

Efficient High-Order Accurate Unstructured Finite-Volume Algorithms for Viscous and Inviscid Compressible Flows

by

Christopher Michalak

B.A.Sc., The University of British Columbia, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April, 2009

© Christopher Michalak 2009

Abstract

High-order accurate methods have the potential to dramatically reduce the computational time needed for aerodynamics simulations. This thesis studies the discretization and efficient convergence to steady state of the high-order accurate finite-volume method applied to the simplified problem of inviscid and laminar viscous two-dimensional flow equations. Each of the three manuscript chapters addresses a specific problem or limitation previously experienced with these schemes. The first manuscript addresses the absence of a method to maintain monotonicity of the solution at discontinuities while maintaining high-order accuracy in smooth regions. To resolve this, a slope limiter is carefully developed which meets these requirements while also maintaining the good convergence properties and computational efficiency of the least-squares reconstruction scheme. The second manuscript addresses the relatively poor convergence properties of Newton-GMRES methods applied to high-order accurate schemes. The globalization of the Newton method is improved through the use of an adaptive local timestep and of a line search algorithm. The poor convergence of the linear solver is improved through the efficient assembly of the exact flux Jacobian for use in preconditioning and to eliminate the additional residual evaluations needed by a matrix-free method. The third manuscript extends the discretization method to the viscous fluxes and boundary conditions. The discretization is demonstrated to achieve the expected order of accuracy. The fourth-order scheme is also shown to be more computationally efficient than the second-order scheme at achieving grid-converged values of drag for two-dimensional laminar flow over an airfoil.

Contents

Abstract	ii
Contents	iii
List of Tables	vi
List of Figures	vii
Acknowledgements	ix
Co-Authorship Statement	x
1 Introduction	1
1.1 Methods in Computational Aerodynamics	2
1.1.1 Modeling	2
1.1.2 Discretization	4
1.1.3 Numerical Solution	6
1.2 Overview of the Solver	7
1.2.1 Governing Equations	8
1.2.2 Vertex Centered Finite-Volume Discretization	9
1.2.3 Reconstruction	10
1.2.4 Interior and Boundary Flux Evaluation	14
1.2.5 Flux Integration	14
1.3 Objectives and Outline	16
1.4 Bibliography	19

2	Monotonicity Enforcement Using a Limiter	24
2.1	Introduction	24
2.2	High-Order Accurate Solution Reconstruction	25
2.3	Second-Order Limiting	27
2.4	High-Order Limiting	30
2.4.1	Monotonicity	30
2.4.2	Accuracy	32
2.4.3	Uniform Regions and Smooth Extrema	34
2.4.4	Boundary Treatment	35
2.4.5	Complete Algorithm	39
2.5	Results	40
2.5.1	Ringleb's Flow	41
2.5.2	Transonic Flow Over an Airfoil	48
2.6	Conclusion	56
2.7	Bibliography	59
3	Efficient Convergence to Steady State	62
3.1	Introduction	62
3.2	High-Order Accurate Finite-Volume Method	64
3.2.1	Governing Equations	64
3.2.2	Finite-Volume Formulation	64
3.2.3	Solution Reconstruction	65
3.2.4	Numerical Flux Calculation and Integration	68
3.3	Globalized Newton Method	69
3.3.1	Pseudo-Timestepping	69
3.3.2	Line Search	70
3.3.3	Selecting the Timestep	71
3.4	Preconditioned GMRES	74
3.4.1	Matrix-Free Method	75
3.4.2	Forming the High-Order Jacobian Matrix	75
3.4.3	Preconditioning	78
3.4.4	Startup	79
3.5	Results	80

Contents

3.5.1	Test Cases	80
3.5.2	Globalization Strategy	81
3.5.3	Solving the Linear System	82
3.6	Conclusion	95
3.7	Bibliography	98
4	Extension to Viscous Flows	101
4.1	Introduction	101
4.2	Governing Equations	102
4.3	Solution Reconstruction	103
4.3.1	Reconstruction for Interior Control Volumes	103
4.3.2	Boundary Condition Enforcement	107
4.4	Flux Integration	111
4.5	Solution Convergence	112
4.6	Results	113
4.6.1	Cylindrical Couette Flow	113
4.6.2	Flow Over an Airfoil	116
4.7	Conclusion	123
4.8	Bibliography	124
5	Conclusion	127
5.1	Significance of Results	127
5.1.1	Limiter	128
5.1.2	Globalized Matrix-Explicit Newton-GMRES	130
5.1.3	Viscous Discretization	131
5.2	Outlook	132
5.2.1	Turbulent Flows	132
5.2.2	Three Dimensional Flows	135
5.3	Bibliography	137

List of Tables

2.1	Convergence order of error in pressure for Ringleb's flow . . .	48
2.2	Computational time for transonic airfoil test case	54
3.1	Relative computational time of flux and Jacobian evaluation without limiter for subsonic case	83
3.2	Relative computational time of flux and Jacobian evaluation with limiter for transonic case	83
3.3	Average number of inner GMRES iterations per Newton iteration	87
3.4	Memory use and CPU time for subsonic case	93
3.5	Memory use and CPU time for transonic case	94
4.1	Order of error norms in velocity for cylindrical Couette flow .	114
4.2	Computational time for airfoil case	120
4.3	Comparison of fine grid results for airfoil case	120

List of Figures

1.1	Median dual control volume	10
1.2	Example reconstruction stencil	12
1.3	Gauss integration for a control volume at a curved boundary	15
2.1	Venkatakrishnan's smooth approximation to $\min(1, y)$	29
2.2	$\widetilde{\min}(1, y)$ compared to $\min(1, y)$	33
2.3	Transition function used to disable the limiter in nearly uniform regions	35
2.4	Two coarsest meshes used for Ringleb's flow test case	41
2.5	Mach number contours from Ringleb's flow	43
2.6	Limiter value for pressure for Ringleb's flow	45
2.7	Entropy for Ringleb's flow	46
2.8	Error convergence for pressure in Ringleb's flow	47
2.9	Mesh for NACA 0012 airfoil case	49
2.10	Entropy for transonic flow over NACA 0012 airfoil	51
2.11	Total pressure for transonic flow over NACA 0012 airfoil	52
2.12	Surface pressure profiles for transonic flow over a NACA0012 airfoil	53
2.13	Convergence history for transonic airfoil case	54
2.14	Surface pressure sensitivity to limiter tuning parameter	55
2.15	Total pressure sensitivity to limiter tuning parameter	57
3.1	Mesh for subsonic multi-element airfoil case	83
3.2	Pressure contours for subsonic flow over a multi-element airfoil	84
3.3	Mesh for transonic NACA 0012 airfoil case	85

List of Figures

3.4	Surface pressure coefficient for transonic flow over a NACA 0012 airfoil	87
3.5	Convergence plots for subsonic case using line search method	88
3.6	Convergence plots for transonic case using line search method	89
3.7	Convergence plots for subsonic case using different timestep- ping methods	90
3.8	Convergence plots for transonic case using different timestep- ping methods	91
3.9	Convergence for matrix-explicit method compared to matrix- free method	96
4.1	Four coarsest grids used in cylindrical Couette flow case . . .	115
4.2	Convergence of error in velocity for cylindrical Couette flow .	116
4.3	Effect of change in boundary reconstruction weight	117
4.4	Three coarsest grids used for airfoil case	118
4.5	Mach number contours for airfoil case	119
4.6	Computational time for airfoil case	119
4.7	Viscous drag coefficient grid convergence	121
4.8	Pressure drag coefficient grid convergence	122
4.9	Grid convergence of the separation point on the upper airfoil surface	122
5.1	Effect of curved boundary on anisotropic control volume . . .	135

Acknowledgements

First and foremost I would like to thank all of the faculty that have guided me in these years as a graduate student. My supervisor, Dr Carl Ollivier-Gooch deserves a special thank you. Carl is a wonderful person, who also happens to be a great research adviser.

I would also like to acknowledge all of my student colleagues which I have had a pleasure of working with during this journey. Amongst these I was especially lucky to have a friend in the lab who was along for the whole ride. Serge, thanks for not letting me forget French!

I am also grateful to all my friends with whom I've shared my life with outside of work, usually in the mountains. The perspective on life you provide helps me make sense of it all. A very special thank you is due to Krystil with whom I have shared my life for the last few years. Krystil, you complete me.

These acknowledgements could not possibly be complete without mentioning my parents. Each in your own way, you have made me who I am, and have always supported me. Thank you.

Co-Authorship Statement

The research ideas and methods explored in the three co-authored manuscripts of this thesis are the fruits of a close working relationship between Dr Carl Ollivier-Gooch and Christopher Michalak. The implementation of the methods, the data analysis, and the manuscript preparation were done by Michalak with invaluable guidance from Ollivier-Gooch throughout the process.

Chapter 1

Introduction

For almost all cases of practical interest, the equations governing fluid flow cannot be solved analytically. Therefore engineers and scientists needing to predict flow properties must depend on physical experiments or on the numerical solution of the flow equations. The latter method, commonly known as computational fluid dynamics (CFD) has grown to be an important and broad field of study. Nonetheless, although immensely useful, these methods have yet to reach a level of accuracy sufficient to completely replace the need for costly and time consuming experimental work.

Although many aspects of the work presented in this thesis could be used for other applications, the immediate focus is on aerodynamics. To date, the most common goal of computational aerodynamics is the prediction of net aerodynamic forces of lift and drag to help guide the design process. Early successes in performing two-dimensional viscous simulations over airfoils and three-dimensional inviscid simulations over aircraft were documented in a review paper by Jameson in the mid 1980s [19]. The results available at the time were sufficiently accurate to gain a qualitative understanding of the flow but highly inaccurate for the quantitative prediction of drag and lift. Since then, advances in modeling, numerical methods, and computational power have greatly improved the fidelity of the results. Nonetheless the accuracy of simulations involving three-dimensional turbulent flow around aircraft configurations is still inadequate for many practical engineering purposes. In recent years a series of drag prediction workshops has been held to evaluate the state of the art in this domain. Studies of the combined results [18] of these workshops have demonstrated that both the numerical methods and the turbulence models used are responsible for these inaccuracies. This work represents a modest contribution towards improving the accuracy of these

types of simulations. However the outlook for computational aerodynamics, and therefore the need for efficient numerical methods, does not end with the accurate prediction of lift and drag. As methods improve and computational power increases, CFD applications will expand to the automatic optimization of aerodynamic configuration, the study of fluid structure interaction, and the study of flight dynamics.

1.1 Methods in Computational Aerodynamics

There are three essential components to the numerical simulation of fluid flow:

1. *Modeling.* The partial differential equations describing the physics of the problem must be defined.
2. *Discretization.* These partial differential equations must be discretized thereby transforming them into a large system of algebraic equations.
3. *Numerical Solution.* The value of the discrete flow properties satisfying the system of algebraic equations must be found.

All three of these aspects remain active fields of research. This thesis introduces new contributions in the areas of discretization and numerical solution which together can be called the numerical method. In this section all three aspects will be introduced with an emphasis on methods most relevant to the current work.

1.1.1 Modeling

For engineering purposes, the Navier-Stokes equations accurately describe the physics of air flow. However the turbulent nature of aerodynamic flows results in large disparities between time and length scales of small eddies and macroscopic flow features. Since, in aggregate, these turbulent eddies have an important effect on the flow they cannot simply be ignored. Currently available computational power is grossly inadequate to solve the discretized

Navier-Stokes equations sufficiently accurately to capture the effects of turbulence for a typical aircraft configuration. Using an assumption that computational power increases by a factor of 5 every five years, Spalart estimated [35] that the direct simulation of turbulence for aircraft will not be possible until the year 2080.

For these reasons the Reynolds averaged Navier-Stokes (RANS) equations are preferred. To close this system of equations a turbulence model is needed to estimate the effects of the fluctuation in the velocity field. The development of these models, which themselves are typically systems of partial differential equation, is an important field of research. The overall accuracy of the solution will depend on the fidelity of the turbulence model and on how accurately the RANS and turbulence model equations are solved by the numerical method.

Large eddy simulation (LES) is a hybrid method in which effects of smaller eddies are simulated with empirical models while larger eddies are simulated directly. This approach can yield much more accurate results than the RANS method, but requires much greater accuracy. In 2000, Spalart estimated [35] that the grid size needed to perform LES simulations using existing numerical methods would be a factor of 30,000 larger than that needed to perform RANS simulations. The cost of LES simulations is further exacerbated by the lack of a steady-state solution necessitating averaging of a time-accurate evolution of the solution. For these reasons Spalart estimates that the computational resources to achieve this using existing numerical approaches will not be available until 2045. Since the benefit of high-order accurate methods increases with the desired solution resolution, the prospects for these methods is even more promising for LES simulations than it is for RANS simulations.

Although the numerical methods introduced in this work are ultimately intended to be applied in conjunction with a turbulence model, they are only demonstrated for inviscid and laminar flows. However some aspects of the extension of our methods to the turbulent flows will be addressed in the concluding chapter.

1.1.2 Discretization

The purpose of discretization is to define a set of algebraic equations that approximate the partial differential equations of the model. A number of broad approaches exist for this. The most commonly used approach in computational aerodynamics, and the one used in this work, is the finite-volume method. In this method the spatial domain considered is tessellated into non-overlapping control volumes. Algebraic equations are then used to relate the control volume average states of the flow properties.

1.1.2.1 Choice of grids

Structured grids are attractive because they simplify many aspects of the solver. However generating structured grids around complex geometries is challenging. Even when a structured grid can be fitted, it often leads to grid spacing which is poorly adapted to the expected local flow gradients. Some methods, such as multi-block grids and overlapping grids, can be used to help alleviate some of these limitations. However structured grids remain difficult to generate for non-trivial geometries.

For these reasons unstructured grids have gained popularity in computational aerodynamics and are used in the present work. They can be fitted to geometries of any complexity and the local grid density can readily be controlled. Unlike with structured grids, the process of generating the mesh can be almost completely automated. However unstructured grids also lead to greater complexity of the solver since the mesh connectivity must be stored and since the discretization cannot take advantage of a structured topology. The first solutions of the Euler equations on unstructured grids were carried out using the finite-element method in the 1980s [1, 22]. Shortly thereafter work using the finite-volume method was carried out by Jameson and Mavriplis [21].

1.1.2.2 Order of accuracy

For any consistent numerical method, the error resulting from the discretization of the partial differential equations should decrease when the grid is

refined. Specifically, a discretization will produce an error of $\mathcal{O}(h^p)$ where h is the mesh length scale and p is the order of accuracy. The disadvantage of methods achieving a higher order of accuracy is that they are more complicated to implement and are more computationally expensive on a given grid. However to achieve the same level of accuracy a high-order accurate method can use a coarser grid than a lower-order accurate method. Therefore, from a computational cost perspective, the choice of discretization accuracy is not an obvious one. Due to their relative simplicity, second-order accurate schemes are by far the most commonly used methods for computational aerodynamics. The question of whether high-order accurate schemes could achieve desired levels of accuracy more efficiently has not been adequately addressed.

A third-order accurate method for structured grids has been studied by De Rango and Zingg [12]. In this study two-dimensional turbulent flow over airfoils was considered. The high-order accurate method was shown to obtain adequately accurate solutions on coarser grids than the second-order scheme. Furthermore the overall computational cost of attaining this level of accuracy was shown to be lower for the third-order scheme than the second-order scheme due to the coarser mesh requirement.

On unstructured grids a nominally third-order accurate finite-volume methods on unstructured grids for the Euler equations was introduced by Barth and Frederickson [3–5]. This work introduced k -exact reconstruction, which was further studied by other researchers [13, 16, 38]. These latter works used a third-order k -exact reconstruction for the advective fluxes and Green-Gauss gradient to compute viscous fluxes. Although a qualitative assessment of accuracy for practical cases was made in these works, the effective nominal accuracy of the scheme has not been demonstrated. Furthermore the superiority of the method, in terms of computational effort to attain a level of accuracy relative to second-order schemes, has also not been demonstrated. Considerable work using the Euler equation has also previously been carried out by others in the present author’s group [26–28]. These works successfully demonstrated that the nominal level of accuracy for schemes up to fourth-order can be obtained for shockless flows. Work in the

group has also been previously carried out demonstrating the effectiveness of the k -exact reconstruction in obtaining high-order accurate solutions of the advection-diffusion equation [29].

The discontinuous Galerkin finite-element method has also been successfully applied to the high-order accurate solution of compressible flow [7, 8, 15]. As with the high-order finite-volume method, the discontinuous Galerkin method has not yet matured enough to be used in production codes. The computational efficiency, in particular, has not been shown to exceed that of second-order finite-volume methods in common use. A disadvantage of this method is that, unlike in the finite-volume method, the number of degrees of freedom increases with the order of accuracy on a given mesh.

1.1.3 Numerical Solution

Once a spatial discretization has been selected, the remaining step is to address the time evolution of the system. For aerodynamics problems modelled using RANS, often only the steady-state solution is of interest. In this case, the time dependent terms could, in theory, be eliminated from the system of equations. If this is done, the remaining problem is to find the root of the equations defining the space discretization. In practice this root finding problem is most easily solved by reintroducing a time-like term to the equations. Therefore, whether a time-dependent or steady-state solution is sought, some form of time integration is needed.

Explicit time integration methods, such as the Runge-Kutta schemes, are relatively simple to implement since they only require a means of evaluating the residual of the space discretized equations. However their stability is limited by the Courant–Friedrichs–Lewy (CFL) condition. This limits the size of the timesteps that can be taken relative to the grid spacing. For typical grids used in computational aerodynamics this results in the need for an inordinate number of residual evaluations. Therefore the computational effort needed by these schemes is prohibitively high. For steady-state problems, the efficiency of explicit time-advance methods can be greatly improved through the use of multigrid methods [11, 20, 23, 24, 30].

Alternatively, implicit time integration methods can be used. Unlike explicit methods, the timestep is not restricted by the CFL condition. Implicit methods can therefore be substantially more efficient, particularly when only a steady-state solution is needed. At the limit when the timestep is infinitely large, the approach reduces to the Newton method. Typically the solution strategy is to use an implicit timestep until the solution is sufficiently converged to transition to a full-Newton method. Alternatively, the timestep can be ramped up gradually and the method approaches the Newton method asymptotically. However implicit methods are considerably more complex to implement since the solution of a large linear system involving the Jacobian of the residual is needed.

Early approaches for solving the linear systems arising from the implicit method were based on direct solution techniques [2, 36, 37]. However the memory and processing requirements of direct methods quickly become intractable. Therefore attention quickly turned to solving these linear systems approximately using iterative methods. In particular Generalized Minimal Residual (GMRES) [33], part of the Krylov subspace family of methods, has been shown to be very effective. On structured grids Pueyo and Zingg [31] demonstrated an efficient Newton-GMRES method for aerodynamic flow computations. A number of researchers have successfully applied Newton-GMRES on unstructured grids [6, 9, 26, 39].

1.2 Overview of the Solver

This section provides a brief overview of the high-order accurate finite-volume solver used in this work. Focus will be placed on the components of the solver that are not thoroughly addressed in the manuscript chapters. Specifically, the focus here will be on the spatial discretization of the inviscid flow equations. The extension to the viscous case is addressed in Chapter 4 and strategies for solving the discrete equations for steady state problems are addressed in Chapter 3.

