the shape seems to be wrong. This could also have something to do with the implementation of the boundary condition at the east face. The ETE solution for the v velocity error seems to be in very good agreement with the true error, even better than the Richardson Extrapolation method in some instances.

The error predictions generated using the Richardson Extrapolation method also performed exceedingly well. The Richardson Extrapolation results are in better agreement with the true error than the ETE solution in the vast majority of the profiles. This is not entirely unexpected as the Richardson Extrapolation methods make use of solutions on three grids to perform error predictions. Ultimately, the goal of this study is to provide a reliable error prediction with only one grid, thus saving a huge amount of

time. The generation of solutions on three grids is not difficult in this simple case, but in some problems of practical interest in the industry, computational grids can contain hundreds of thousands of data points. Generating additional, unnecessary solutions on such a grid would prove to be prohibitively time consuming. Again, it should be stated that the goal of the ETE methodology is not necessarily to outperform Richardson Extrapolation, but rather to provide estimates of the error that are reasonable while requiring substantially less computational effort.

Overall, the ETE methodology presented here performed exceedingly well in the case of backward facing step flow. The predicted error throughout the domain appears to be in good qualitative and quantitative agreement with the true error. The ETE solution for both the u and v velocity error also appears to decay at the anticipated rate. While the Richardson Extrapolation method does appear to provide a better estimate of the error in many cases, the time savings associated with the ETE methodology should make it an attractive option.

Figure 8.24: U Velocity Error Profile at X=0.5 for 21x11, 61x31, and 101x51 Grids

Figure 8.25: U Velocity Error Profile at X=1.0 for 21x11, 61x31, and 101x51 Grids

Figure 8.26: U Velocity Error Profile at X=1.5 for 21x11, 61x31, and 101x51 Grids

Figure 8.27: U Velocity Error Profile at X=2.0 for 21x11, 61x31, and 101x51 Grids

Figure 8.28: V Velocity Error Profile at Y=0.06 for 21x11, 61x31, and 101x51 Grids

Figure 8.29: V Velocity Error Profile at Y=0.14 for 21x11, 61x31, and 101x51 Grids

Chapter 9

Applications to a Suddenly Accelerated Flat Plate

To this point in the study, results have been presented for a wide range of problems including advection diffusion in both one and two dimensions, developing flow between parallel plates, lid driven cavity flow, and backward facing step flow. The common factor in all of these problems is that only the steady state solutions were considered. The derivation for the error transport equations derived from the Navier-Stokes equations includes a time derivative term. The unsteady solution of the error transport equation is implemented in a manner very similar to the solution of the flow field in any unsteady problem. The solution of the error transport equation for an unsteady problem is not significantly more complex than the steady state solutions already considered. The only real added layer of complexity is that error source terms must be generated at each time step. While this adds to the time required to implement the ETE methodology it does not complicate the solution.

To assess the performance of the ETE methodology in the unsteady case it is desirable to have a closed form analytic solution to use in the calculation of the true error. Fortunately, as with the flow between parallel plates, there are several problems with specific conditions that allow for simplifying assumptions to be made. One such problem is known as Stokes first problem which describes the unsteady flow over a suddenly accelerated plate.

9.1 Problem Description and Analytic Solution

The unsteady flow of a fluid over a suddenly accelerated flat plate is one of several closed form analytic solutions to the unsteady Navier-Stokes equations. As with fully developed flow between parallel plates that was considered in a previous chapter, several assumptions about the flow field are made which result in significant

simplifications to the Navier-Stokes equations, allowing for a closed form analytic solution. A schematic representation of the problem under consideration is shown in Figure 9.1.

Figure 9.1: Schematic of the Flow Above a Suddenly Accelerated Flat Plate

The problem under consideration is defined by a fluid at rest above an infinitely long flat plate that is also at rest. At $t=0$ the plate is put into motion. The function representing $U(t)$ can be as complex as desired. However, for this study, consideration will be given only to a constant plate velocity U_0 . Parallel flow is assumed so that there is no flow in the cross stream direction ($v=0$). The infinitely long plate implies that there is also no change in the fluid velocity in the x coordinate direction. The u velocity only varies in the y coordinate direction. Additionally, the flow is completely driven by the motion of the plate, implying no pressure gradient in the x coordinate direction. Application of these assumptions to the full continuity and momentum equations results in the following simplified forms

$$\frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

9.1

In the above equations u represents the velocity in the x coordinate directions and ν is the fluid viscosity. The simplified forms of the Navier-Stokes equations provided above are analogous to the unsteady heat conduction equation in which the bottom plate is suddenly changed to a different temperature. The closed form analytic solution of this equation is provided through the use of the complimentary error function.

$$\frac{u}{U_0} = 1 - \operatorname{erf}\left(\frac{y}{2\sqrt{\nu t}}\right)$$

Where

9.2

$$\operatorname{erf}(\beta) = \frac{2}{\sqrt{\pi}} \int_0^\beta e^{-x^2} dx$$

A plot of the velocity profiles generated by the above analytic solution assuming a kinematic viscosity of $\nu=0.1$ and a constant plate velocity of $U_0=1.0$ are provided in Figure 9.2 for several different moments in time.

Figure 9.2: Profiles from the Analytic Solution for a Suddenly Accelerated Flat Plate at Different Values of Time

9.2 Numerical Solution

9.2.1 Description of Domain, Solver, and Boundary Conditions

A computational domain was developed based on knowledge of the simplifying assumptions and the analytic solutions. The suddenly accelerated plate problem is solved on a computational domain that includes $0 < x < 3$ and $0 < y < 0.3$. The length of the computational domain was set to be significantly larger than the height of the computational domain because of the simplifying assumption of an infinitely long plate. This works to ensure fully developed parallel flow. The height of the computational domain was select such that the north boundary of the flow would be well out of the shear layer created by the moving plate at the south boundary. It can be shown that the shear layer thickness (defined as the point where $u/U_0=0.01$) is approximately defined as $\delta \approx 3.64\sqrt{\nu t}$. It will be shown that the shear layer thickness in this study will be at most 0.063 thus the domain height is sufficiently large to represent the free stream.

Due to the availability of a closed form analytic solution to the suddenly accelerated plate there is no need to calculate a solution on an extremely fine grid to approximate the true error. In this case, a more traditional refinement method is employed where the grid spacing is halved for each successively finer grid. Solutions were completed on three different uniform grids with decreasing grid spacing. The 41x21 grid results in $\Delta x=0.075$ and $\Delta y=0.015$, the 81x41 grid results in $\Delta x=0.0375$ and $\Delta y=0.0075$, and the 161x81 grid results in the finest grid spacing where $\Delta x=0.01875$ and $\Delta y=0.00375$.

For this first attempt at applying the ETE methodology to an unsteady solution it was decided that a small time step would be used thus reducing the temporal discretization error present in the solution. A time step for the 41x21 grid was chosen such that $\Delta t=0.001$. This time step was then halved as the grid spacing was also halved. Thus for the 81x41 grid $\Delta t=0.0005$ and for the 161x81 grid $\Delta t=0.00025$.

In the interest of time it was desirable to only investigate the performance of the ETE methodology on as few time steps as possible. Because of the presence of the time derivative in the error source term it was necessary to calculate the time derivative of the numerical solution in the continuous domain. While the spatial derivatives are still being calculated using the WSM, it was decided that the time derivatives would be calculated using 2nd order finite differencing. This requires a minimum of three time steps in each solution. Therefore, the solution was completely converged at three time steps on the 41x21 grid. This results in a maximum time of only 0.003 on the coarse grid solution. Because the time step is halved on each successively finer grid, the solution on each grid must be converged at twice as many time steps (six for the 81x41 grid and twelve for the 161x81 grid). Given that the maximum value of time considered in this study is 0.003 the kinematic viscosity was set to a constant value such that $\nu=0.1$. Substituting these variables into the equation for shear layer thickness results in a maximum shear layer thickness mentioned in the first paragraph of this section.