1.2.1 Governing Equations

The two-dimensional Euler equations for a control volume V_i can be written in integral form as

$$\frac{d}{dt} \int_{V_i} U dA + \oint_{\Omega_i} F dS = 0 \quad (1.1)$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad F = \begin{bmatrix} \rho u_n \\ \rho u u_n + P \hat{n}_x \\ \rho v u_n + P \hat{n}_y \\ (E + P) u_n \end{bmatrix} \quad (1.2)$$

where U is the solution vector in conservative variables, F is the flux vector normal to the control volume boundary, \hat{n}_x and \hat{n}_y are the unit normal components, and $u_n = u\hat{n}_x + v\hat{n}_y$. The conserved variables consist of the density (ρ), two components of momentum (ρu and ρv), and energy (E). Air is modeled as an ideal gas with the equation of state

$$P = \rho R T$$

where P is pressure, R is the specific gas constant for air, and T is the temperature. The following thermodynamic relationships are also used

$$\gamma = \frac{C_p}{C_v}, \quad C_p = \frac{\gamma R}{\gamma - 1}, \quad C_v = \frac{R}{\gamma - 1}, \quad R = C_p - C_v$$

$$e = C_v T, \quad e_t = e + \frac{1}{2}(u^2 + v^2), \quad E = \rho e_t, \quad h = e_t + \frac{P}{\rho}$$

where γ is the ratio of specific heats for air, C_p and C_v are the specific heats, e is the specific internal energy, e_t is the specific total energy, E is the total energy, and h_t is the specific total enthalpy. An important property for compressible flows is the Mach number M which is the ratio of the flow speed to the speed of sound a

$$M = \frac{\sqrt{u^2 + v^2}}{a}$$

1.2. Overview of the Solver

$$a = \sqrt{\gamma RT}$$

To reduce round-off error and to improve the conditioning of the discretization, the flow properties are normalized. For this, the following reference quantities are used

$$\begin{aligned} \rho_{ref} &= \rho_{\infty}, & T_{ref} &= T_{\infty}, & P_{ref} &= \gamma P_{\infty} \\ u_{ref} &= v_{ref} = a_{\infty}, & R_{ref} &= R_{\infty} = \text{const} \end{aligned}$$

where $[\cdot]_{ref}$ is the reference value, and $[\cdot]_{\infty}$ is the far-field value for external aerodynamics problems or some other representative value for internal flows. The governing equations can then be expressed in terms of the non-dimensionalized quantities

$$\begin{aligned} \rho_n &= \frac{\rho}{\rho_{ref}}, & T_n &= \frac{T}{T_{ref}}, & P_n &= \frac{P}{P_{ref}} \\ u_n &= \frac{u}{u_{ref}}, & v_n &= \frac{v}{v_{ref}}, & R_n &= \frac{R}{R_{ref}} = 1 \end{aligned}$$

The important non-dimensionalized constitutive relations become

$$P_n = \frac{\rho_n T_n}{\gamma}, \quad a_n = \sqrt{T_n}, \quad E_n = \frac{P_n}{\gamma - 1} + \frac{1}{2} \rho_n (u_n^2 + v_n^2), \quad h_n = \frac{E_n + P_n}{\rho_n}$$

For the remainder of this thesis we will refer only to the normalized values and the subscript n is therefore dropped.

1.2.2 Vertex Centered Finite-Volume Discretization

The grids used in the present work are generated using a Delaunay refinement algorithm [10, 34]. In two dimensions these algorithms generate meshes composed of triangular cells. The solver used in the present work uses the median dual control volume formulation shown in Figure 1.1. Control volumes are associated with each vertex and are defined by connecting cell centroids with face centroids of the primal mesh. The dependent variables, which will be solved for, represent the control volume averaged values of the

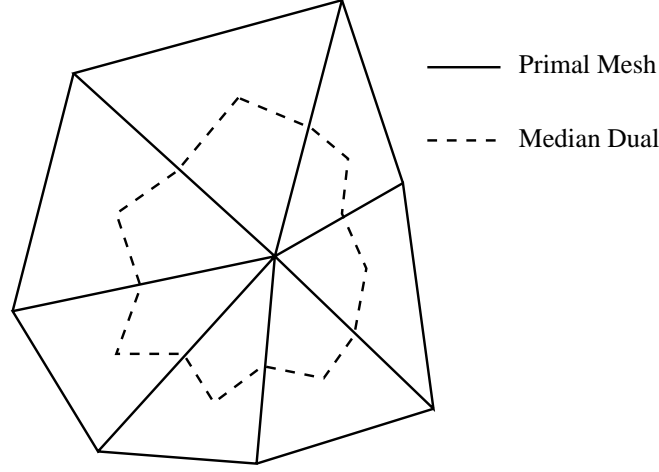


Figure 1.1: Median dual control volume

flow properties

$$\bar{U}_i = \frac{1}{A_i} \int_{V_i} U dA$$

1.2.3 Reconstruction

Reconstruction is a central component to the high-order accurate finite-volume method. Some of the details of reconstruction are therefore repeated to various extent in the manuscript chapters of this thesis. However, for completeness, the fundamental details of reconstruction for the inviscid scheme are summarized here. The extension to the viscous case is described in Chapter 4.

To obtain a high-order accurate numerical approximation to the residual we reconstruct a polynomial representation of the solution within each control volume. As is common, the reconstruction is carried out in the primitive variables $U = [\rho, u, v, P]^T$ rather than the conserved variables. As in previous work [29], the high-order accurate reconstruction is obtained by formulating and solving a least-squares problem.

To obtain an accurate estimate of the solution at flux quadrature points, we represent the solution within each control volume by the Taylor series

expansion

$$\begin{aligned}
 U_i^R(x - x_i, y - y_i) = & U|_i + \frac{\partial U}{\partial x}\bigg|_i (x - x_i) + \frac{\partial U}{\partial y}\bigg|_i (y - y_i) + \\
 & \frac{\partial^2 U}{\partial x^2}\bigg|_i \frac{(x - x_i)^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_i (x - x_i)(y - y_i) + \\
 & \frac{\partial^2 U}{\partial y^2}\bigg|_i \frac{(y - y_i)^2}{2} + \dots
 \end{aligned} \tag{1.3}$$

where U_i^R is the value of the reconstructed solution and $\frac{\partial^{k+l} U_i}{\partial x^k \partial y^l}$ are its derivatives at the reference point (x_i, y_i) of control volume i . The reference point typically used for vertex-centered schemes is the vertex location. The order of accuracy that the numerical method will achieve is, amongst other factors, limited by the truncation error of this reconstruction. Truncating this polynomial to degree k will result in a scheme which is at most $(k + 1)$ -order accurate. In the present work we consider reconstructions of up to cubic degree resulting in schemes up to fourth-order accurate. The coefficients of the polynomial are computed so that the mean value of the solution in the control volume is conserved and the reconstruction approximates nearby control volume averages. The neighboring control volumes used for reconstruction are selected based on their topological distance as shown in Figure 1.2. Entire topological layers are added until the number of neighbors equals or exceeds a set minimum. In the present work we use a minimum of 4 neighbors for linear reconstruction, 10 neighbors for quadratic reconstruction, and 16 neighbors for cubic reconstruction.

Conservation of the mean within a control volume is satisfied when

$$\begin{aligned}
 \bar{U}_i = \frac{1}{A_i} \int_{V_i} U_i^R dA \equiv & U|_i + \frac{\partial U}{\partial x}\bigg|_i \bar{x}_i + \frac{\partial U}{\partial y}\bigg|_i \bar{y}_i + \\
 & \frac{\partial^2 U}{\partial x^2}\bigg|_i \frac{\bar{x}_i^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_i \bar{x}_i \bar{y}_i + \\
 & \frac{\partial^2 U}{\partial y^2}\bigg|_i \frac{\bar{y}_i^2}{2} + \dots
 \end{aligned} \tag{1.4}$$

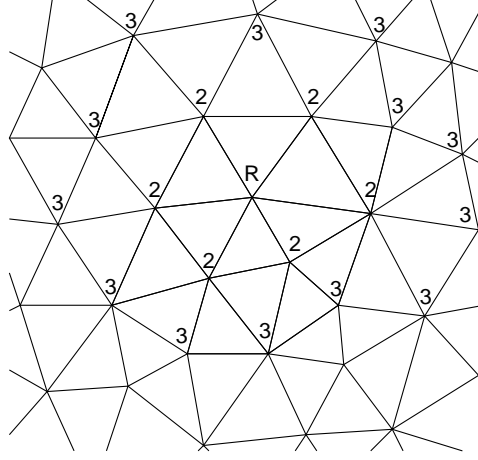


Figure 1.2: Example reconstruction stencil for second- and third-order scheme. In this example the fourth-order stencil would be identical to the third-order stencil.

where

$$\overline{x^n y^m}_i \equiv \frac{1}{A_i} \int_{V_i} (x - x_i)^n (y - y_i)^m dA$$

As we shall see, this mean constraint can be eliminated analytically from the least-squares system. The terms used to construct the reconstruction least-squares problem have the form

$$\begin{aligned} \frac{1}{A_j} \int_{V_j} U_i^R(\vec{x} - \vec{x}_i) dA &= U|_i + \frac{\partial U}{\partial x} \Big|_i \widehat{x}_{ij} + \frac{\partial U}{\partial y} \Big|_i \widehat{y}_{ij} \\ &+ \frac{\partial^2 U}{\partial x^2} \Big|_i \frac{\widehat{x}_{ij}^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_i \widehat{x}_{ij} \widehat{y}_{ij} + \frac{\partial^2 U}{\partial y^2} \Big|_i \frac{\widehat{y}_{ij}^2}{2} + \dots \end{aligned} \quad (1.5)$$

The geometric terms in this equation are of the general form

$$\begin{aligned} \widehat{x^n y^m}_{ij} &\equiv \frac{1}{A_j} \int_{V_j} ((x - x_j) + (x_j - x_i))^n \cdot ((y - y_j) + (y_j - y_i))^m dA \\ &= \sum_{l=0}^m \sum_{k=0}^n \frac{m!}{l! (m-l)!} \frac{n!}{k! (n-k)!} (x_j - x_i)^k \cdot (y_j - y_i)^l \cdot \overline{x^{n-k} y^{m-l}}_j \end{aligned}$$

The resulting least-squares problem to be solved for each solution variable in

1.2. Overview of the Solver

each control volume takes the form

$$\begin{aligned}
 & \begin{bmatrix} 1 & \bar{x}_i & \bar{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \cdots \\ w_{i1} & w_{i1}\widehat{x}_{i1} & w_{i1}\widehat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\ w_{i2} & w_{i2}\widehat{x}_{i2} & w_{i2}\widehat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\ w_{i3} & w_{i3}\widehat{x}_{i3} & w_{i3}\widehat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\widehat{x}_{iN} & w_{iN}\widehat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots \end{bmatrix} \begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i \\
 &= \begin{pmatrix} \bar{U}_i \\ w_{i1}\bar{U}_1 \\ w_{i2}\bar{U}_2 \\ w_{i3}\bar{U}_3 \\ \vdots \\ w_{iN}\bar{U}_N \end{pmatrix} \tag{1.6}
 \end{aligned}$$

where the weights can be used to emphasize geometrically nearby data

$$w_{ij} = \frac{1}{|\vec{x}_j - \vec{x}_i|^n} \tag{1.7}$$

where typically $n \in [0, 2]$. For isotropic or lightly anisotropic grids (such as those used for laminar flows) the choice of n does not impact the accuracy of the scheme but does affect its conditioning and stability. In the present work we use $n = 1$ for the inviscid cases and $n = 0$ for viscous cases. The first equation in the linear system is the mean constraint, which can be removed by Gauss elimination, leaving an unconstrained least-squares problem of reduced size. The efficient solution of these least-squares systems is discussed in Chapter 3. Once the polynomial coefficients are known, the reconstructed flow properties at any point in the control volume can easily be found using Equation 1.3.

1.2.4 Interior and Boundary Flux Evaluation

To maintain the numerical stability of the system, careful upwinding of the Euler system of equations is needed. The discretization of the flux follows the pioneering work of Godunov [17] who suggested solving what amounts to a shock tube or Riemann problem at the interface of control volumes. Since solving this Riemann problem exactly is computationally costly, a number of approximate Riemann solvers have been developed. In the present work, Roe's approximate Riemann solver [32] is used. The reconstructed solution from each of two control volumes incident on the Gauss point is used to define the two states of the Riemann solver.

The inviscid boundary conditions are enforced by using a special flux function at boundary Gauss points. In general this flux is simply taken as the analytic flux of Equation 1.2 computed using some information from the single reconstructed value from the interior and some information from the boundary condition. Specifically, for walls, we use the reconstructed value of ρ and P and a zero velocity normal to the wall. For inflows and outflows the boundary flux is constructed based on an eigenvalue analysis. Specifically, for supersonic inflows all values are taken from the boundary condition. For subsonic inflows the reconstructed value of pressure from the interior is used along with the total pressure, total temperature, and flow direction from the upstream condition. Analogously, for supersonic outflows the flux is based entirely on reconstructed values from the interior while for subsonic outflows pressure is defined by the boundary condition while the remaining flow properties are obtained from the reconstruction.

1.2.5 Flux Integration

As part of the numerical method, the contour integral in Equation 1.1 needs to be discretized. This is obtained through the use of Gauss quadrature. To achieve the desired order of accuracy, a single Gauss point per median dual edge is used for first- and second-order schemes and two Gauss points per edge are used for third- and fourth-order schemes. This represents, respectively, two and four points per edge of the primal mesh.

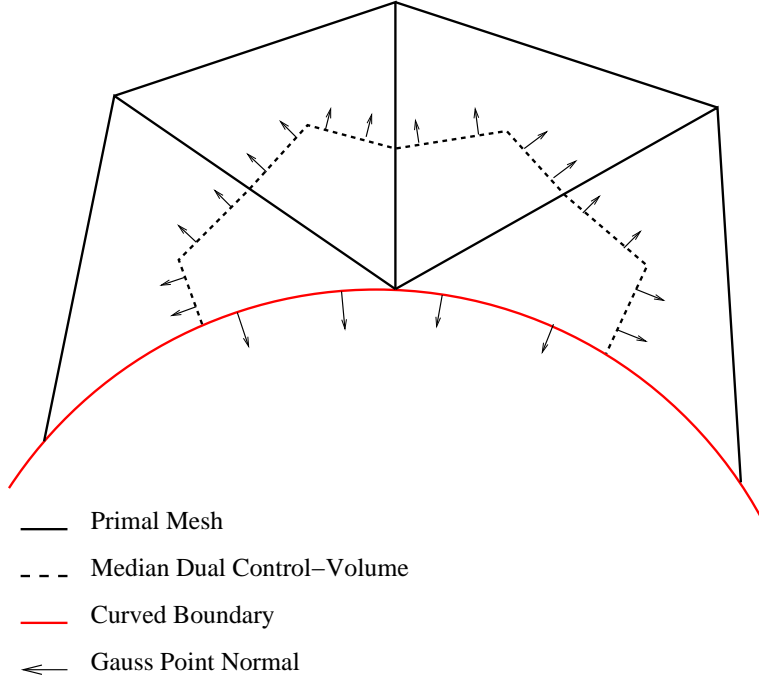


Figure 1.3: Approximate location and normal of Gauss integration points for a third- or fourth-order scheme on a boundary control volume

To achieve third- and high-order accurate schemes it is well known that Gauss points must be correctly located on curved boundaries [27, 29]. Recently it has also been shown that for vertex-centered schemes the curvature of the boundary must be accounted for when determining the Gauss point locations and the normals to the interior median dual edge incident on the boundary [14]. For this reason, the Gauss point locations, normals and weights are all computed from the curved boundary information as shown in Figure 1.3.

In the present work we are concerned with steady-state flows. In these problems, the time derivative term in Equation 1.1 is eliminated and the remaining task is to find the root of the system of equations defining the discrete flux integrals. The novel approach taken in this work, and how it compares to more commonly used methods, is discussed in Chapter 3.

1.3 Objectives and Outline

The objective of this thesis is to demonstrate that carefully implemented high-order accurate unstructured finite-volume methods for computational aerodynamics can be more computationally efficient than the second-order methods commonly used. This broad objective is attained by considering a number of specific, previously unresolved issues. Although ultimately this study is aimed at resolving issues with the simulation of three-dimensional turbulent flows, the current study is carried out in two-dimensions. Furthermore, only inviscid and laminar viscous flows are considered. We use these simplifications since certain aspects of implementing the high-order method for turbulent three-dimensional flows have not yet been addressed. These challenges will be discussed in the concluding chapter. The core chapters of this thesis are composed of three manuscripts submitted for journal publication. Further motivation and relevant background information is therefore addressed in each individual chapter.

Many aerodynamic flows of engineering interest occur in the transonic or supersonic regimes. Chapter 2 addresses the previously unresolved issue of how to eliminate unwanted oscillations near solution discontinuities such as shocks while maintaining the accuracy of the high-order method in smooth regions. The chapter studies the extension of limiters used for second-order methods to the high-order case. The requirements for accuracy and efficient convergence are discussed. A new limiting procedure is proposed. Ringleb's flow problem is used to demonstrate that nearly nominal orders of accuracy for schemes up to fourth-order can be achieved in smooth regions using the new limiter. The results for the fourth-order accurate solution of transonic flow demonstrates good convergence properties and significant qualitative improvement of the solution relative to the second-order method. It is also demonstrated that the new limiter can reduce the dissipation of second-order schemes with minimal sacrifices in convergence properties relative to existing approaches.

Although the matrix-free Newton-GMRES method commonly used to converge the solution of the second-order scheme to steady-state has been

shown to be relatively effective for the third-order scheme, it has also been shown to be less effective for the fourth-order scheme [25, 26]. Chapter 3 successfully addresses the poor convergence to steady state of the fourth-order scheme through the use of novel Newton globalization techniques and the use of the exact flux Jacobian with a matrix-explicit GMRES method. We show that robustness of the globalization can be improved by supplementing the pseudo-timestepping method commonly used with a line search method. The number of timesteps required for convergence can be reduced by using a timestep that scales with the local residual. We also show that it is possible to form the high-order Jacobian explicitly at a reasonable computational cost. This is demonstrated for cases using both limited and unlimited reconstruction. This exact Jacobian can be used for preconditioning and directly in the GMRES method. The benefits resulting from improvements in preconditioning and the elimination of residual evaluations in the inner iterations of the matrix-free GMRES method are substantial. Computational results focus on second- and fourth-order accurate schemes with some results for the third-order scheme. Overall computational cost for the matrix-explicit method is lower than the matrix-free method for all cases. The fourth-order matrix-explicit scheme is a factor of 1.6 to 3 faster than the matrix-free scheme while requiring about 50-100% more memory.

In Chapter 4, the methods successfully applied for the high-order solution of inviscid flow are extended to laminar viscous flow. A means of enforcing the viscous boundary conditions by introducing additional rows in the reconstruction matrices is devised. The solution gradients needed for the viscous fluxes are obtained from the same reconstruction polynomials used for the inviscid flux. Rapid convergence of steady-state problems is made possible by computing the exact flux Jacobian and the use of a Newton-GMRES algorithm. A grid convergence study of cylindrical Couette flow is used to demonstrate the order of accuracy of the scheme. Laminar flow over an airfoil on a sequence of anisotropic grids is also used to qualitatively demonstrate the superior accuracy of the fourth-order scheme.

The conclusion in Chapter 5 discusses the combined significance of the methods and results presented in the manuscript chapters. A number of ex-

1.3. Objectives and Outline

tensions to the present work are suggested. In particular the extensions of the methods to three dimensions and turbulent flows are discussed. The expected difference in the computational efficiency trade-offs for three-dimensional methods relative to their two-dimensional counterparts are addressed. Specific issues of high-order methods applied to the highly anisotropic grids needed to solve high Reynolds number flows are discussed. The need for a stable high-order accurate discretization of turbulence model is also addressed.

1.4 Bibliography

- [1] F. Angrand and A. Dervieux. Some explicit triangular finite element schemes for the Euler equations. *International Journal for Numerical Methods in Fluids*, 4:749–764, Aug. 1984.
- [2] H. E. Bailey and R. M. Beam. Newton’s method applied to finite difference approximations for steady state compressible Navier-Stokes equations. *Journal of Computational Physics*, 93:108–127, 1991.
- [3] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *Unstructured Grid Methods for Advection-Dominated Flows*, pages 6–1 – 6–61. AGARD, Neuilly sur Seine, France, 1992. AGARD-R-787.
- [4] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA paper 93-0668, Jan. 1993.
- [5] T. J. Barth and P. O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, Jan. 1990.
- [6] T. J. Barth and S. W. Linton. An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. AIAA paper 95-0221, Jan. 1995.
- [7] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-\omega$ turbulence model equations. *Computers and Fluids*, 34(4-5):507 – 540, 2005. Residual Distribution Schemes, Discontinuous Galerkin Schemes and Adaptation.
- [8] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.

1.4. Bibliography

- [9] M. Blanco and D. W. Zingg. Fast Newton-Krylov method for unstructured grids. *American Institute of Aeronautics and Astronautics Journal*, 36(4):607–612, 1998.
- [10] C. Boivin and C. F. Ollivier-Gooch. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *International Journal for Numerical Methods in Engineering*, 55(10):1185–1213, Dec. 2002.
- [11] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*, volume 960 of *Lecture Notes in Mathematics*. Springer-Verlag, 1982.
- [12] S. De Rango and D. W. Zingg. Higher-order spatial discretization for turbulent aerodynamic computations. *American Institute of Aeronautics and Astronautics Journal*, 39(7):1296–1304, July 2001.
- [13] M. Delanaye and J. A. Essers. Quadratic-reconstruction finite volume scheme for compressible flows on unstructured adaptive grids. *American Institute of Aeronautics and Astronautics Journal*, 35(4):631–639, Apr. 1997.
- [14] B. Diskin and J. Thomas. Accuracy analysis for mixed-element finite-volume discretization schemes. Technical Report 2007-08, National Institute of Aerospace, 2007.
- [15] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 207(1):92 – 113, 2005.
- [16] P. Geuzaine, M. Delanaye, and J.-A. Essers. Computation of high Reynolds number flows with an implicit quadratic reconstruction scheme on unstructured grids. In *Proceedings of the Thirteenth AIAA Computational Fluid Dynamics Conference*, pages 610–619. American Institute of Aeronautics and Astronautics, 1997.

- [17] S. K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskie Sborniki*, 47:357–393, 1959.
- [18] M. J. Hemsch. Statistical analysis of CFD solutions from the Drag Prediction Workshop. In *Fortieth AIAA Aerospace Sciences Meeting*, Jan. 2002. AIAA paper 2002-842.
- [19] A. Jameson. Successes and challenges in computational aerodynamics. In *Proceedings of the Eighth AIAA Computational Fluid Dynamics Conference*, June 1987. AIAA paper 87-1184.
- [20] A. Jameson. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence in transonic and hypersonic flows. AIAA paper 93-3359-CP, July 1993.
- [21] A. Jameson and D. Mavriplis. Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *American Institute of Aeronautics and Astronautics Journal*, 24(4):611–618, 1986.
- [22] R. Loehner, K. Morgan, J. Peraire, and O. C. Zienkiewicz. Finite element methods for high speed flows. In *Proceedings of the Eighth AIAA Computational Fluid Dynamics Conference*, pages 403–410, 1985. AIAA paper 1985-1531.
- [23] D. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *American Institute of Aeronautics and Astronautics Journal*, 28(2):213–221, Feb. 1990.
- [24] D. Mavriplis. Directional agglomeration multigrid techniques for high reynolds number viscous flow solvers. Thirty-sixth AIAA Aerospace Sciences Meeting, Jan. 1998.
- [25] A. Nejat and C. Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. *Journal of Computational Physics*, 227(4):2366–2386, 2008.

1.4. Bibliography

- [26] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2592–2609, 2008.
- [27] C. Ollivier-Gooch, A. Nejat, and C. Michalak. On obtaining and verifying high-order finite-volume solutions to the Euler equations on unstructured meshes. *American Institute of Aeronautics and Astronautics Journal*, Submitted, 2008.
- [28] C. F. Ollivier-Gooch. High-order ENO schemes for unstructured meshes based on least-squares reconstruction. AIAA paper 97-0540, Jan. 1997.
- [29] C. F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.
- [30] N. Pierce, M. Giles, A. Jameson, and L. Martinelli. Accelerating three-dimensional Navier-Stokes calculations. In *Proceedings of the Thirteenth AIAA Computational Fluid Dynamics Conference*, pages 676–698. American Institute of Aeronautics and Astronautics, July 1997. AIAA 97-1953-CP.
- [31] A. Pueyo and D. W. Zingg. Efficient Newton-Krylov solver for aerodynamic computations. *American Institute of Aeronautics and Astronautics Journal*, 36(11):1991–1997, 1998.
- [32] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [33] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3):856–869, July 1986.
- [34] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1–3):21–74, May 2002.

1.4. Bibliography

- [35] P. R. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow*, 21(3):252 – 263, 2000.
- [36] V. Venkatakrishnan. Newton solution of inviscid and viscous problems. Twenty-sixth AIAA Aerospace Sciences Meeting, 1988. AIAA Paper 88-0143.
- [37] V. Venkatakrishnan and T. J. Barth. Application of direct solvers to unstructured meshes for the Euler and Navier-Stokes equations using upwind schemes. Twenty-seventh AIAA Aerospace Sciences Meeting, 1989. AIAA Paper 89-0364.
- [38] D. Vigneron, J.-M. Vaassen, and J.-A. Essers. An implicit finite volume method for the solution of 3D low Mach number viscous flows using a local preconditioning technique. *Journal of Computational and Applied Mathematics*, 215(2):610 – 617, 2008.
- [39] P. Wong and D. W. Zingg. Three-dimensional aerodynamic computations on unstructured grids using a Newton-Krylov approach. *Computers and Fluids*, 37:107–120, 2008.

Chapter 2

Monotonicity Enforcement Using a Limiter[†]

2.1 Introduction

High-order discretizations have been shown to reduce computational effort on structured grids [6, 22]. High-order finite-volume methods on unstructured grids, although well known [3, 4, 7], have not yet effectively been applied to large scale aerodynamics problems. An outstanding issue with these methods is how to deal with discontinuities, such as shocks in the flow, while maintaining good accuracy and convergence.

One means of dealing with discontinuities is to use the classic MUSCL [20] scheme with the addition of a slope limiter. For the second-order case, Barth and Jespersen [5] demonstrated the use of limited reconstruction for the solution of the Euler equations. Efficient convergence to steady state was achieved by Venkatakrishnan [21] by modifying the limiter to be differentiable.

Third-order accurate schemes using k -exact reconstruction have been demonstrated first by Barth and Frederickson [4] and subsequently by others [2, 7, 9, 13, 14, for example]. Transonic and supersonic solutions have been computed in some of these works by various extensions of second-order limiters. However, the work of Barth [2] presents a limiting approach which causes difficulties in steady-state convergence, while other works [7, 13] present approaches that do not strictly enforce monotonicity and therefore allow some undesirable oscillations to occur. Furthermore none of these works

[†]A version of this chapter will be submitted for publication. C. Michalak and C. Ollivier-Gooch. Accuracy preserving limiter for the high-order accurate solution of the Euler equations.

formally demonstrate that high-order accuracy is maintained in smooth regions of the flow.

An alternative to MUSCL for obtaining high-order accurate solutions is the essentially non-oscillatory (ENO) scheme [1, 10, 18, for instance]. These methods avoid the need for slope limiters by selecting a smooth flux stencil at each iteration. Due to the inherent non-differentiability of this process, convergence of the solution to steady state is not possible. Weighted ENO (WENO) schemes [8, 11, 12, for instance] were introduced in part to resolve this issue. However these schemes do not converge to steady state as efficiently as MUSCL schemes. The computational cost per residual evaluation is also much higher than for reconstruction based solvers. A hybrid between WENO and MUSCL schemes named Quasi-ENO [15] has similar limitations. This method's reconstruction step is much more expensive than traditional MUSCL since the reconstruction least-squares matrix changes at each iteration. For these reasons ENO-like schemes have not seen widespread application to aerodynamics problems.

The present work formulates the requirements and presents a candidate for a limiter that achieves fourth-order accurate solutions in smooth regions while maintaining monotonicity and good convergence properties. An overview of the high-order MUSCL scheme is given in Section 2.2. Second-order limiters are reviewed in Section 2.3. Our extension of these methods to high-order schemes is presented in Section 2.4. Finally, we present results in Section 2.5 that demonstrate that our method achieves nominal order of accuracy in smooth regions of the flow while effectively eliminating oscillations at shocks and maintaining good convergence properties.

2.2 High-Order Accurate Solution Reconstruction

The third- and fourth-order accurate reconstruction procedure we use here is documented by Ollivier-Gooch and Van Altena [16] and is briefly reviewed in this section. Only the equations that are needed for the discussion of limiters

2.2. High-Order Accurate Solution Reconstruction

are presented.

In the finite-volume method, the domain is tessellated into non-overlapping control volumes. Each control volume V_i has a geometric reference point \vec{x}_i . While in principle any point can be chosen as the reference point, the usual choices (which we recommend) are the cell centroid for cell-centered control volumes and the vertex for vertex-centered control volumes. For any smooth function $U(\vec{x})$ and its control volume averaged values \bar{U}_i , the k -exact least-squares reconstruction will use a compact stencil in the neighborhood of control volume i to compute an expansion $U_i^R(\vec{x} - \vec{x}_i)$ that conserves the mean in control volume i and reconstructs exactly polynomials of degree $\leq k$ (equivalently, $U_i^R(\vec{x} - \vec{x}_i) - U(\vec{x}) = \mathcal{O}(\Delta x^{k+1})$).

Conservation of the mean requires that the average of the reconstructed function U_i^R and the original function U over control volume i be the same:

$$\bar{U}_i \equiv \frac{1}{V_i} \int_{V_i} U_i^R(\vec{x} - \vec{x}_i) dA = \frac{1}{V_i} \int_{V_i} U(\vec{x}) dA \quad (2.1)$$

The expansion $U_i^R(\vec{x} - \vec{x}_i)$ can be written as:

$$\begin{aligned} U_i^R(\vec{x} - \vec{x}_i) &= U|_{\vec{x}_i} + \frac{\partial U}{\partial x} \Big|_{\vec{x}_i} (x - x_i) + \frac{\partial U}{\partial y} \Big|_{\vec{x}_i} (y - y_i) \\ &\quad + \frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} \frac{(x - x_i)^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} ((x - x_i)(y - y_i)) \\ &\quad + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} \frac{(y - y_i)^2}{2} + \dots \end{aligned} \quad (2.2)$$

Taking the control volume average of this expansion over control volume i and equating it to the mean value gives

$$\begin{aligned} \bar{U}_i &= U|_{\vec{x}_i} + \frac{\partial U}{\partial x} \Big|_{\vec{x}_i} \bar{x}_i + \frac{\partial U}{\partial y} \Big|_{\vec{x}_i} \bar{y}_i \\ &\quad + \frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} \frac{\bar{x}^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} \bar{xy} + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} \frac{\bar{y}^2}{2} + \dots \end{aligned} \quad (2.3)$$

where

$$\overline{x^n y^m}_i \equiv \frac{1}{A_i} \int_{V_i} (x - x_i)^n (y - y_i)^m dA. \quad (2.4)$$

are control volume moments. This condition, which must be satisfied exactly, is combined with the reconstruction goal of approximating nearby control volume averages to obtain a constrained least-squares problem for the solution of the Taylor series expansion coefficients. Since the resulting least-squares matrix depends only on geometric terms, its pseudoinverse may be found in a preprocessing step. Therefore the reconstruction step at each flux evaluation is reduced to a matrix-vector product and the exact flux Jacobian can be computed as described in the manuscript reproduced in Chapter 3. Ollivier-Gooch [15] presents a modification to the reconstruction procedure resulting in a quasi-ENO scheme. This scheme eliminates the requirement for a limiter by varying the weights of the rows in the least-squares matrix at each iterations based on a measure of smoothness. However, since the pseudoinverse can no longer be precomputed, this scheme is computationally expensive.

2.3 Second-Order Limiting

To avoid introducing oscillation in the solution process, no new local extrema must be formed during reconstruction. Barth and Jespersen [5] introduced the first limiter for unstructured grids. The scheme consists of finding a limiter value Φ_i for each primitive flow variable in each control volume that will limit the gradient in the piecewise-linear reconstruction of the solution. In the second-order reconstruction case, if the reference location \vec{x}_i is taken to be the control volume centroid, the point-wise value $U|_{\vec{x}_i}$ is equal to the control volume average \bar{U}_i . This leads to the limited reconstruction of the form

$$U_i^R(\vec{x} - \vec{x}_i, \Phi_i) = \bar{U}_i + \Phi_i \nabla U_i \cdot (\vec{x} - \vec{x}_i), \quad \Phi \in [0, 1]$$

The goal is to find the largest Φ_i which prevents the formation of local extrema at the flux integration Gauss points. The following procedure is used by Barth and Jespersen :

2.3. Second-Order Limiting

1. Find the largest negative ($\delta U_i^{min} = \min(\bar{U}_j - \bar{U}_i)$) and positive ($\delta U_i^{max} = \max(\bar{U}_j - \bar{U}_i)$) difference between the solution in the immediate neighbors j and the current control volume i .
2. Compute the unconstrained reconstructed value at each Gauss point ($U_{ik} = U_i^R(\vec{x}_k - \vec{x}_i)$).
3. Compute a maximum allowable value of Φ_{ik} for each Gauss point k .

$$\Phi_{ik} = \begin{cases} \min\left(1, \frac{\delta U_i^{max}}{U_{ik} - \bar{U}_i}\right), & \text{if } U_{ik} - \bar{U}_i > 0 \\ \min\left(1, \frac{\delta U_i^{min}}{U_{ik} - \bar{U}_i}\right), & \text{if } U_{ik} - \bar{U}_i < 0 \\ 1, & \text{if } U_{ik} - \bar{U}_i = 0 \end{cases}$$

4. Select $\Phi_i = \min(\Phi_{ik})$.
5. Compute the limited reconstruction $U_i^R(\vec{x} - \vec{x}_i, \Phi_i)$ at flux Gauss integration points.

Clearly, steps 1, 3, and 4 introduce non-differentiability in the computation of the reconstructed function. Consequently, the second-order flux is also non-differentiable. This has severe adverse effect on the convergence properties of the solver. This is particularly evident for implicit schemes, but even explicit time advance schemes are unable to obtain more than two or three orders of magnitude in residual reduction.

In practice, the non-differentiability of step 3 causes the greatest degradation in convergence performance. For this reason, Venkatakrishnan [21] introduces a smooth alternative to step 3 of the Barth-Jespersen procedure by replacing the function $\min(1, y)$ with

$$\phi(y) = \frac{y^2 + 2y}{y^2 + y + 2} \quad (2.5)$$

The effect of this modification can be seen in Figure 2.1. This function is differentiable and is entirely contained within the monotonicity bounds of the original limiter. Furthermore by satisfying the condition $\phi(2) = 1$ this limiter can be shown to preserve second-order accuracy in regions where

2.3. Second-Order Limiting

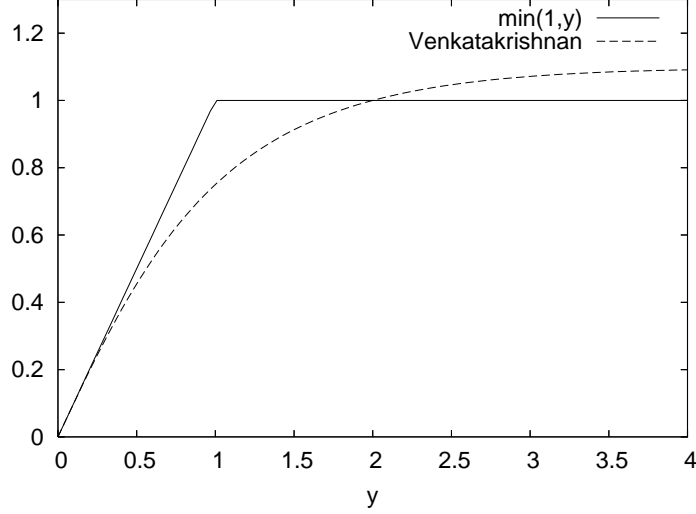


Figure 2.1: Venkatakrishnan's smooth approximation to $\min(1, y)$

no extrema exist for perfectly uniform meshes. Specifically for smooth solutions on a uniform mesh, the value of the Gauss point is expected to be located at the midpoint connecting two control volume centroids. Therefore for any smooth function $\frac{\delta U_i^{max}}{U_{ik} - \bar{U}_i}$ is expected to be $2 \pm \mathcal{O}(x_i - x_{ik})$, and $\phi(y)$ is therefore expected to be $1 \pm \mathcal{O}(x_i - x_{ik})$. Hence, modifying the gradient in this manner for second-order solutions introduces an error in the reconstruction which is on the order of truncation for smooth flows on uniform grids. However, for general unstructured grids the Gauss point can be located at a distance $\mathcal{O}(x_i - x_{ik})$ away from the midpoint between centroids. Therefore, as will be seen in the results, this limiter leads to accuracy loss relative to the unlimited scheme.

A further modification introduced by Venkatakrishnan is a method to avoid applying the limiter in regions of nearly uniform flow and smooth extrema. In these regions we expect the solution to vary such that $U_{ik} - \bar{U}_i = \mathcal{O}(\Delta x^2)$ where Δx is the characteristic length of the control volume i . Therefore if the effect of the limiter can be eliminated when $U_{ik} - \bar{U}_i \leq (K\Delta x)^{1.5}$, where K is a tunable parameter, accuracy in smooth extrema regions can be improved without compromising monotonicity enforcement in

2.4. High-Order Limiting

other regions.