The suddenly accelerated plate problem was solved using the same in-house developed code that has been applied in the rest of this study. The solver utilizes the finite volume method with 2nd order central differencing used to discretize the diffusion terms and a hybrid scheme (order of 1st – 2nd) on the convection terms which applies upwind or central differencing based on the cell Peclet number. The pressure-velocity coupling was

handled using the SIMPLE algorithm. This transient problem was solved using a fully implicit scheme as to eliminate any limitations on the Courant number.

The boundary conditions for the suddenly accelerated plate problem are quite easily implemented. The v velocity is set equal to zero at all boundaries thus imposing the parallel flow condition that was discussed earlier. The only velocity considered in this problem is the u velocity. The fluid in the computational domain is initially at rest. At time zero the south boundary (plate) is put into motion. In this case the velocity at the south boundary was set such that $U_0=1.0$. The boundary conditions at the west and east boundaries were set to ensure that the flow was not changing in the x coordinate direction. Thus $\frac{\partial u}{\partial x} = 0$ at both the west and east boundaries. Finally, the north boundary is set as the free stream fluid which remains uninfluenced by the accelerated plate. Thus the boundary condition applied at the north boundary is $\frac{\partial u}{\partial y} = 0$.

The final discretized equations were solved using the TDMA line by line iteration method. The solver alternates sweeping direction from iteration to iteration to allow the boundary conditions to propagate through the flow field as efficiently as possible. The solution was allowed to converge to machine accuracy at each time step. This ensures that the residuals do not need to be taken into account when solving the error transport equation.

9.2.2 Solution Verification

Solutions were generated on three grids using the methodology described above. All time steps on each grid were allowed to converge fully so that an accurate solution was solved for at each time step. As the simplifying assumptions associated with this problem result in parallel flow, the v velocity is zero everywhere and will not be represented in this study. Verification and validation in previous studies has typically started with a qualitative inspection of velocity contour plots. However, as the boundary conditions applied to this problem force fully developed parallel flow, the resulting solution has no variation in the x coordinate direction thus making contour plots of little use. For the problem of the suddenly accelerated plate all data will be presented as a profile in the y coordinate direction. Profiles of the u velocity on all grids are provided in Figure 9.3 through Figure 9.5. Each plot includes both the analytical and numerical u

velocity profile at $t=0.001$, $t=0.002$, and $t=0.003$. This corresponds to all of the time steps calculated on the 41×21 grid. More time steps were solved on the finer grids but only these time steps will be shown as they are common among every grid.

The first important observation to be made is that the boundary conditions that were imposed in the model are being satisfied as anticipated. The velocity at the south boundary is 1.0 and in agreement with the analytical solution. At the north boundary condition of $\frac{\partial u}{\partial y} = 0$ is certainly being satisfied. As anticipated the computational domain height of 0.3 appears to be more than enough to represent the free stream given the shear layer thickness that is developed in this study.

The second important observation is that the agreement between the numerical solution and the analytical solution is obviously improving as the grid spacing and time step are reduced. There actually appears to be a significant amount of error in the solution on the 41×21 grid. However, the agreement with the analytical solution improves dramatically on the 81×41 grid. The improvement observed from further refinement of the grid does not appear to be nearly as significant. The shear layer thickness is increasing as anticipated with time. Ultimately it appears that the in-house developed solver appears to be more than capable of solving this problem.

Figure 9.3: U Velocity Profiles at $t=0.001$, $t=0.002$, and $t=0.003$ on the 41x21 Grid

Figure 9.4: U Velocity Profiles at $t=0.001$, $t=0.002$, and $t=0.003$ on the 81x41 Grid

Figure 9.5: U Velocity Profiles at $t=0.001$, $t=0.002$, and $t=0.003$ on the 161x81 Grid

9.3 Solution of the Error Transport Equation

The ETE derived from first principles here is identical to what was derived for use in the cases of flow between two parallel plates and lid driven cavity flow. The equations are repeated below for reference purposes only.