Venkatakrisnan uses a method inspired by van Albada [19] to make the switch between limited and unlimited regions of flow using a differentiable function so that convergence properties are not adversely affected. For the case $U_{ik} - \bar{U}_i > 0$ the limiter becomes

$$\phi_{ik} = \frac{1}{\Delta_-} \left[\frac{(\Delta_+^2 + \epsilon^2)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_- \Delta_+ + \epsilon^2} \right] \quad (2.6)$$

In this equation $\Delta_- = U_{ik} - \bar{U}_i$, $\Delta_+ = \delta U_i^{max}$ and $\epsilon^2 = (K\Delta x)^3$. In addition to improving accuracy this modification is known to be critical to achieving good convergence. When the solution is perturbed in nearly uniform regions or near smooth extrema the results of steps 1 and 4 of the limiting procedure can be expected to change more frequently than in non-uniform regions. Therefore the non-differentiability of these steps is a much greater hindrance to convergence in uniform regions than in non-uniform regions. By effectively disabling the limiter in uniform regions, the addition of the ϵ term in Equation 2.6 greatly improves convergence. The choice of the parameter K is a compromise. Large values of K are favorable to accuracy in smooth regions and good convergence. However, since for any $K > 0$ the limiter no longer strictly enforces monotonicity, large values of K can lead to significant overshoots near discontinuities in the solution.

2.4 High-Order Limiting

2.4.1 Monotonicity

The first challenge of extending the limiting procedure to third- and fourth-order accurate schemes is to express the monotonicity requirement including the high-order reconstruction terms. In second-order schemes the assumption that the solution at the reference point is equal to the control volume average is often made. For cell-centered schemes, where the reference point is the centroid, this assumption is correct. For vertex-centered scheme, the reference point is usually chosen as the vertex location, therefore the assumption is

2.4. High-Order Limiting

not strictly correct. However, for third- and higher-order schemes the control volume average solution is in general not equal to the centroidal value of the reconstruction. Therefore, when devising a limiter for these schemes, making the distinction between control volume average values and reference-point values becomes critical for maintaining high-order accuracy. With this in mind, the reconstruction polynomial in Equation 2.2 can be rewritten in terms of the control volume average with the help of equation 2.3 to yield:

$$\begin{aligned}
 U_i^R(\vec{x} - \vec{x}_i) = & \bar{U}_i + \left(\frac{\partial U}{\partial x} \Big|_{\vec{x}_i} ((x - x_i) - \bar{x}_i) + \frac{\partial U}{\partial y} \Big|_{\vec{x}_i} ((y - y_i) - \bar{y}_i) \right) \\
 & + \left(\frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} \left(\frac{(x - x_i)^2}{2} - \frac{\bar{x}^2}{2} \right) + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} ((x - x_i)(y - y_i) - \bar{x}\bar{y}) \right. \\
 & \left. + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} \left(\frac{(y - y_i)^2}{2} - \frac{\bar{y}^2}{2} \right) + \dots \right) \quad (2.7)
 \end{aligned}$$

This can be interpreted as meaning that the reconstructed solution at any point is the control volume average plus second and high-order contributions from the reconstruction :

$$U_i^R(\vec{x} - \vec{x}_i) = \bar{U}_i + S(\vec{x} - \vec{x}_i) + H(\vec{x} - \vec{x}_i)$$

Therefore, analogous to the second-order case, no new extrema will be formed if

$$\delta U_i^{min} \leq S(\vec{x} - \vec{x}_i) + H(\vec{x} - \vec{x}_i) \leq \delta U_i^{max}$$

As in the work of Barth[2], the limited form of the high-order accurate reconstruction can be expressed as

$$U_i^R(\vec{x} - \vec{x}_i, \Phi_i) = \bar{U}_i + \Phi_i \left(S(\vec{x} - \vec{x}_i) + H_i(\vec{x} - \vec{x}_i) \right) \quad (2.8)$$

Given this formulation, the same limiting procedure used in Section 2.3 can be applied to the high-order reconstruction.

Some previous works [7, 13] have suggested a formulation where the

2.4. High-Order Limiting

limiter value multiplies only the second-order terms while the high-order terms are “switched off” when discontinuities are detected. This formulation has the form of

$$U_i^R(\vec{x} - \vec{x}_i, \Phi_i, \sigma_i) = \bar{U}_i + (\Phi_i(1 - \sigma_i) + \sigma_i) S(\vec{x} - \vec{x}_i) + \sigma_i H(\vec{x} - \vec{x}_i)$$

where σ_i , the discontinuity detector, is zero near discontinuities and one in smooth regions of the flow. However, this approach may violate the monotonicity requirement as the high-order terms may have contributed to reducing the overshoot in the unlimited reconstruction used in determining the value of Φ_i . Therefore, the value of Φ_i computed may be insufficient to reduce the slope such that overshoots occur when the high-order reconstruction terms are disabled. The additional parameters introduced in a smooth switching function for σ_i can be considered a further disadvantage.

2.4.2 Accuracy

As previously mentioned, on uniform grids a limiter for the second-order scheme maintains nominal accuracy as long as $|\phi - 1| \leq \mathcal{O}(\Delta x)$ since this results in an error that is on the order of the quadratic term in the Taylor expansion. However, when a third- or fourth-order scheme is used, the limiter must satisfy $|\phi - 1| \leq \mathcal{O}(\Delta x^2)$ or $|\phi - 1| \leq \mathcal{O}(\Delta x^3)$, respectively, for the effect of the limiter in smooth regions to be on the order of truncation error.

For this reason, Venkatakrishnan’s limiter will not provide sufficient accuracy even in smooth regions without any local extrema. While the Barth-Jespersen limiter does satisfy these conditions, the lack of differentiability will make achieving a steady-state solution difficult. Therefore, we seek a new approximation for $\min(1, y)$ used in step 3 of the limiting procedure, which we will call $\widetilde{\min}(1, y)$. Like Venkatakrishnan’s function, given in Equation 2.5, we require that it be differentiable at all points and that it be entirely contained under the function $\min(1, y)$. However, unlike Equation 2.5, we also require this new limiting function to have a value of exactly 1 for a range of values $y \geq y_t$ where $1 < y_t < 2$ represents a threshold value. For

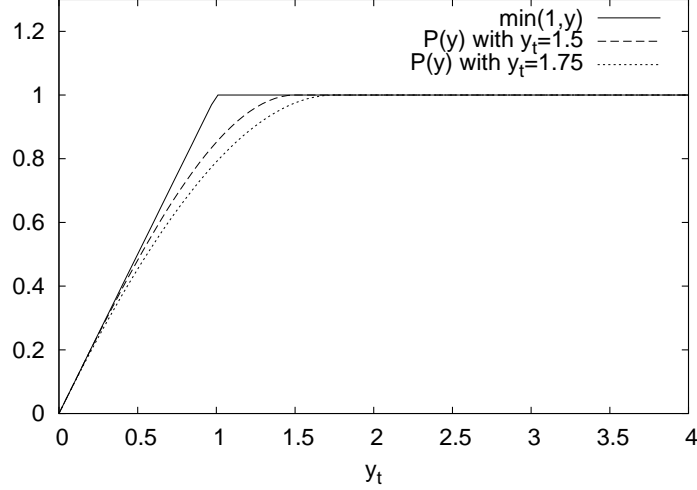


Figure 2.2: $\widetilde{\min}(1, y)$ with $y_t = 1.5$ and $y_t = 1.75$ compared to $\min(1, y)$

this function, we propose the form

$$\widetilde{\min}(1, y) = \begin{cases} P(y) & y < y_t \\ 1 & y \geq y_t \end{cases}$$

where $P(y)$ is a polynomial satisfying

$$\begin{aligned} P|_0 &= 0 & P|_{y_t} &= 1 \\ \frac{dP}{dy}|_0 &= 1 & \frac{dP}{dy}|_{y_t} &= 0 \\ P(y) &\leq \min(1, y), \quad y \in [0, y_t] \end{aligned}$$

The resulting polynomials for $y_t = 1.5, 1.75$ are plotted in Figure 2.2. This function allows the preservation of high-order accuracy on uniform grids by satisfying $|\widetilde{\min}(1, y) - 1| \leq \mathcal{O}(\Delta x^3)$. Additionally, this function is also effective in maintaining high-order accuracy in regions of mild mesh non-uniformity. The degree of non-uniformity that can be accommodated is dictated by the choice of the threshold value y_t . Smaller values of y_t are less likely to unduly activate the limiter on non-uniform meshes but result in a limiter that approaches non-differentiability. Therefore the the choice of y_t

2.4. High-Order Limiting

is a compromise between maintaining good accuracy on non-uniform grids and maintaining good convergence properties. For the results presented in this work, we use $y_t = 1.5$ which yields the following cubic polynomial

$$P(y) = -\frac{4}{27}y^3 + y$$

2.4.3 Uniform Regions and Smooth Extrema

For the reasons already mentioned in Section 2.3, it is desirable to eliminate the effect of the limiter in regions of uniform flow or near smooth extrema. To maintain high-order accuracy it is essential to permit monotonicity to be violated near smooth extrema. Furthermore the limiter value changes rapidly and is non-differentiable in uniform regions which prevents good convergence of the solver. Therefore, analogous to the second-order limiter of Venkatakrishnan, we wish to eliminate the effect of the limiter when the local solution variation is $\mathcal{O}(\Delta x^2)$ or smaller. Specifically, we propose to disable the limiter when

$$\delta U \equiv (\delta U_i^{max} - \delta U_i^{min}) < (K\Delta x)^{\frac{3}{2}}$$

where K is a tunable parameter. However using a simple switch would introduce a non-differentiable step in the residual evaluation which would cause convergence problems. Therefore, to maintain differentiability, the following procedure is proposed:

$$\tilde{\Phi}_i = \tilde{\sigma}_i + (1 - \tilde{\sigma}_i)\Phi_i \tag{2.9}$$

where Φ_i is the limiter value as calculated in step 4 of the procedure in Section 2.3 and $\tilde{\sigma}_i$ is the following function:

$$\tilde{\sigma}_i = \begin{cases} 1 & \delta U^2 \leq (K\Delta x)^3 \\ s\left(\frac{\delta U^2 - (K\Delta x)^3}{(K\Delta x)^3}\right) & (K\Delta x)^3 < \delta U^2 < 2(K\Delta x)^3 \\ 0 & \delta U^2 \geq 2(K\Delta x)^3 \end{cases} \tag{2.10}$$

where the transition function s is defined by

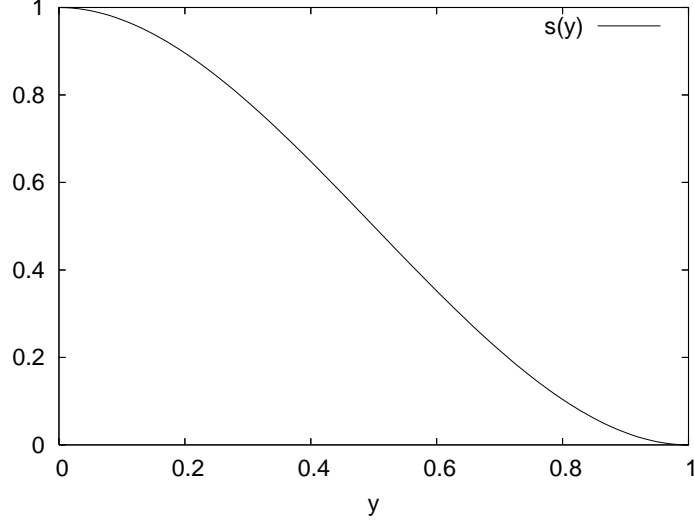


Figure 2.3: Transition function $s(y) = 2y^3 - 3y^2 + 1$ used to smoothly disable the limiter in nearly uniform regions

$$s(y) = 2y^3 - 3y^2 + 1 \quad (2.11)$$

and is plotted in Figure 2.3. The limited reconstruction is then computed for each Gauss point by evaluating $U_i^R(\vec{x} - \vec{x}_i, \tilde{\Phi}_i)$.

Although this two stage limiting procedure is somewhat more computationally expensive than Venkatakrishnan’s limiter in the general case, some “short circuiting” is possible in uniform regions of flow. Since $\tilde{\sigma}_i$ depends only on neighboring control volume averages, unlike Φ_i which also depends on an evaluation of the unconstrained reconstruction at each Gauss point, it is relatively inexpensive to compute. When $\tilde{\sigma}_i$ evaluates to 1, computational effort can be saved by not computing Φ_i since it does not affect the value of the final limiter $\tilde{\Phi}_i$.

2.4.4 Boundary Treatment

Maintaining high-order accuracy near domain boundaries represents a special challenge. Local extrema are expected to exist on the boundary for smooth

2.4. High-Order Limiting

flows. Unlike smooth extrema in the interior of the domain, these points are not characterized by a zero first derivative of flow property with respect to space. Therefore, the method used in Subsection 2.4.3 will not be effective in disabling the limiter in these regions.

As a first measure, in our implementation of both Venkatakrishnan and the new limiter we elect to only iterate over interior face Gauss points in Step 3 of the limiting procedure. Therefore an extremum forming on a boundary Gauss point will not cause the limiter to activate. In our experience this does not cause any oscillatory issues at shocks. However, the Gauss points on interior faces of boundary control volumes will often also have reconstructed values of smooth solutions that form an extremum relative to the control volume averages of the solution in the reconstruction stencil. In some cases even Gauss points of control volumes adjacent to boundary control volumes will exhibit this behavior. In our experience eliminating all of these Gauss points from Step 3 of the limiting procedure causes unacceptable oscillations in the solution near shocks. Therefore, to maintain high-order accuracy near boundaries while maintaining solution monotonicity another approach is needed.

Thus far the proposed limiting procedure is entirely physics agnostic. However we have been unable to find a satisfactory means of dealing with the issue of accuracy and monotonicity of boundary and near-boundary control volumes in this manner. Therefore we propose a method specific to the Euler equations.

For external and many internal flows the far-field boundary usually has nearly uniform flow conditions. For these boundaries the limiter is effectively disabled by the method applied in Subsection 2.4.3. Therefore we will focus our attention on the wall boundary. Two flow conditions can be present at a wall boundary condition: the flow can be tangential to the wall, or it can be a stagnation point. We propose methods for maintaining high-order accuracy in both cases.

2.4.4.1 Tangential Flow

For every wall boundary control volume we will consider a “ghost” control volume which is a mirror image of the boundary control volume about the boundary. It will have an approximate control volume average value of the solution assigned to it consistent with a shockless flow. These values will be used to expand the stencil used to determine monotonicity outside of the flow domain. This will effectively make the reconstructed Gauss point values no longer extrema for smooth flows. To preserve high-order accuracy, these ghost control volume values will not be used in the least-squares reconstruction process. They will only be used when determining the values of δU_{min} and δU_{max} in the limiting procedure. For any control volume that includes the boundary control volume as its first neighbor the respective mirror control volume will also be included in determining δU_{min} and δU_{max} .

To determine an appropriate solution value for the mirror control volume we begin by noting that any well resolved curved boundary locally resembles a circular arc. We approximate the radius of curvature from the boundary Gauss point normals of the curved boundary edges. Since this information is needed by the high-order boundary flux integration scheme, it is readily available. Next, we make the assumption of steady momentum in the direction normal to the streamline

$$\frac{\partial P}{\partial n} = -\frac{\rho V^2}{R}$$

where n is the direction normal to the wall, ρ is density, V is velocity and R the radius. Since the ghost value is only used in computing the limiter, a first-order approximation is sufficient. We can therefore approximate the ghost value of pressure by

$$\bar{P}_{gi} = \bar{P}_i - 2d \cdot \frac{\bar{\rho}_i \bar{V}_i^2}{R} \quad (2.12)$$

where \bar{P}_i , $\bar{\rho}_i$, and \bar{V}_i are the control volume average pressure, density and velocity respectively, and d is the distance of the control volume centroid

2.4. High-Order Limiting

from the wall in the convex direction (d is negative for concave boundaries). The ghost values of Mach number and density can be obtained by considering the isentropic transformation from the control volume state to the ghost value state with a pressure of \bar{P}_{gi}

$$\begin{aligned}\bar{\rho}_{gi} &= \bar{\rho}_i \left(\frac{\bar{P}_{gi}}{\bar{P}_i} \right)^{\frac{1}{\gamma}} \\ \bar{M}_{gi}^2 &= \frac{2}{\gamma - 1} \left(\left(\frac{\bar{P}_{t_i}}{\bar{P}_{gi}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right)\end{aligned}$$

where \bar{M}_{gi} is the Mach number in the ghost control volume and \bar{P}_{t_i} is the total pressure as calculated using the boundary control volume average flow properties. This, together with the assumption that the flow direction remains tangential to the surface, fully establishes the state of the ghost control volume.

2.4.4.2 Stagnation Point

Additional steps need to be taken to prevent the application of the limiter at stagnation points. Although affecting a very small fraction of control volumes, obtaining high-order accuracy near stagnation points can be critical to global accuracy.

In practical aerodynamics problems, the flow discontinuity which requires the proper application of the limiter is the shock. Since supersonic flow is required to produce a shock, and stagnation points are necessarily in subsonic regions, it is possible to simply disable the limiter in the latter areas. Specifically, we propose that the reconstruction of a control volume can only be adversely affected by a shock if at least one of the control volumes in its reconstruction stencil contains supersonic flow. Practically, we wish to smoothly disable the limiter as the highest control volume average Mach number in the reconstruction stencil is reduced. For this purpose we can reuse the approach of Subsection 2.4.3. Specifically, the limiter value is once

2.4. High-Order Limiting

again modified such that

$$\hat{\Phi}_i = \hat{\sigma}_i + (1 - \hat{\sigma}_i)\tilde{\Phi}_i, \quad (2.13)$$

where $\tilde{\Phi}_i$ is the limiter value after applying the uniform-flow fix in Equation 2.9 and

$$\hat{\sigma}_i = \begin{cases} 1 & M_{i,max} \leq M_1 \\ s \left(\frac{M_{i,max} - M_1}{M_2 - M_1} \right) & M_1 < M_{i,max} < M_2 \\ 0 & M_{i,max} \geq M_2 \end{cases} \quad (2.14)$$

where $M_{i,max}$ is the maximum Mach number of the control volume averages of the reconstruction stencil of control volume i , and s is the function in Equation 2.11. The parameters M_1 and M_2 define the Mach numbers at which the effect of the limiter is fully disabled and fully enabled respectively. In the present work we use $M_1 = 0.8$ and $M_2 = 0.85$. The new limited reconstruction used by the flux integration scheme becomes $U_i^R(\vec{x} - \vec{x}_i, \hat{\Phi}_i)$. For typical aerodynamics flows there is no harm in applying this step to all control volumes, even if they are not near boundaries. Doing so can reduce computational effort since $\tilde{\Phi}_i$ does not need to be evaluated if $\hat{\sigma}_i$ evaluates to 1.