Continuity Error Transport Equation

$$\frac{\partial \varepsilon_u}{\partial x} + \frac{\partial \varepsilon_v}{\partial y} = -\tau_m \quad 9.3$$

U Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_u)}{\partial t} + 2 \frac{\partial(u^n \varepsilon_u)}{\partial x} + \frac{\partial(v^n \varepsilon_u)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial x} - \mu \left[\frac{\partial^2(\varepsilon_u)}{\partial x^2} + \frac{\partial^2(\varepsilon_u)}{\partial y^2} \right] + \rho \frac{\partial(u^n \varepsilon_v)}{\partial y} = -\tau_u \quad 9.4$$

V Velocity Error Transport Equation

$$\rho \left[\frac{\partial(\varepsilon_v)}{\partial t} + \frac{\partial(u^n \varepsilon_v)}{\partial x} + 2 \frac{\partial(v^n \varepsilon_v)}{\partial y} \right] + \frac{\partial(\varepsilon_p)}{\partial y} - \mu \left[\frac{\partial^2(\varepsilon_v)}{\partial x^2} + \frac{\partial^2(\varepsilon_v)}{\partial y^2} \right] + \rho \frac{\partial(v^n \varepsilon_u)}{\partial x} = -\tau_v \quad 9.5$$

Where τ_m , τ_u , and τ_v are calculated by mapping the numerical solution at each time step to the continuous domain using the WSM and evaluating the original conservation equations. For simplicity, the time derivative is estimated using 2nd order finite differencing at this time. In the future it should be possible to include the time variable in the WSM. As with the u velocity in the previous section, the simplifying assumptions force the flow to only vary in the y coordinate direction so again the use of contour plots would be ineffective. Profiles of the continuity error source term on all grids for t=0.001, t=0.002, and t=0.003 are provided in Figure 9.6. Profiles of the u momentum error source term on all grids for t=0.001, t=0.002, and t=0.003 are provided in Figure 9.7. Finally, profiles of the v momentum error source term on all grids for t=0.001, t=0.002, and t=0.003 are provided in Figure 9.8.

Figure 9.6: Continuity Error Source Term τ_m at All Time Steps on All Grids

Figure 9.7: U Momentum Error Source Term τ_u at All Time Steps on All Grids

Figure 9.8: V Momentum Error Source Term τ_v at All Time Steps on All Grids

The first obvious observation from an inspection of the error source terms is that both the continuity and v momentum source terms are negligibly small. This is again due to the simplifying assumptions made for the suddenly accelerated plate problem. The only source terms that has values large enough to take into account is the u momentum error source term. Here, as would be anticipated, the largest sources of error are near the south boundary at the moving plate. The error source term decays to zero as the profile moves toward the north boundary. The u momentum error source terms on all grids have a similar shape but different magnitudes. The error source terms do vary from time step to time step thus necessitating the calculation of a new error source term at each time step.

The application of the boundary conditions in the suddenly accelerated plate problem is similar to that of previous problems considered in this study. The error in the v velocity was set equal to zero at all boundaries. Technically, the v velocity error at the west and east boundaries is set equal to the error associated with the linear interpolation used as a result of the staggered grid. However, as the v velocity is essentially zero everywhere in the computational domain this error is also essentially zero. The u velocity error at the south boundary was also set based on the error in the linear interpolation used to set the velocity at that boundary. This error does take on a finite value and is represented well by the ETE solution. The u velocity error boundary condition at the west and east boundaries was set analogous to the u momentum condition at those boundaries using the Neumann boundary condition $\frac{\partial \varepsilon_u}{\partial x} = 0$. The u velocity error boundary condition at the north wall was also set analogous to the u momentum condition at that boundary using the Neumann boundary condition $\frac{\partial \varepsilon_u}{\partial y} = 0$.

Using the calculated error source terms and the described boundary conditions, the ETE is solved on the three meshes considered in this study. The same discretization scheme that was applied to the governing PDEs was also applied to the ETE. The solution was allowed to fully converge at each time step so that no iterative convergence error needed to be taken into account. The true error was calculated using the analytic solution discussed earlier in this chapter. Again, profiles will be used here to compare the ETE solution to the true error. Profiles of the u velocity error on all grids are provided at each time step in Figure 9.9 through Figure 9.11.