2.4.5 Complete Algorithm

The complete algorithm is, in the general case, more complex and costly than the second-order limiting procedure presented in Section 2.3. However, the various additions can, in some cases, be used to “short circuit” the evaluation of the limiter, therefore reducing computational effort. The complete algorithm for applying the limiter to the reconstruction for each flow property of each control volume is

1. Find the maximum Mach number of the reconstruction neighbors control volume averages of the solution and evaluate $\hat{\sigma}_i$ using Equation 2.14. If $\hat{\sigma}_i = 1$ then $\hat{\Phi}_i = 1$ and the algorithm jumps to step 9.
2. Find the largest negative ($\delta U_i^{min} = \min(\bar{U}_j - \bar{U}_i)$) and positive ($\delta U_i^{max} = \max(\bar{U}_j - \bar{U}_i)$) difference between the solution in the immediate neigh-

bors j and the current control volume i . If the control volume or any of its immediate neighbors are adjacent to a wall boundary include their ghost values using Equation 2.12 and isentropic relations.

3. Compute $\tilde{\sigma}_i$ using Equation 2.10. If $\tilde{\sigma}_i = 1$ then $\hat{\Phi}_i = \tilde{\Phi}_i = 1$ and the algorithm jumps to step 9.
4. Compute the unconstrained reconstructed value at each Gauss point ($U_{ik} = U_i^R(\vec{x}_k - \vec{x}_i)$).
5. Compute a maximum allowable value of ϕ_{ik} for each Gauss point k .

$$\phi_{ik} = \begin{cases} \widetilde{\min} \left(1, \frac{\delta U_i^{max}}{U_{ik} - \bar{U}_i} \right), & \text{if } U_{ik} - \bar{U}_i > 0 \\ \widetilde{\min} \left(1, \frac{\delta U_i^{min}}{U_{ik} - \bar{U}_i} \right), & \text{if } U_{ik} - \bar{U}_i < 0 \\ 1, & \text{if } U_{ik} - \bar{U}_i = 0 \end{cases}$$

6. Select $\Phi_i = \min(\phi_{ik})$.
7. Compute $\tilde{\Phi}_i$ using Equation 2.9.
8. Compute $\hat{\Phi}_i$ using Equation 2.13.
9. Compute the limited reconstruction $U_i^R(\vec{x} - \vec{x}_i, \hat{\Phi}_i)$ at flux Gauss integration points using Equation 2.8.

2.5 Results

The presented results were obtained using a Newton-GMRES [17] vertex-centered finite-volume solver. The solution process consists of two stages. In the preiteration stage the linear system resulting from a local timestepping is solved at each iteration. The Jacobian from the first-order accurate scheme is used on the left-hand side and the full-order accurate flux is used on the right-hand side. At each Newton iteration, the linear system is approximately solved using incomplete-lower-upper factorization (ILU) preconditioned GMRES. During the second stage, the left-hand side is

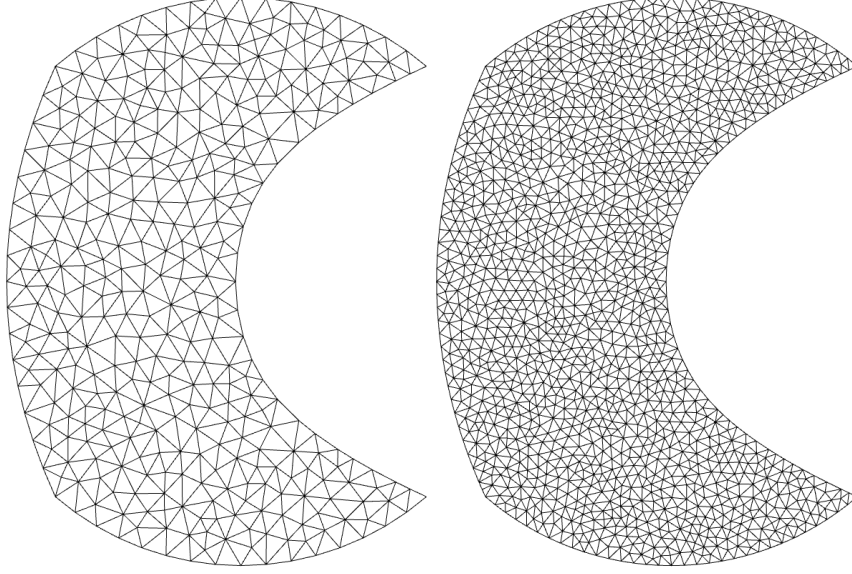


Figure 2.4: Two coarsest meshes used for Ringleb's flow test case

replaced with the full-order accurate Jacobian as described in the manuscript reproduced in Chapter 3.

In addition to results using the new limiter, results are also presented for the high-order scheme using the procedure in Section 2.4.1 but with Venkatakrishnan's limiting function. For Venkatakrishnan's function we use a tuning parameter of $K = 2$ and for the new limiter we use $K = 1$.

2.5.1 Ringleb's Flow

We begin by considering Ringleb's flow which is transonic but shockless and has a known exact solution. This will enable us to quantify the negative effects of Venkatakrishnan's limiter and the new limiter on the accuracy of the solution in smooth regions of flow. We consider four meshes consisting of 369, 1426, 5467 and 20690 control volumes. The two coarsest meshes are shown in Figure 2.4.

The exact solution to Ringleb's flow is given by the streamlines defined

by:

$$\begin{aligned} x &= \frac{1}{2} \frac{1}{\rho} \left(\frac{1}{q^2} - \frac{2}{k^2} \right) + \frac{J}{2} \\ y &= \pm \frac{1}{k\rho q} \sqrt{1 - \left(\frac{q}{k} \right)^2} \end{aligned}$$

where k is constant along streamlines and

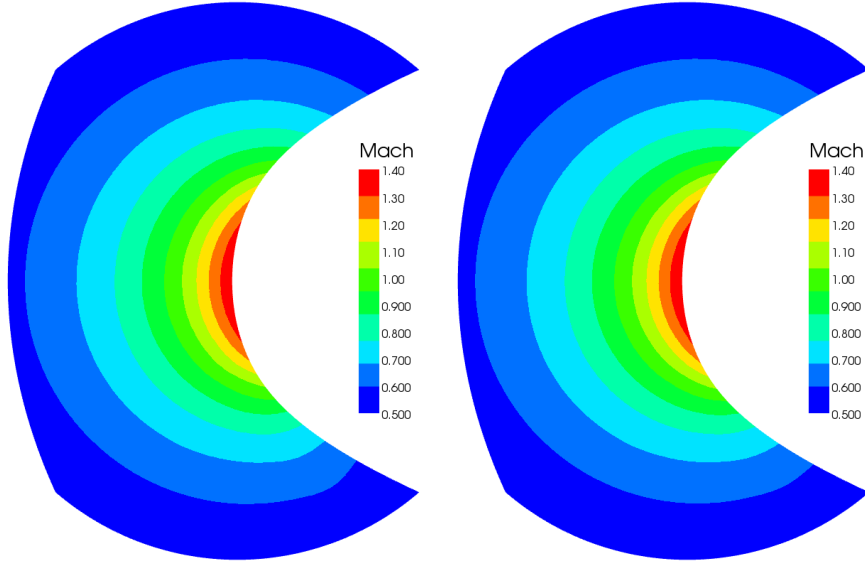
$$\begin{aligned} q &= |\vec{V}| \\ J &= \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2} \ln \frac{1+c}{1-c} \\ c &= \sqrt{1 - \frac{\gamma-1}{2} q^2} \\ \rho &= c^{2/(\gamma-1)} \end{aligned}$$

The computational domain is constructed using solid walls at the streamlines for $k = 0.55, 1.2$ and placing the inlet and outlet at $q = 0.5$. This results in a geometry that is symmetric about the horizontal axis. The flow is subsonic at the inlet and outlet but is supersonic near the inside wall of the throat. Using this domain, rather than just the upper half, is a more stringent test of the accuracy of the computational scheme since errors produced at the throat are allowed to propagate to the lower half of the domain.

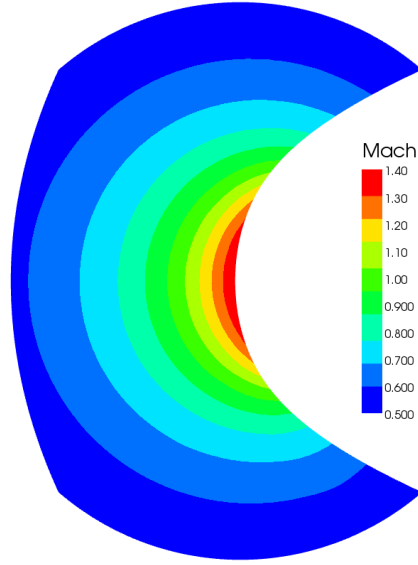
Since this test case contains no stagnation points, the Mach number dependent deactivation of the limiter should not be necessary. For this reason, and to make the test as stringent as possible, we do not apply steps 1 and 8 of the high-order limiting procedure as presented in Subsection 2.4.5.

As an initial qualitative assessment, we compare the Mach number contours generated on the 1426 control volume mesh using Venkatakrishnan's limiter and the new limiter in Figure 2.5. The additional dissipation of the Venkatakrishnan limited solutions is evident in the reduced Mach number at the inner wall near the outflow. Using the same criterion we can visually detect a slight superiority of the fourth-order scheme over the second-order scheme when using the new limiter. The Venkatakrishnan limited procedure, on the other hand, does not benefit from the use of the high-order accurate

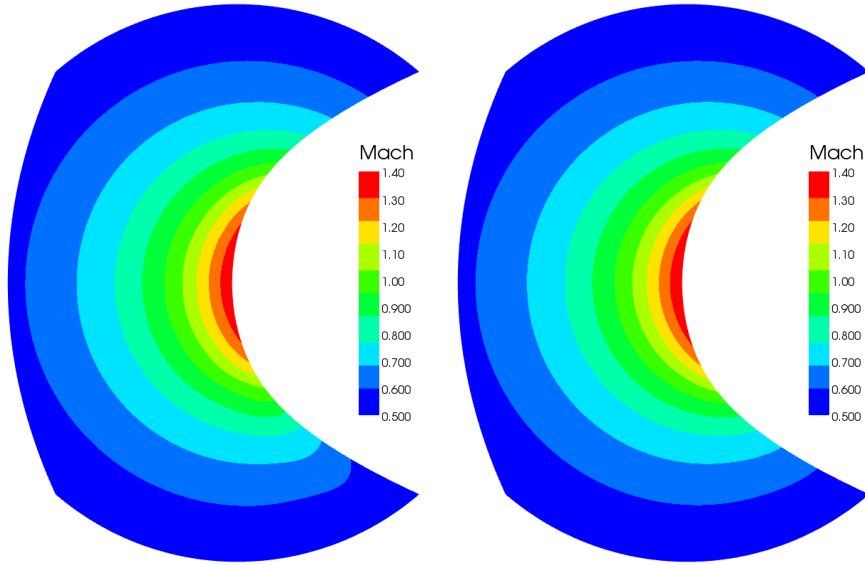
2.5. Results



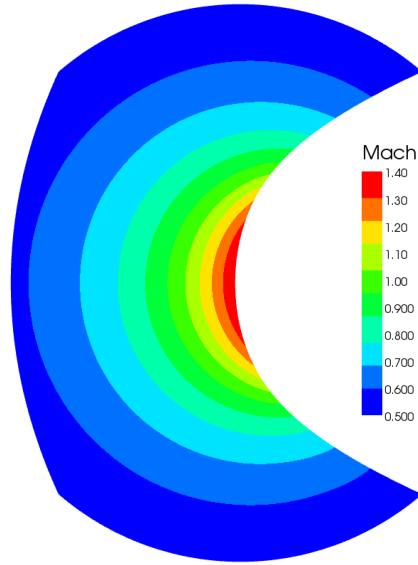
(a) 2nd-order with Venkatakrishnan limiter



(b) 2nd-order with new limiter



(c) 4th-order with Venkatakrishnan limiter



(d) 4th-order with new limiter

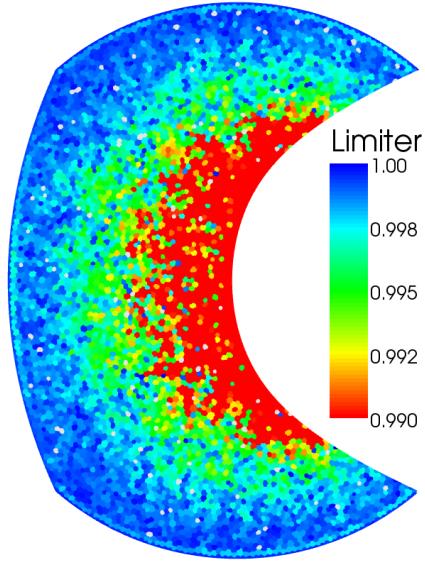
Figure 2.5: Mach number contours from numerical solution of Ringleb's flow on 1426 control volume mesh

reconstruction.

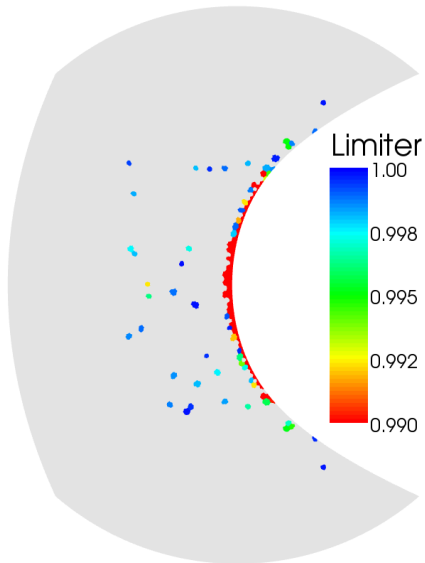
Next we examine the value of the limiter value Φ for the pressure component at steady state for the fourth-order solution on the 5467 control volume mesh using the two limiting schemes in Figure 2.6. In addition to Venkatakrishnan’s limiter and the new limiter, we also consider the new limiter without the special boundary ghost value treatment introduced in steps 3 and 7 of the high-order limiting procedure. The new limiter successfully avoids limiting in almost all control volumes. All the control volumes that are limited have values $\Phi \geq 0.99$. Without the boundary ghost treatment significant limiting occurs near the inner wall boundary. Venkatakrishnan’s limiter, on the other hand, applies some limiting to virtually all control volumes.

Next, in Figure 2.7, entropy is plotted for the second- and fourth-order schemes on the second-finest grid. The new limiter causes no distinguishable additional entropy production relative to the unlimited scheme for the second-order method, and a very slight increase for the fourth-order method. On the other hand, Venkatakrishnan’s limiter increases entropy by approximately an order of magnitude for the second-order scheme and two to four orders for the fourth-order scheme. Once again we note that when applying Venkatakrishnan’s limiter there is no apparent benefit to using the fourth-order scheme over the second-order scheme.

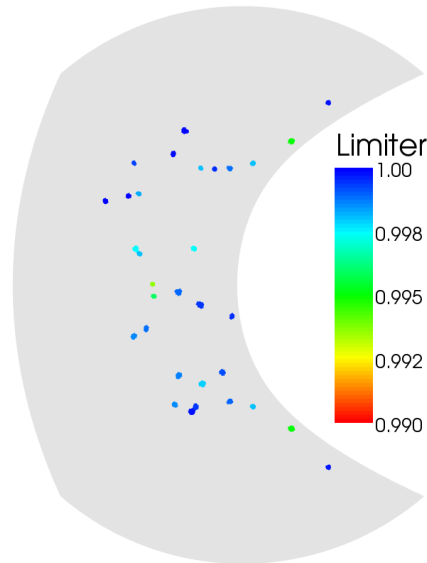
The L_2 norm of error of the converged solution compared to the known exact solution is used to generate Figure 2.8. The grid-convergence orders of the error norms using a regression fit of the results from all grids are given in Table 2.1. We begin by noting that the unlimited schemes attain their nominal orders of accuracy in L_1 and L_2 norms and one order accuracy less than nominal for the L_∞ norm. Although the error of second-, third-, and fourth-order schemes using Venkatakrishnan’s limiter is lower than that of the first-order scheme, the error does not converge with nominal accuracy. In fact the grid convergence of the error when using Venkatakrishnan’s limiter is only first-order. The new limiter, on the other hand, allows for the second- and third-order schemes to perform virtually identically to the unlimited case and has only minimal adverse effect on the accuracy of the fourth-order scheme. The results using the new limiter without the boundary ghost value



(a) Venkatakrishnan's limiter



(b) New limiter without boundary ghost values



(c) New limiter

Figure 2.6: Limiter value for pressure for the fourth-order converged solution on the 5467 control volume mesh. Only values $\Phi \neq 1$ are plotted

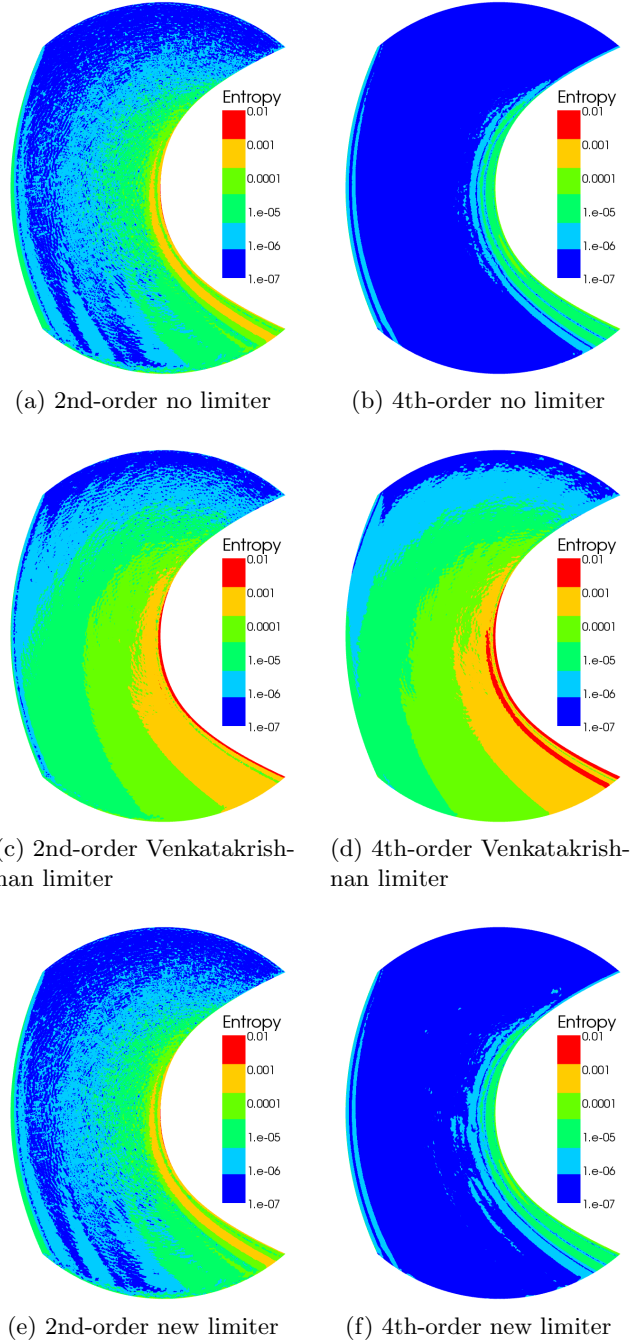
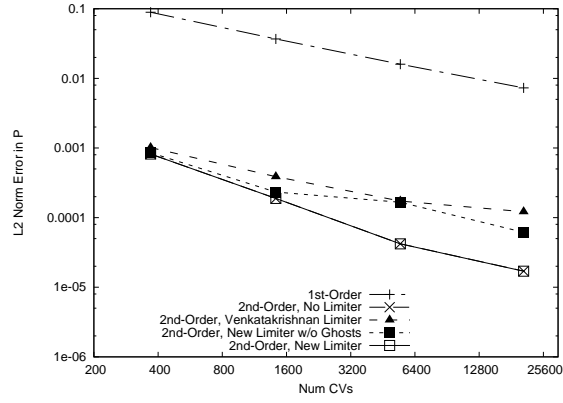
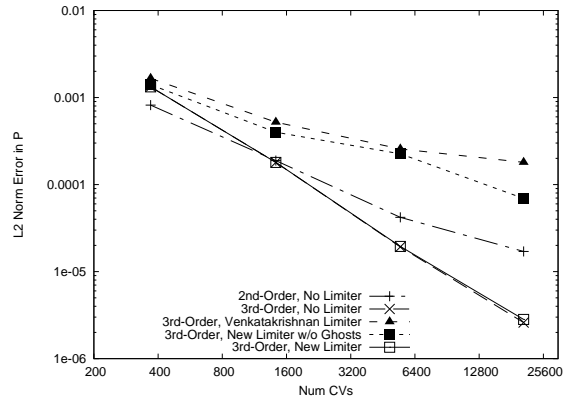