Figure 9.9: Profiles of the ETE Solution and True Error on all Grids at $t=0.001$

Figure 9.10: Profiles of the ETE Solution and True Error on all Grids at $t=0.002$

Figure 9.11: Profiles of the ETE Solution and True Error on all Grids at $t=0.003$

An inspection of the error profiles indicates that the ETE solution is in excellent qualitative and quantitative agreement with the true error on all grids at all time steps. The locations of the maximum and minimum values of error are very well represented by the ETE solution on all grids at all time steps. The agreement between the ETE solution and the true error is nearly perfect at all time steps on the 81x41 and 161x81 grids. The ETE solution on the 41x21 grid does not match the true error as well as the solutions on the other grids but the agreement is still quite reasonable.

Another important observation is that both the true error and ETE solution appear to be reducing proportional to grid spacing at all time steps as would be expected if the solver is performing as anticipated. To better represent this important observation, the L2 normal of both the ETE solution and the true error was calculated on every grid at every time step. The L2 normal of the true error is plotted in Figure 9.12 while the L2 normal of the ETE solution is plotted in Figure 9.13. The profiles in the plot of the L2 normal of the true error serve as a final validation of the numerical model for this problem. The L2 normal of the true error at every time step decays proportion to the grid spacing at a rate that is consistent with the computational model. The L2 normal of the true error decays proportional to the grid spacing raised to the power of 1.758, 1.812, and 1.781 at $t=0.001$, $t=0.002$, and $t=0.003$ respectively. Because the discretization scheme employed in this study is a hybrid $1^{st}/2^{nd}$ order scheme these values are quite reasonable. The same analysis was also completed using the ETE solution. Unfortunately the results were not quite as favorable. The L2 normal of the ETE solution decays proportional to the grid spacing raised to the power of 0.783, 1.391, and 1.770 at $t=0.001$, $t=0.002$, and $t=0.003$ respectively. The rate of error decay is reasonable for $t=0.002$ and very good for $t=0.003$ however, the rate of error decay at $t=0.001$ is in very poor agreement. However it should be noted that this is entirely due to the comparatively poor prediction of the true error on the 41x21 grid. The 41x21 grid solution is not considered the L2 normal of the ETE solution decays at the anticipated rate at all time steps.

Ultimately, the ETE methodology applied to the unsteady case of a suddenly accelerated flat plate performed exceedingly well. The ETE solution provided an excellent qualitative and quantitative estimate of the true error. There is much promise going forward with further applications to unsteady problems.

Figure 9.12: L2 Normal of the True Error as a Function of Grid Spacing

Figure 9.13: L2 Normal of the ETE Solution as a Function of Grid Spacing

Chapter 10

Conclusions and Future Work

10.1 Conclusions

Given the ever increasing reliance on numerical analysis in the industry, it is has become imperative that the numerical analyst provides a reasonable estimation of the discretization error along with the intended results of the analysis. Currently, the most often utilized method for the prediction of discretization error is Richardson Extrapolation. This method involves using solutions on three successively finer grids in order to extrapolate to the exact solution, thus allowing the approximation of the error. If Richardson Extrapolation is applied properly, the methodology is capable of very accurate error prediction. However, the problem is that solutions must be generated on at least three different grids for proper application of the method. Additionally, all of the solutions must be in the asymptotic regime, which could potentially require even more solutions to ensure that this is the case. Generating solutions on multiple grids is not problematic when considering very simplistic problems. On the other hand, when considering complex problems of practical interest in the industry, generating solutions on multiple grids can be prohibitively time consuming. Because of this, alternate methods of error prediction have been an increasingly common subject in the literature.

A literature review has been completed regarding the topic of discretization error prediction using the ETE. It is readily apparent that the methodology behind the derivation of the ETE and its solution are reasonably straight forward. The real complication with this method is the calculation of the resulting source term in the ETE. Regardless of how the ETE is derived the calculation of the error source term will be problematic.

A good deal of research has been conducted in the area of error source term determination. For the most part the general consensus is that the majority of the suggested methods work reasonably well for smooth linear conservation equations but

struggle when applied to coupled, non-linear conservation equations (Navier-Stokes equations).