Figure 2.7: Difference in dimensionless entropy from the freestream value for Ringleb's flow

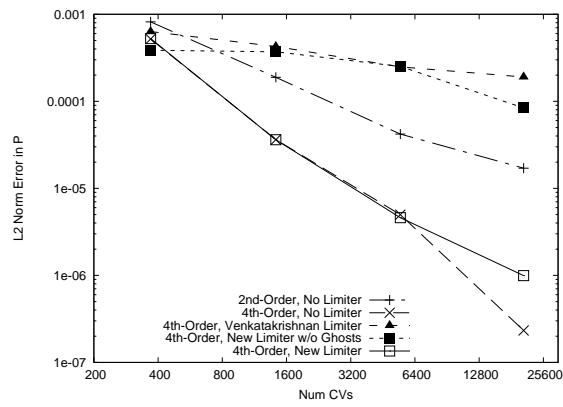
2.5. Results



(a) Second-order



(b) Third-order



(c) Fourth-order

Figure 2.8: Error convergence for pressure in Ringleb's flow

2.5. Results

Nominal Order	Limiter	L_1 Norm	L_2 Norm	L_∞ Norm
1st	None	1.24	1.24	0.99
2nd	None	2.22	1.96	1.24
2nd	Venkatakrishnan	1.10	1.07	0.47
2nd	New w/o Ghosts	1.71	1.23	0.47
2nd	New	2.22	1.96	1.24
3rd	None	3.18	3.12	2.70
3rd	Venkatakrishnan	1.14	1.10	0.51
3rd	New w/o Ghosts	1.83	1.43	0.59
3rd	New	3.14	3.08	2.40
4th	None	4.07	3.74	3.06
4th	Venkatakrishnan	0.62	0.62	0.12
4th	New w/o Ghosts	1.07	0.74	0.23
4th	New	3.24	3.11	2.08

Table 2.1: Convergence order of norms of error in pressure for Ringleb’s flow computed using regression fit of all mesh results.

treatment outperform Venkatakrishnan’s limiter only slightly. This indicates that the application of the limiter in even an isolated region has severe implications for global accuracy.

2.5.2 Transonic Flow Over an Airfoil

Next, we present results for transonic flow over a NACA 0012 airfoil at Mach 0.8 and an angle of attack of 1.25 degrees. The computational mesh consists of 4656 control volumes and is shown in Figure 2.9. We will consider second- and fourth-order schemes using Venkatakrishnan’s limiter, the new limiter, and the new limiter without the stagnation region fix which disables the limiter at low Mach number in steps 1 and 8 of the high-order limiting procedure.

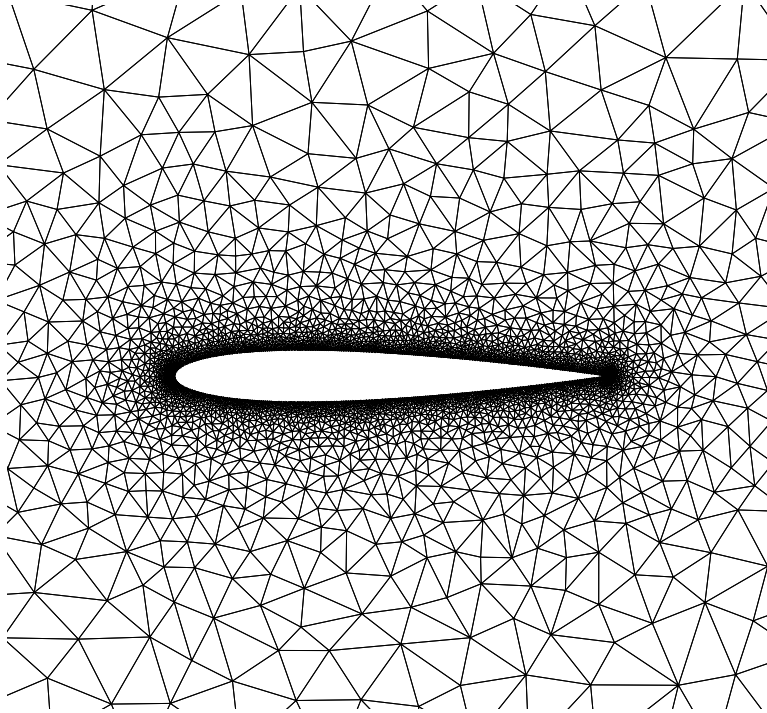


Figure 2.9: Mesh consisting of 4656 control volumes used for the NACA 0012 test case

2.5.2.1 Accuracy

We begin by assessing the quality of the solution upstream of the shock. Entropy near the leading edge of the airfoil is plotted in Figure 2.10. The fourth-order solution with Venkatakrishnan’s limiter once again fails to outperform its second-order counterpart. The new limiter produces approximately an order of magnitude less entropy than Venkatakrishnan’s limiter for the second-order scheme. The new limiter applied to the fourth-order scheme results in even less entropy production. Disabling the stagnation region fix results in a modest increase in entropy production.

The quality of the solutions can also be compared by examining the stagnation pressure ratio along the upper surface of the airfoil shown in Figure 2.11. The decrease in total pressure across the shock is comparable for all schemes. However, the schemes limited with the Venkatakrishnan limiter result in substantial stagnation pressure loss upstream of the shock. For the second-order scheme the stagnation region fix has little effect while for the fourth-order scheme applying this step in the limiting procedure results in a further improvement in the conservation of total pressure.

2.5.2.2 Shock Monotonicity

The performance of the new limiter in controlling oscillations and overshoots in the solution near the strong shock on the upper surface of the airfoil is demonstrated in the pressure plot of Figure 2.12. The new limiter and Venkatakrishnan’s limiter are both effective in eliminating any substantial oscillations near the shock. The pressure computed on the upper surface of the airfoil using the different limiters is virtually indistinguishable. Once again the lower dissipation of the new limiter is demonstrated by the sharper profile of the weak shock on the lower surface of the airfoil.

2.5.2.3 Convergence

Next, we consider the residual convergence properties of the new limiting scheme coupled with our Newton-GMRES solver. Figure 2.13 shows the

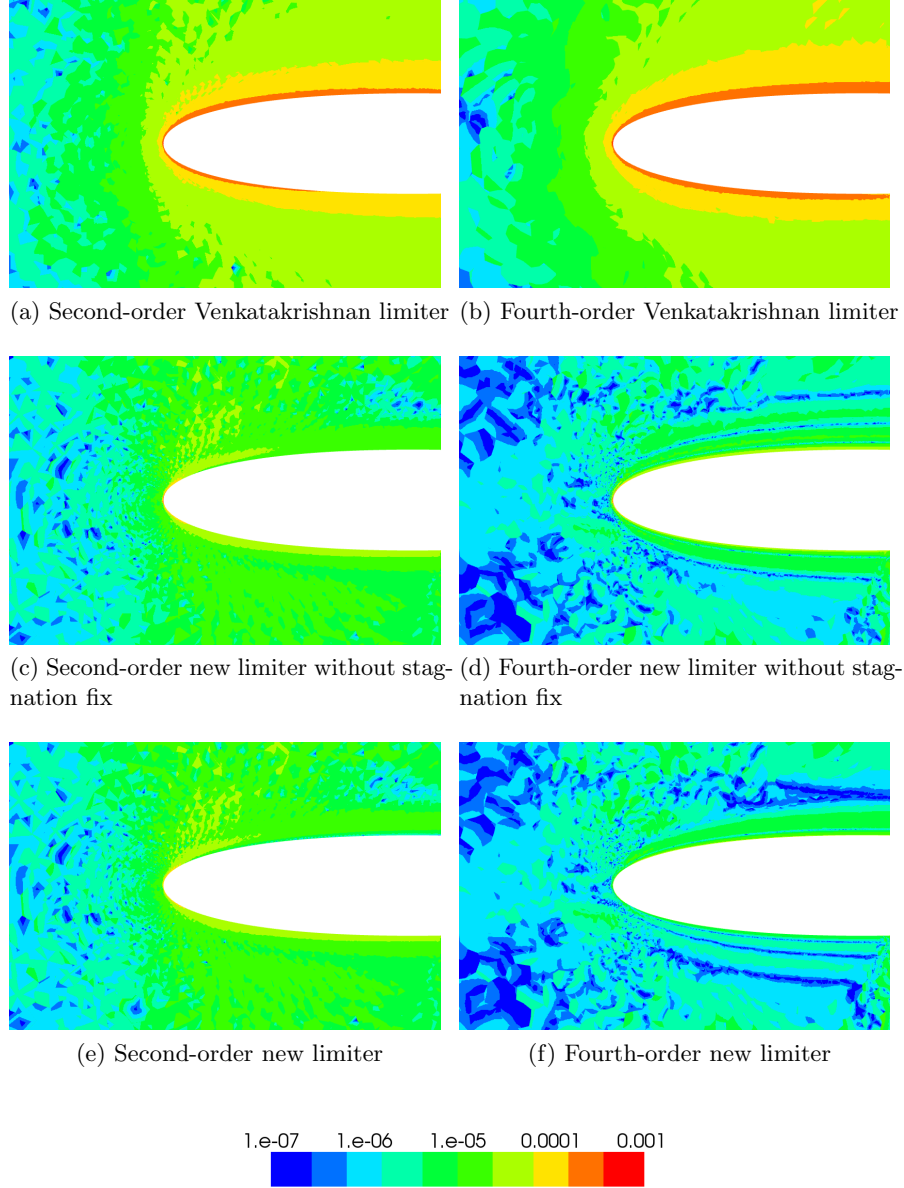
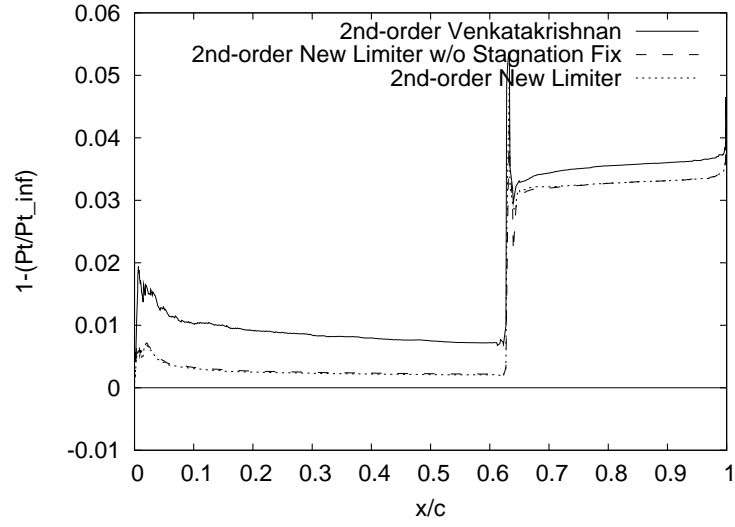
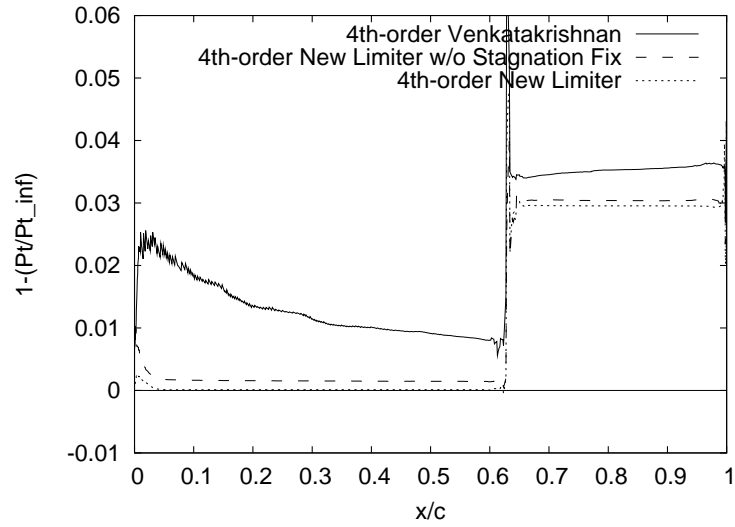


Figure 2.10: Difference in dimensionless entropy from the freestream value for Mach 0.8, $\alpha = 1.25$ flow around a NACA 0012 airfoil

2.5. Results



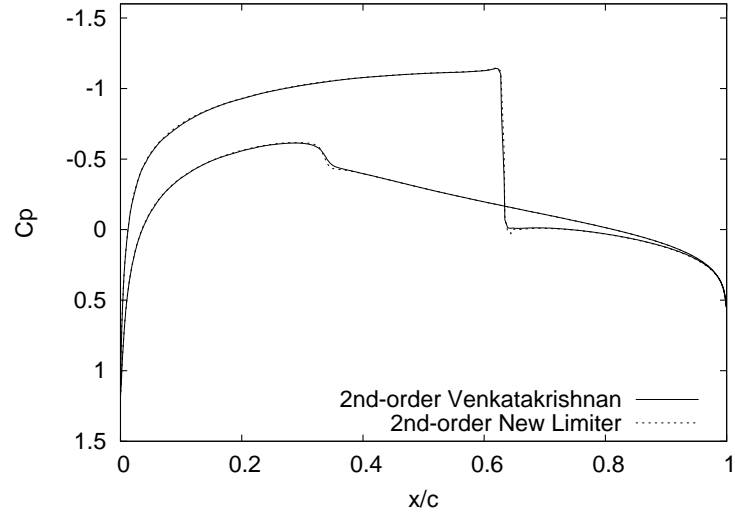
(a) Second-order



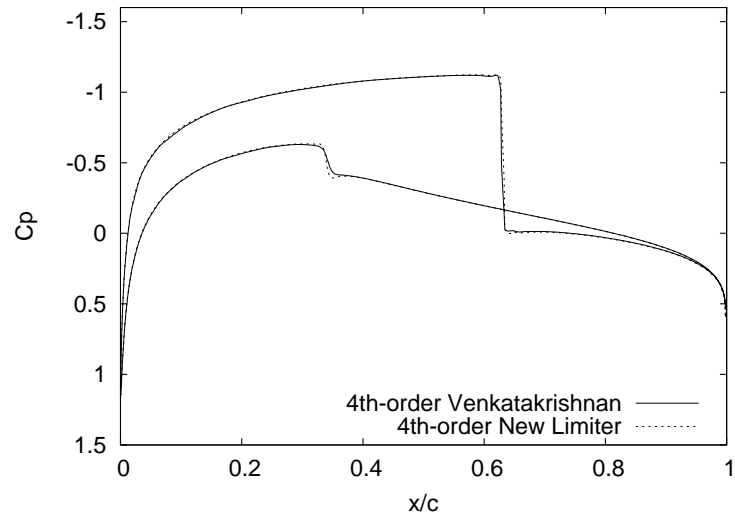
(b) Fourth-order

Figure 2.11: Decrease in total pressure along the upper surface of the NACA 0012 airfoil at Mach 0.8 $\alpha = 1.25$

2.5. Results



(a) Second-order



(b) Fourth-order

Figure 2.12: Surface pressure profiles for transonic flow over a NACA0012 airfoil

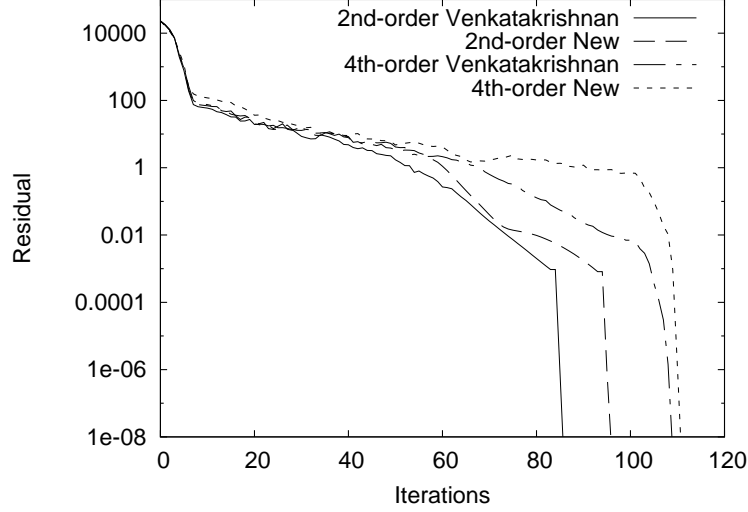


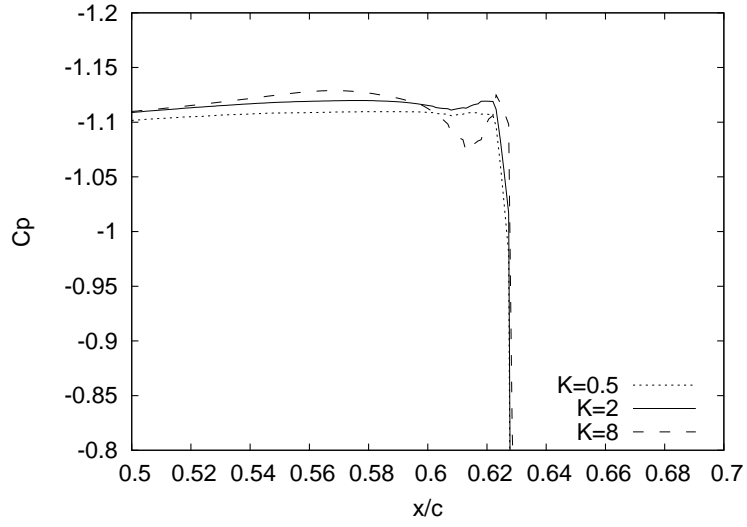
Figure 2.13: Convergence history for transonic airfoil case

Order	Limiter	Computational Time (s)
2nd	Venkatakrishnan	37
2nd	New	40
4th	Venkatakrishnan	84
4th	New	99

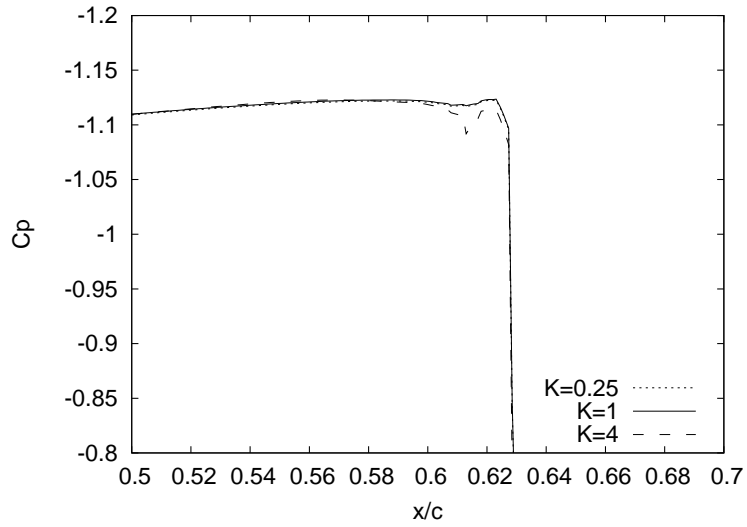
Table 2.2: Computational time for transonic airfoil test case

convergence of the scheme with the new limiter relative to Venkatakrishnan's limiter for second and fourth order accurate schemes. We have also provided the computational time required for convergence in Table 2.2. The scheme with the new limiter exhibits a slightly poorer convergence rate than Venkatakrishnan's limiter. This is likely due to the lower dissipation of the scheme which results in poorer conditioning of the nonlinear system of equations. Similarly, we find that the fourth-order scheme requires slightly more iterations to converge than the second-order scheme.