From the literature review, it was observed that one of the most implemented methods for calculating the error source term was to evaluate the original partial differential equation (PDE) with the numerical solution. This requires that the numerical solution be mapped to the continuous domain before it can be evaluated in the original PDE. In the reviewed publications, difficulties predicting the discretization error for coupled non-linear conservation equations were attributed to inadequate performance of the numerical solution data fit.

A method has been proposed to calculate the error source term using a data fitting method originally intended for generating smooth topographical functions from scattered data sets. The method is described as the weighted spline method. The original method has been modified by implementing the use of radial basis functions to improve the fit of the original function.

The methodology presented here has been verified using several common steady state benchmark cases including one-dimensional and two-dimensional advection diffusion, developing flow between parallel plates, square lid driven cavity flow, and backward facing step flow. Additionally, the ETE methodology has been applied to the unsteady case of a suddenly accelerated flat plate. Literature on the application of the ETE methodology to unsteady problems is severely lacking. The results of this study are very encouraging. The ETE methodology appears to be capable of generating accurate predictions of the true error across a wide range of problems.

The ETE methodology was first applied to the one-dimensional and two-dimensional advection diffusion problems. As it has been noted in the literature, the ETE methodology performs exceedingly well in these cases. In the one-dimensional case the ETE methodology was applied to solutions with increasing values of Peclet number. It was observed that the regions of largest error source were near areas of high gradient in the original solution, which intuitively makes sense. When extended to the two-dimensional advection diffusion problem, the ETE methodology continues to provide an accurate prediction of the discretization error at all locations. In all instances the ETE

solution accurately predicts the location and magnitude of the maximum and minimum values of error.

The real problem noted in the literature is the application of the ETE methodology to strongly coupled, non-linear, partial differential equations, such as the Navier-Stokes equations for the solution of a flow field. One of the simplest solutions to the Navier-Stokes equations is that of fully developed flow between parallel plates. The simplifications make it possible to generate a closed form analytic solution that is used to calculate the true error to be used to evaluate the performance of the ETE solution. An in-house analysis tool was used to calculate developing flow between parallel plates on three different grids and the ETE methodology was applied to the entire computational domain. The ETE solution was compared to the true error in the fully developed region of the flow near the outlet. The ETE solution was in nearly perfect agreement with the true error on all grids. Also, it was noted that the ETE solution predicted the appropriate error decay rate as grid spacing was reduced. A final important observation from this portion of the study was that because of the staggered grid employed in the model the error at the boundaries is not always zero. This is because the u velocity cannot be defined at the south and north boundaries and the v velocity cannot be defined at the west and east boundaries. The boundary conditions at these locations are implemented using a ghost point outside of the computational domain and linear interpolation. This results in some finite amount of error at the boundaries that must be accounted for.

The ETE methodology was also applied to more complicated problems including square lid driven cavity flow and backward facing step flow. These more complicated problems do not have closed form analytic solutions to use in the calculation of the true error. In the absence of true error there is no way to evaluate the performance of the ETE methodology. In these cases it is acceptable to substitute a well-accepted benchmark solution on a fine grid for the exact solution. These types of solutions exist in the literature for both of these problems. However, the use of these solutions requires the use of some form of interpolation to the grids applied in this study. Unfortunately, the interpolation error is often on the same order of magnitude or larger than the discretization error making comparisons to the ETE solution invalid. Because of this it was decided that the exact solution would be approximated using an exceedingly fine grid solution from the in-house solver that was utilized in this study. For both cases, the solution to the error transport equation agreed very well with the approximated true error.

Additional comparisons were made to the predicted error from Richardson Extrapolation techniques. In most instances Richardson Extrapolation produced a more accurate prediction of the discretization error. This is not entirely surprising as Richardson Extrapolation employs information from all grids rather than just one. The goal of the ETE methodology is not necessarily to produce more accurate error predictions than Richardson Extrapolation but rather to produce acceptably accurate error predictions while not being prohibitively time consuming. Again, as in the case of the developing flow between parallel plates, the ETE solution appears to reduce proportionally with grid spacing at a rate that is consistent with the discretization schemes employed in this study.