2.5. Results



(a) Venkatakrishnan limiter



(b) New limiter

Figure 2.14: Upper surface pressure profile upstream of the shock for transonic flow over a NACA0012 airfoil using fourth-order scheme

2.5.2.4 Sensitivity to Tuning Parameter

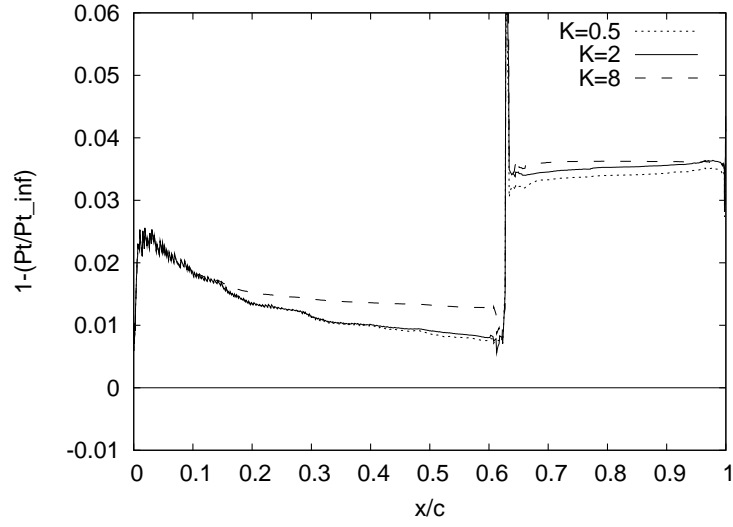
Venkatakrishnan's limiter and the new limiter both require a tuning parameter K to detect regions of uniform flow. Ideally the solutions produced by the schemes should be relatively insensitive to the choice of this parameter. Therefore we consider the effect of increasing and decreasing the parameter by a factor of 4 from the original choice. For Venkatakrishnan's scheme we consider $K = 0.5, 2, 8$ and for the new limiter we consider $K = 0.25, 1, 4$. In Figure 2.14 we examine the behaviour of pressure on the upper surface of the airfoil just upstream of the shock using the fourth-order scheme. The solution using Venkatakrishnan's limiter exhibits a strong sensitivity to the value of the parameter. Specifically, $K = 0.5$ results in a lower peak pressure ahead of the shock, while $K = 8$ results in significant oscillations. The result using the new limiter shows less sensitivity. Specifically, the results using $K = 0.25$ are virtually indistinguishable from those with $K = 1$ while the results with $K = 4$ result in some oscillation.

To further examine the effect of this parameter on the dissipation of the scheme, the stagnation pressure along the upper surface of the airfoil is shown in Figure 2.15. For Venkatakrishnan's scheme the value of the tuning parameter has little effect on the total pressure loss near the leading edge of airfoil. However the stagnation pressure loss across the shock exhibits a strong dependence on this parameter. The new limiter exhibits almost no sensitivity to the parameter ahead of the shock. Downstream of the shock the results using parameters $K = 0.25$ and $K = 1$ are indistinguishable while the stagnation pressure with $K = 4$ is higher.

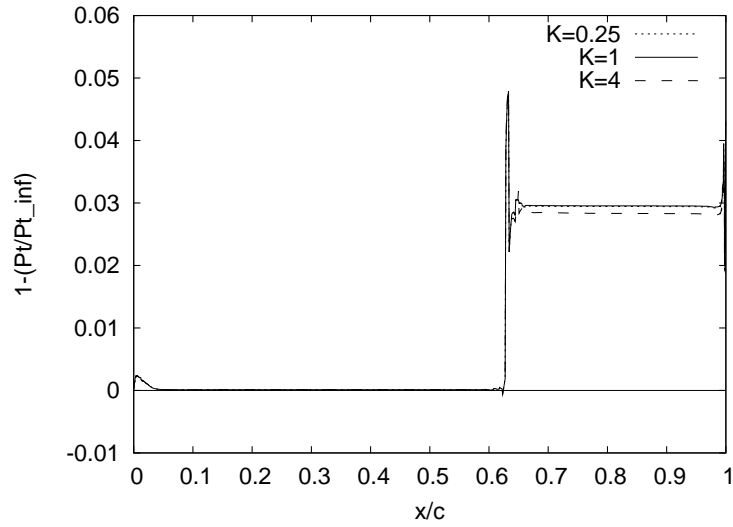
2.6 Conclusion

A new unstructured grid limiter broadly based on the framework introduced by the Barth and Jespersen limiter has been developed specifically for high-order schemes. Like the Venkatakrishnan limiter, it is designed to be differentiable to enable good convergence to steady state by implicit schemes. To achieve this and to maintain high-order accuracy, a new function defining

2.6. Conclusion



(a) Venkatakrishnan limiter



(b) New limiter

Figure 2.15: Sensitivity of total pressure along the upper surface of the NACA 0012 airfoil to the choice of limiter tuning parameter

the relation between the limiter value and the unlimited reconstruction at Gauss points has been constructed. To preserve accuracy at smooth extrema, a smooth switch disables the limiter in regions of nearly uniform flow. To avoid limiting regions of smooth flow near boundaries, the range defining monotonicity is carefully expanded based on a constant momentum extrapolation of conditions beyond the boundary. Application of the limiter is avoided at stagnation points by smoothly disabling it in regions of low Mach number.

The results indicate that the new limiter is effective in maintaining high-order accuracy in smooth flows while effectively suppressing oscillations near shocks. Additionally, the new scheme is also relatively insensitive to tuning parameters which should lead to robustness and reproducibility of solutions. Although the greatest gains in accuracy are seen when using a high-order scheme, the new limiter is also effective in reducing the dissipation and increasing the accuracy of second-order schemes relative to the currently most widely used limiter. Since only minor modifications to the reconstruction procedure are needed, the addition of the new method to existing solvers should be relatively straightforward.

Although the present work presents only inviscid results in two dimensions, the extension to viscous flows and three dimensions should be relatively simple. Specifically, for viscous flows the stagnation region fix should be sufficient in eliminating the effect of the limiter near walls making the wall ghost value fix unnecessary. For three-dimensional inviscid flows the wall ghost value fix will need to account for streamline curvature even though streamlines will not be coordinate aligned.

2.7 Bibliography

- [1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.
- [2] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA paper 93-0668, Jan. 1993.
- [3] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *Lecture Series 1994-05*, Rhode-Saint-Genèse, Belgium, Mar. 1994. von Karman Institute for Fluid Dynamics.
- [4] T. J. Barth and P. O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, Jan. 1990.
- [5] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA paper 89-0366, Jan. 1989.
- [6] S. De Rango and D. W. Zingg. Higher-order spatial discretization for turbulent aerodynamic computations. *American Institute of Aeronautics and Astronautics Journal*, 39(7):1296–1304, July 2001.
- [7] M. Delanaye and J. A. Essers. Quadratic-reconstruction finite volume scheme for compressible flows on unstructured adaptive grids. *American Institute of Aeronautics and Astronautics Journal*, 35(4):631–639, Apr. 1997.
- [8] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144(1):194–212, July 1998.
- [9] P. Geuzaine, M. Delanaye, and J.-A. Essers. Computation of high Reynolds number flows with an implicit quadratic reconstruction scheme

- on unstructured grids. In *Proceedings of the Thirteenth AIAA Computational Fluid Dynamics Conference*, pages 610–619. American Institute of Aeronautics and Astronautics, 1997.
- [10] A. G. Godfrey, C. R. Mitchell, and R. W. Walters. Practical aspects of spatially high-order accurate methods. *American Institute of Aeronautics and Astronautics Journal*, 31(9):1634–1642, Sept. 1993.
 - [11] C. Q. Hu and C. W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, Mar. 1999.
 - [12] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, June 1996.
 - [13] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2592–2609, 2008.
 - [14] C. F. Ollivier-Gooch. High-order ENO schemes for unstructured meshes based on least-squares reconstruction. AIAA paper 97-0540, Jan. 1997.
 - [15] C. F. Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.
 - [16] C. F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.
 - [17] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3):856–869, July 1986.
 - [18] C.-W. Shu, T. A. Zang, G. Erlebacher, D. Whitaker, and S. Osher. High-order ENO schemes applied to two- and three-dimensional compressible flow. *Applied Numerical Mathematics*, 9(1):45–71, 1992.

2.7. Bibliography

- [19] G. D. van Albada, B. van Leer, and W. W. Roberts, Jr. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76–84, Apr. 1982.
- [20] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *Journal of Computational Physics*, 32:101–136, 1979.
- [21] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady-state solutions. AIAA paper 93-0880, Jan. 1993.
- [22] D. Zingg, S. De Rango, M. Nemec, and T. Pulliam. Comparison of several spatial discretizations for the Navier-Stokes equations. *Journal of Computational Physics*, 160:683–704, 2000.

Chapter 3

Efficient Convergence to Steady State[†]

3.1 Introduction

High-order discretizations on structured grids have been shown [6, 24] to reduce computational effort for a given level of solution accuracy. Although high-order finite-volume methods on unstructured grids are relatively well known [1, 2, 7], means to efficiently converge the solution to steady state have not been adequately studied. For this reason, the advantages of high-order methods have not been convincingly demonstrated.

Although Newton-GMRES [21] methods provide rapid convergence, they are known to be less robust than explicit schemes. This lack of robustness is directly related to the choice of globalization technique. For compressible flows, the technique of choice has been the addition of a timestep that is increased as the solution converges. Successful convergence often depends on the careful choice of parameters controlling the growth of this timestep. High-order schemes, due to their lower diffusivity, are particularly sensitive to these robustness issues. In this work, we demonstrate how the timestep approach can be improved by the addition of a line search method. We also demonstrate that scaling the local timestep based on the local residual can dramatically improve performance and robustness.

The use of the GMRES method to approximately solve the linear systems arising from implicit schemes is an increasingly popular approach for solution

[†]A version of this chapter will be submitted for publication. C. Michalak and C. Ollivier-Gooch. Globalized matrix-explicit Newton-GMRES for the high-order accurate solution of the Euler equations.

convergence for second-order unstructured flow solvers. Previous work has focused on using the matrix-free [3] approach. With this method the full flux Jacobian does not need to be explicitly constructed. Instead, the matrix-vector products required by GMRES are computed using Frechet derivatives. However, a preconditioning matrix is still required for good GMRES convergence; the typical choice here is to compute the Jacobian of the first-order scheme (though perhaps using the reconstructed second-order data) and use an incomplete LU factorization of this matrix to precondition the linear system. Our group's previous work in developing an efficient high-order accurate flow solver extended this approach to high-order residual calculations, including careful study of preconditioning methods [15, 16].

While those studies were successful in dramatically reducing CPU time requirements for high-order schemes, especially for third-order accuracy, there are also clear indications that the fourth-order scheme in particular is poorly preconditioned by the first-order Jacobian. The research described in this paper explores the possibility of forming and exploiting the full order Jacobian. On the one hand, we expect that preconditioning will be much better if we use the full order Jacobian. Also, having the Jacobian on hand will eliminate all the flux evaluations done within GMRES inner iterations in matrix-free schemes; this savings in CPU time should help offset the cost of computing the Jacobian. The major drawback to forming the full order Jacobian is the memory requirements.

Section 3.2 briefly reviews the high-order discretization used in the present work, focusing on the reconstruction step. In Section 3.3 globalization techniques for the Newton method are discussed. In this section we discuss how the line search method can be effectively combined with the timestepping method. We also present our novel method of deriving a local timestep from the residual. In Section 3.4 we discuss the matrix-free and matrix-explicit GMRES methods and their preconditioning. Some detail on how to efficiently form the exact Jacobian needed by the matrix-explicit method is provided. Numerical results for a multi-element subsonic test case and a single element transonic test case are provided in Section 3.5. Concluding remarks are found in Section 3.6.

3.2 High-Order Accurate Finite-Volume Method

The high-order accurate finite-volume discretization relies on a piecewise polynomial reconstruction in each control volume to provide an accurate approximation to the solution at Gauss integration points. A series of Riemann problems for the Euler equation are solved at these points using the Roe scheme [20]. This results in a discrete flux integral for each control volume which must be reduced to zero at steady state. In the present work, the control volumes are formed using a vertex-centered median dual of an unstructured triangular grid.

3.2.1 Governing Equations

The non-dimensionalized two-dimensional Euler equations for a control volume V_i can be written in integral form as

$$\frac{d}{dt} \int_{V_i} U dA + \oint_{\Omega_i} F dS = 0 \quad (3.1)$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad F = \begin{bmatrix} \rho u_n \\ \rho u u_n + P \hat{n}_x \\ \rho v u_n + P \hat{n}_y \\ (E + P) u_n \end{bmatrix} \quad (3.2)$$

where U is the solution vector in conservative variables, and F is the flux vector normal to the control volume boundary. The total energy per unit volume, E , and the pressure, P , are related by the ideal gas equation of state

$$P = \rho R T \quad (3.3)$$

3.2.2 Finite-Volume Formulation

The governing equations are discretized in space using the finite-volume method on an unstructured grid. The control volume average of the solution

in each control volume is defined by

$$\bar{U}_i = \frac{1}{A_i} \int_{V_i} U dA$$

which can be used to rewrite Equation 3.1 as

$$\frac{d\bar{U}_i}{dt} = -\frac{1}{A_i} \oint_{\Omega_i} F dS \equiv -R_i(\bar{U}) \quad (3.4)$$

Since in the present work we are concerned only with steady-state solutions the time derivative term can be dropped. Therefore the system of equations to be solved is simply

$$R(\bar{U}) = 0$$

where R is the vector function representing the flux integral, or residual, in all control volumes.

3.2.3 Solution Reconstruction

To obtain a high-order accurate numerical approximation to the residual we reconstruct a polynomial representation of the solution within each control volume. As in previous work [17], the high-order accurate reconstruction is obtained by formulating and solving a least-squares problem.

To obtain an accurate estimate of the solution at flux quadrature points, we represent the solution within each control volume by the Taylor series expansion

$$\begin{aligned} U_i^R(x - x_i, y - y_i) = & U|_i + \frac{\partial U}{\partial x}\bigg|_i (x - x_i) + \frac{\partial U}{\partial y}\bigg|_i (y - y_i) + \\ & \frac{\partial^2 U}{\partial x^2}\bigg|_i \frac{(x - x_i)^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_i (x - x_i)(y - y_i) + \\ & \frac{\partial^2 U}{\partial y^2}\bigg|_i \frac{(y - y_i)^2}{2} + \dots \end{aligned}$$

3.2. High-Order Accurate Finite-Volume Method

where U_i is the value of the reconstructed solution and $\frac{\partial^{k+l}U_i}{\partial x^k \partial y^l}$ are its derivatives at the reference point (x_i, y_i) of control volume i . The coefficients of the polynomial are computed so that the mean value of the solution in the control volume is conserved and the reconstruction approximates nearby control volume averages.

Conservation of the mean within a control volume is satisfied when

$$\begin{aligned} \bar{U}_i = \frac{1}{A_i} \int_{V_i} U_i^R dA \equiv & U|_i + \frac{\partial U}{\partial x} \Big|_i \bar{x}_i + \frac{\partial U}{\partial y} \Big|_i \bar{y}_i + \\ & \frac{\partial^2 U}{\partial x^2} \Big|_i \frac{\bar{x}_i^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_i \bar{x}_i \bar{y}_i + \\ & \frac{\partial^2 U}{\partial y^2} \Big|_i \frac{\bar{y}_i^2}{2} + \dots \end{aligned} \quad (3.5)$$

where

$$\overline{x^n y^m}_i \equiv \frac{1}{A_i} \int_{V_i} (x - x_i)^n (y - y_i)^m dA$$

As we shall see, this mean constraint can be eliminated analytically from the least-squares system. The terms used to form the reconstruction least-squares problem have the form

$$\begin{aligned} \frac{1}{A_j} \int_{V_j} U_i^R (\vec{x} - \vec{x}_i) dA = & U|_i + \frac{\partial U}{\partial x} \Big|_i \widehat{x}_{ij} + \frac{\partial U}{\partial y} \Big|_i \widehat{y}_{ij} \\ & + \frac{\partial^2 U}{\partial x^2} \Big|_i \frac{\widehat{x}_{ij}^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_i \widehat{x}_{ij} \widehat{y}_{ij} + \frac{\partial^2 U}{\partial y^2} \Big|_i \frac{\widehat{y}_{ij}^2}{2} + \dots \end{aligned} \quad (3.6)$$

The geometric terms in this equation are of the general form

$$\begin{aligned} \widehat{x^n y^m}_{ij} & \equiv \frac{1}{A_j} \int_{V_j} ((x - x_j) + (x_j - x_i))^n \cdot ((y - y_j) + (y_j - y_i))^m dA \\ & = \sum_{l=0}^m \sum_{k=0}^n \frac{m!}{l! (m-l)!} \frac{n!}{k! (n-k)!} (x_j - x_i)^k \cdot (y_j - y_i)^l \cdot \overline{x^{n-k} y^{m-l}}_j \end{aligned}$$

The resulting least-squares problem to be solved for each solution variable in

3.2. High-Order Accurate Finite-Volume Method

each control volume takes the for

$$\begin{aligned}
 & \begin{bmatrix} 1 & \bar{x}_i & \bar{y}_i & \overline{x^2}_i & \overline{xy}_i & \overline{y^2}_i & \cdots \\ w_{i1} & w_{i1}\widehat{x}_{i1} & w_{i1}\widehat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\ w_{i2} & w_{i2}\widehat{x}_{i2} & w_{i2}\widehat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\ w_{i3} & w_{i3}\widehat{x}_{i3} & w_{i3}\widehat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\widehat{x}_{iN} & w_{iN}\widehat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots \end{bmatrix} \begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i \\
 &= \begin{pmatrix} \bar{U}_i \\ w_{i1}\bar{U}_1 \\ w_{i2}\bar{U}_2 \\ w_{i3}\bar{U}_3 \\ \vdots \\ w_{iN}\bar{U}_N \end{pmatrix} \tag{3.7}
 \end{aligned}$$

where the weights can be used to emphasize geometrically nearby data:

$$w_{ij} = \frac{1}{|\vec{x}_j - \vec{x}_i|^n} \tag{3.8}$$

where typically $n \in [0, 2]$. The first equation in this linear system is the mean constraint, which can be removed by Gauss elimination, leaving an unconstrained least-squares problem.