Finally, the ETE methodology was applied to the unsteady case of a suddenly accelerated flat plate. The application of the ETE methodology to unsteady problems is severely lacking in the literature. The application and solution of the ETE methodology is no more difficult than in the steady state case. There is just slightly more computational time required as the error source terms must be computed at each time step. In this study the flow field was resolved on three successively finer grids with successively smaller time steps. The solution of the ETE was compared to the true error and the results agreed exceedingly well. The agreement of the ETE solution and the true error was not as impressive on the coarsest grid as it was on the two finer meshes but it was still acceptable. Ultimately, it seems that the ETE methodology can easily be extended to unsteady problems.

Based on data gathered across a wide range of verification and validation problems, it seems that the ETE methodology applied here using the WSM to generate the error source terms is quite promising. Reasonable predictions of the discretization error can be generated using data from only one grid. This would significantly reduce the computational expense of error estimation in more complex flows. In the backward facing step flow considered in this study, the required computational time was 0.15 hours for the coarse grid, 0.27 hours for the medium grid, and 0.65 hours for the fine grid. The time required to perform Richardson Extrapolation is essentially negligible meaning the total time for the error prediction is 1.07 hours. The error transport equation methodology requires 0.15 hours for a coarse grid solution, 0.05 hours for the determination of the error source terms, and 0.10 hours for the solution of the error transport equations. The total time for error prediction is 0.3 hours. In this instance the time required for

Richardson Extrapolation is 3.57 times longer than the time required for the error transport equation methodology. Ultimately this improvement will be highly dependent on the size and complexity of the problem being solved. In the end this methodology may mean the difference between an analyst in the industry presenting an analysis of the discretization error with the results and just presenting unbounded results.

10.2 Future Work

Based on the experience gathered during the completion of this study, the following future work is recommended to further this research.

- The ETE methodology should be applied to increasingly complex unsteady problems. As with the steady state problems considered in this study, much understanding can be gained in regards to the performance of the presented methodology through application to more complex problems.
- While only two-dimensional problems were given consideration in this study, these methods should be equally applicable to three-dimensional problems as well. The WSM should be equally applicable in three-dimensions for the calculation of the error source terms.
- These methods should also be applied to non-uniform grids. This is one potential major benefit to using the WSM to calculate the error source terms instead of finite differencing as was employed in much of the literature. The WSM assumes a set of scattered data so a structured grid versus an unstructured grid should make no difference.
- The use of this methodology also extends beyond simple error predictions. One such extension would be solving the ETE in every iteration and then applying that error as a correction to the calculated flow variables. It is possible that this could potentially speed convergence while also providing a real time estimate of the discretization error.
- Finally, the source terms calculated in this methodology could be of potential use in grid refinement. Rather than applying additional grid points in the areas where the error ultimately ends up, it may be beneficial to apply these additional grid points to areas of the largest error source.

Bibliography

- [1] Mehta, U., Oberkampf, W., Sindir, M., and Conlisk, A., 1998, Guide for the Verification and Validation of Computational Fluid Dynamics Simulations, American Institute of Aeronautics and Astronautics.
- [2] Roy, C., 2005, "Review of Code and Solution Verification Procedures for Computational Simulation," J. Comput. Phys., **205**, pp. 131–156.
- [3] Roache, P. J., 2009, Fundamentals of Verification and Validation, Hermosa Publishers.
- [4] Celik, I., Li, J., Hu, G., and Shaffer, C., 2005, "Limitations of Richardson Extrapolation and Some Possible Remedies," J. Fluids Eng., **127**, pp. 795–805.
- [5] Celik, I., and Zhang, W. M., 1993, "Application of Richardson Extrapolation to Some Simple Turbulent Flow Calculations," Proc. of Symp. on Quantification of Uncertainty in Computational Fluid Dynamics, Washington, DC, pp. 29–38.
- [6] Celik, I., and Karatekin, O., 1997, "Numerical Experiments on Application of Richardson Extrapolation With Nonuniform Grids," J Fluids Eng, **119**, pp. 584–590.
- [7] Celik, I., Chen, C. J., Roache, P. J., and Scheurer, G., 1993, "Quantification of Uncertainty in Computational Fluid Dynamics," ASME Fluids Engineering Summer Meeting, Washington, DC.
- [8] Roache, P. J., 1993, "A Method for Uniform Reporting of Grid Refinement Studies," Prec. of Symp. on Quantification of Uncertainty in Computational Fluid Dynamics, I. Celik, ed., ASME, New York, pp. 109–120.
- [9] Roache, P. J., 1998, "Verification and Validation in Computational Science and Engineering," Hermosa Publ.
- [10] Celik, I., and Hu, G., 2004, "Single Grid Error Estimation Using Error Transport Equation," J. Fluids Eng., **126**, pp. 778–790.
- [11] Zhang, X. D., Trepanier, J. Y., and Camarero, R., 1997, "An A Posteriori Error Estimation Method Based on an Error Equation," Proc. of the AIAA 13th CFD Conf., Snowmass, CO.
- [12] Roache, P. J., 1998, Fundamentals of Computational Fluid Dynamics, Hermosa Publishers.
- [13] Van Straalen, B., 1996, "A Posteriori Error Estimation for Finite Volume Simulations of Fluid Flows," Masters, University of Waterloo.
- [14] Zhang, X. D., Trepanier, J. Y., and Camarero, R., 2000, "A Posteriori Error Estimation for Finite-Volume Solutions of Hyperbolic Conservation Laws," Comput. Methods Appl. Mech. Eng., **185**, pp. 1–19.
- [15] Ilinca, C., Zhang, X. D., Trepanier, J. Y., and Camarero, R., 2000, "A Comparison of Three Error Estimation Techniques for Finite Volume Solutions of Compressible Flows," Comput. Methods Appl. Mech. Eng., **189**, pp. 1277–1294.
- [16] Celik, I., and Hu, G., 2002, "Discretization Error Estimation Using Error Transport Equation," Proceedings ASME FEDSM.
- [17] Hay, A., and Visonneau, M., 2011, "Error Estimation Using the Error Transport Equation for Finite-Volume Methods and Arbitrary Meshes," Int. J. Comput. Fluid Dyn., **20**(7), pp. 463–479.
- [18] Cavallo, P. A., and Sinha, N., 2007, "Error Quantification for Computational Aerodynamics Using an Error Transport Equation," J. Aircr., **44**(6), pp. 1954–1963.
- [19] Junkins, J., Miller, G., and Jancaitis, J., 1973, "A Weighting Function Approach to Modeling of Irregular Surfaces," J. Geophys. Res., **78**(11), pp. 1794–1803.

- [20] Roy, C., and Sinclair, A., 2009, "On the Generation of Exact Solutions for Evaluating Numerical Schemes and Estimating Discretization Error," *J. Comput. Phys.*, **228**, pp. 1790–1802.
- [21] Hardy, R., 1990, "Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988," *Comput. Math. Appl.*, pp. 163–208.
- [22] Carr, J. C., Beaston, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., and McCallum, B. C., 2001, "Reconstruction and Representation of 3D Objects with Radial Basis Functions," *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New Zealand, pp. 67–76.
- [23] Ghia, U., Ghia, K. N., and Shin, C. T., 1982, "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method," *J. Comput. Phys.*, **48**, pp. 387–411.
- [24] Goldstein, R. J., Eriksen, V. J., Olson, R. M., and Eckert, E. R. G., 1970, "Laminar separation reattachment and transition flow over a downstream-facing step," *Trans. ASME J. Basic Eng.*, **92**, p. 732.
- [25] Denham, M. K., and Patrick, M. A., 1974, "Laminar flow over a downstream facing step in a two-dimensional flow channel," *Trans. Inst. Chem. Eng.*, **52**, pp. 361–367.
- [26] Ethridge, D. W., and Kemp, P. H., "Measurements of turbulent flow downstream of a backward facing step," *J. Fluid Mech.*, **86**, p. 545.
- [27] Armaly, B. F., Durst, F., Pereira, J. C. F., and Schoenung, B., "Experimental and theoretical investigation of backward-facing step flow," *J. Fluid Mech.*, **127**, pp. 473–496.