In previous work [14], the least-squares problem was solved at each flux evaluation using QR factorization [9]. However since the matrix contains only geometric terms, is identical for each solution variable in a given control volume and does not change between iterations, substantial savings in computational time can be achieved by precomputing and storing the pseudoinverse of the reconstruction matrix for each control volume. To obtain the pseudoinverse in a numerically stable manner, the present work uses the singular value decomposition (SVD) method [9]. Given a SVD of a

reconstruction matrix A , the pseudoinverse A^\dagger can easily be obtained

$$\begin{aligned} A &= U\Sigma V^T \\ A^\dagger &= V\Sigma^\dagger U^T \end{aligned} \tag{3.9}$$

where the diagonal entries of Σ are the singular values of A , the columns of U and V are the left- and right-singular vectors, Σ^\dagger is a diagonal matrix containing the reciprocal values of Σ .

Once this precomputation has been performed, the reconstruction coefficients in each control volume can be obtained by performing a matrix-vector multiplication; the number of columns in this matrix equals the number of control volumes in the reconstruction stencil, while the number of rows equals the number of required reconstruction coefficients. From these coefficients the reconstructed values of the solution can easily be computed at the flux quadrature points.

When solving flows with shocks, a slope limiter is applied to the reconstruction polynomial. We refer the reader to the manuscript reproduced in Chapter 2 for details of the high-order limiting algorithm used for the transonic results of the present work.

3.2.4 Numerical Flux Calculation and Integration

The control volume boundaries are formed by connecting the cell centroids with the face centroids surrounding vertices of the triangular grid. The contour integral in Equation 3.4 is discretized using Gauss quadrature over the dual edges. One Gauss point per edge is used for the second-order scheme. Two Gauss points are used for the third- and fourth-order schemes. The procedure to compute the numerical flux F is:

1. Translate the control volume average solution from conserved variables (density, momentum, energy) to primitive variables (density, velocity, and pressure).
2. Compute the reconstruction coefficients using the precomputed pseudoinverses of the reconstruction matrices and, if necessary, the limiter.

3. At each Gauss point, compute the left and right solution states using the reconstruction coefficients from both adjacent control volumes.
4. At each Gauss point, approximately solve the Riemann problem using the reconstructed solution from both adjacent control volumes. In the present work, the Roe scheme [20] is used.

3.3 Globalized Newton Method

Once a procedure to evaluate the discrete numerical residual has been established the remaining problem is to find the root of the system of equations

$$R(\bar{U}) = 0 \quad (3.10)$$

When a good starting guess for \bar{U} is known, the Newton method can be used. This results in an iterative method of the form

$$\begin{aligned} \frac{\partial R}{\partial \bar{U}} \delta \bar{U} &= -R(\bar{U}^n) \\ \bar{U}^{n+1} &= \bar{U}^n + \delta \bar{U} \end{aligned} \quad (3.11)$$

However, in practical problems a sufficiently good initial solution for \bar{U} is not known. Therefore the Newton method must be modified to improve its robustness. Two methods are commonly used for this in computational fluid dynamics. For aerodynamics flows most authors use a form of pseudo-timestepping [10, 12]. For incompressible flow a line search algorithm is sometimes used [18, 22]. In the present work we modify the Newton method by applying both a pseudo-timestep and a line search algorithm.

3.3.1 Pseudo-Timestepping

In pseudo-timestepping, instead of directly solving Equation 3.10 using the Newton method, we consider the series of root-finding problems defined by the implicit timestep

$$\bar{U}^{n+1} = \bar{U}^n - \text{diag}(\Delta t) R(\bar{U}^{n+1})$$

3.3. Globalized Newton Method

where \bar{U}^n is the solution from the previous timestep, \bar{U}^{n+1} is the solution at the next timestep, and Δt is a vector of local timesteps. This results in a root-finding problem at each timestep which we will define as

$$\tilde{R}(\bar{U}^{n+1}) \equiv R(\bar{U}^{n+1}) + \frac{I}{\Delta t} (\bar{U}^{n+1} - \bar{U}^n) = 0 \quad (3.12)$$

The addition of a timestep component has two related benefits. First, the root of the new residual \tilde{R} is closer to the previous iterate \bar{U}^n which makes it a better candidate for the Newton method. Second, since it mimics the time derivative in the original time-dependent Euler equations, it causes the transient solution to evolve in a physically feasible way. This prevents physically unfeasible states, such as negative pressure or density, which result in an undefined residual, from occurring.

The most common approach is to approximately solve \tilde{R} at each timestep using a single Newton step and to increase Δt as the solution progresses towards steady state. In this case the scheme reduces to the backward-Euler time advance scheme

$$\begin{aligned} \left(\frac{I}{\Delta t} + \frac{\partial R}{\partial \bar{U}} \right) \delta \bar{U} &= -R(\bar{U}^n) \\ \bar{U}^{n+1} &= \bar{U}^n + \delta \bar{U} \end{aligned} \quad (3.13)$$

3.3.2 Line Search

To add robustness, instead of solving Equation 3.12 using a pure Newton method we apply a single iteration of a line search algorithm [8]. The resulting solution update is given by

$$\bar{U}^{n+1} = \bar{U}^n + \alpha \delta \bar{U}$$

where $\delta \bar{U}$ is computed using a Newton iteration and α is a positive scalar step length. To select the value of α we consider an optimization problem related to Equation 3.12. Specifically, the objective function

$$z(x) = \frac{1}{2} \left\| \tilde{R}(x) \right\|_2^2$$

3.3. Globalized Newton Method

is required to satisfy a sufficient decrease condition given by

$$z(\bar{U}^{n+1}) \leq z(\bar{U}^n) + c_1 \alpha \nabla z^T \delta \bar{U} \quad (3.14)$$

where c_1 is taken to be 10^{-4} in the present work. The product of the gradient of the objective function with the Newton update vector is simply

$$\nabla z^T \delta \bar{U} = \tilde{R}^T \Big|_n \left(\frac{\partial \tilde{R}}{\partial \bar{U}} \Big|_n \delta \bar{U} \right)$$

and is therefore easily obtained. Satisfying this inequality, known as the Armijo condition, is necessary for the good convergence of the algorithm. If the initial guess $\alpha = 1$ does not satisfy this condition a quadratic line search algorithm [8] is used to find a value $\alpha \in [0, 1]$ which does. Although the details of this algorithm are beyond the scope of this article, it should be noted that in the case $\alpha \neq 1$ additional evaluations of the objective function, and therefore of the flux, are needed.

Since the ultimate goal of the scheme is to reduce the residual R , we are not particularly concerned with the accuracy of the solutions to the intermediate problem $\tilde{R}(\bar{U}^{n+1}) = 0$. Therefore only a single iteration of the line search algorithm is applied per timestep. As long as the residual is sufficiently well behaved for the full Newton step to satisfy the Armijo condition the result of this algorithm is the same as the backward-Euler time advance scheme with only a minimal additional expense of verifying that the sufficient decrease condition is satisfied. On the other hand, when the line search determines that backtracking is necessary the additional flux evaluations involved are a reasonable cost for the improved stability of the resulting modified step.

3.3.3 Selecting the Timestep

3.3.3.1 Global timestepping

In a nonlinear sense, the timestep serves to improve the convexity of the modified residual in Equation 3.12 so that the Newton method can be applied.

3.3. Globalized Newton Method

In a linear sense it modifies the left-hand side of Equation 3.13 by making it more diagonally dominant. These modifications are most important when the current solution is far from steady state and therefore the linearized residual at the current state does not accurately reflect the behavior of the residual at the steady state solution. Therefore, since the linearization is expected to improve as the solution converges, it is common to use the norm of the residual to control the size of the timestep. The most common algorithm, named successive evolution-relaxation (SER) [13], is defined as

$$\tau^{n+1} = c_2 \cdot \tau^n \cdot \frac{\|R(\bar{U}^{n-1})\|_2}{\|R(\bar{U}^n)\|_2} \quad (3.15)$$

where τ is a scalar timestep. For global timestepping we set $\Delta t_i = \tau$ for all control volumes. Although this method has been shown to be effective, it requires careful tuning of the initial timestep and the parameter c_2 to prevent the solution process from diverging. Conversely, if c_2 or the initial timestep are taken to be too small the number of iterations needed to converge may be unnecessarily large. The robustness of this approach can be improved by reducing the timestep when a solution update would result in unphysical flow properties [11]. Specifically, when we consider SER in the present work, we reduce the timestep by a factor of two if the solution update would result in the reconstructed values of density or pressure becoming negative at any of the Gauss integration points.

The selection of α satisfying the condition in Equation 3.14 is a measure of how “well behaved”, in the sense of convexity, the modified residual of Equation 3.12 is. Since the use of a sufficiently small timestep has the effect of improving the behaviour of the modified residual function, α can be used to determine the quality of the previous timestep. Therefore we propose to use the result of the line search to help control the timestep such that

$$\tau^{n+1} = \tilde{c}_2 \cdot \tau^n \cdot \frac{\|R(\bar{U}^{n-1})\|_2}{\|R(\bar{U}^n)\|_2} \quad (3.16)$$

where

$$\tilde{c}_2(\alpha) = \begin{cases} c_2, & \text{if } \alpha = 1 \\ \max(\alpha, 0.5), & \text{if } \alpha \neq 1 \end{cases}$$

This modification essentially detects when the timestep has been increased too rapidly and decreases the size of the subsequent timestep. Since this feedback loop prevents the timestep from growing too quickly it is possible to be more aggressive with the choice of c_2 than with the standard SER method.

3.3.3.2 Local Timestepping

Since we do not seek a time-accurate solution, it is not necessary to use a scalar timestep. The concept of local timestepping is rooted in the stability of explicit schemes. These schemes become unstable for CFL numbers larger than one. Although this condition is not binding for implicit schemes, the CFL number is nonetheless commonly used to select a local timestep. Specifically, it is common to derive a local timestep from the global timestep such that

$$\Delta t_i^{n+1} = \tau^{n+1} \cdot \frac{\text{speed}_i^n}{\Delta x_i} \quad (3.17)$$

where speed_i^n is the largest characteristic speed in the control volume, Δx_i a characteristic size, and τ^{n+1} is calculated using Equation 3.15. Alternatively, a global timestepping scheme can be obtained by selecting the smallest timestep, computed using Equation 3.17, amongst all control volumes.

In the present work we propose an alternative means of computing a local timestep. Since the residual was an effective guide for selecting the global timestep in Equation 3.15, we propose to use the local residual to guide the selection of the local timestep. Specifically,

$$\begin{aligned} \Delta t_i^{n+1} &= \zeta_i^{n+1} \tau^{n+1} \cdot \frac{\text{speed}_i^n}{\Delta x_i} \\ \zeta_i^{n+1} &= \max \left(\min \left(\frac{\|R(\bar{U}^n)\|_2}{\|R_i(\bar{U}^n)\|_2}, 10^2 \right), 10^{-2} \right) \end{aligned} \quad (3.18)$$

where $R_i(\bar{U}^n)$ is the residual in control volume i in conserved variables, and τ^{n+1} is evaluated using Equation 3.16. In other words, if the flux in the control volume was small in the previous timestep we expect the local linearization of the residual to be a good representation and a large timestep can be taken. The intended result is that small timesteps are taken in regions of high non-linearity while larger timesteps are taken elsewhere. The goal is to give the scheme some multigrid-like properties such that error waves can be rapidly propagated away from areas where the solution is changing rapidly. For example, it is expected that this scheme will result in small timesteps near an evolving shock while allowing larger timesteps to be taken elsewhere in the domain.

3.4 Preconditioned GMRES

Solving the sparse linear system in Equation 3.13 exactly is prohibitively expensive in computational time and memory. Therefore, as is common, we solve the system approximately using the GMRES [21] method. Solving a linear system with GMRES only requires a means of computing the product of the left-hand-side matrix with an arbitrary vector. Therefore it is common to avoid forming the exact Jacobian matrix and instead approximate the matrix-vector product using Frechet derivative. However, since the convergence properties of GMRES are highly sensitive to the conditioning of the matrix, the system must be preconditioned using an approximate factorization of a simplified Jacobian matrix. This matrix is usually chosen to be the Jacobian of the first-order scheme or an approximation to it. In the present work we consider both this “matrix-free” method and a method which explicitly forms the exact high-order Jacobian.

3.4.1 Matrix-Free Method

In the matrix-free method the matrix-vector products are approximated using a Frechet derivative

$$\frac{\partial R}{\partial \bar{U}} a = \frac{R(\bar{U} + h \cdot a) - R(\bar{U})}{h}$$

where h is a small value. This scheme requires one flux evaluation per inner GMRES iteration plus one flux evaluation per outer iteration.

3.4.2 Forming the High-Order Jacobian Matrix

Alternatively, the exact Jacobian of the high-order scheme can be formed explicitly. The explicit Jacobian can then be used as an improved preconditioner and to avoid the use of Frechet derivatives and their associated residual evaluations.

3.4.2.1 Without Limiters

The analytic Jacobian can be represented explicitly as

$$\frac{\partial R}{\partial \bar{U}} \equiv \frac{\partial FluxInt}{\partial CVars} = \frac{\partial FluxInt}{\partial Flux} \frac{\partial Flux}{\partial RecSol} \frac{\partial RecSol}{\partial RecCoef} \frac{\partial RecCoef}{\partial PVars} \frac{\partial PVars}{\partial CVars}$$

where $FluxInt$ is the flux integral, $Flux$ are the numerical fluxes, $RecSol$ are the reconstructed solutions at Gauss points, $RecCoef$ are the reconstruction coefficients, $PVars$ are the control volume averages of the primitive variables, and $CVars$ are the control volume averages of the conserved variables. To compute the Jacobian, the following procedure is used at each timestep:

1. $\frac{\partial PVars}{\partial CVars}$ is computed for each control volume and stored. This is the standard Jacobian for the change of variables from $(\rho \ \rho u \ \rho v \ E)^T$ to $(\rho \ u \ v \ P)^T$.
2. For each Gauss point, do the following for each of the two adjacent control volumes:

- (a) $\frac{\partial RecSol}{\partial PVars} = \frac{\partial RecSol}{\partial RecCoef} \frac{\partial RecCoef}{\partial PVars}$ is computed. The $\frac{\partial RecCoef}{\partial PVars}$ term is simply the pseudoinverse of the reconstruction matrix precomputed in Equation 3.9, while the $\frac{\partial RecSol}{\partial RecCoef}$ term is a geometric term that depends on the location of the Gauss quadrature point. Although the entire $\frac{\partial RecSol}{\partial PVars}$ term is purely geometric and could be precomputed and stored, this would dramatically increase the memory requirements since, unlike $\frac{\partial RecCoef}{\partial PVars}$, this term is unique for each Gauss point of every control volume. Since the computational efficiency gains would be small, this increase in memory is not worthwhile.
- (b) $\frac{\partial Flux}{\partial RecSol}$, the Jacobian of the Roe flux, is computed. Due to its complexity, the code required to compute this term was automatically generated using the ADIC [4] automatic differentiation tool.
- (c) The product $\frac{\partial Flux}{\partial PVars} = \frac{\partial Flux}{\partial RecSol} \frac{\partial RecSol}{\partial PVars}$ is computed efficiently by taking advantage of the sparsity of the reconstruction terms that is due to the lack of coupling between solution variables. $\frac{\partial Flux}{\partial PVars}$ couples all solution variables in the reconstruction stencil.
- (d) The product $\frac{\partial Flux}{\partial CVars} = \frac{\partial Flux}{\partial PVars} \frac{\partial PVars}{\partial CVars}$ is computed. Since $\frac{\partial Flux}{\partial PVars}$ couples all solution variables in the reconstruction stencil, this step is computationally intensive.
- (e) $\frac{\partial FluxInt}{\partial CVars} = \frac{\partial FluxInt}{\partial Flux} \frac{\partial Flux}{\partial CVars}$ is the contribution to the flux integral due to one side of this Gauss point. This component is computed by using the appropriate Gauss integration weight. The result is added to the total flux Jacobian.

The sparse analytic Jacobian is found once for every Newton iteration and used to produce the matrix-vector products needed by the GMRES solver. These products require fewer operations to compute than the flux evaluation needed by the matrix-free method. They are also easy to optimize to fully benefit from caching and vector operations that may be available on the processor.

3.4.2.2 With Limiters

For transonic and supersonic flows, a limiter must be used to eliminate overshoots in the solution. We will describe a means of including the effect of our limiter, described in Chapter 2, which has been shown to maintain high-order accuracy in smooth regions. The procedure using another limiter, such as the Venkatakrishnan limiter [23], would be the same. The limiting procedure consists in computing the maximum limiting factor $\Phi \in [0, 1]$ which eliminates overshoots. The limiter is then applied as:

$$RecCoef = \begin{bmatrix} 1 & \Psi\bar{x} & \Psi\bar{y} & \Psi\bar{x}^2 & \Psi\bar{x}\bar{y}_i & \Psi\bar{y}^2_i & \dots \\ & \Phi & & & & & \dots \\ & & \Phi & & & & \dots \\ & & & \Phi & & & \dots \\ & & & & \Phi & & \dots \\ & & & & & \Phi & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_{unlmt} \quad (3.19)$$

where $unlmt$ represents the coefficients found by using unlimited reconstruction and $\Psi \equiv \Phi - 1$.

The Jacobian is therefore computed exactly as for the case without limiters except for the the $\frac{\partial RecCoef}{\partial PVars}$ term which is computed as:

$$\frac{\partial RecCoef}{\partial PVars} = \frac{\partial RecCoef}{\partial \Phi} \frac{\partial \Phi}{\partial PVars} + \frac{\partial RecCoef}{\partial RecCoef_{unlmt}} \frac{\partial RecCoef_{unlmt}}{\partial PVars}$$

where $RecCoef_{unlmt}$ are the coefficients of the unlimited reconstruction. To balance computational time and memory use, the $\frac{\partial \Phi}{\partial PVars}$ term is computed and stored once for each flow variable in each control volume in the same way as $\frac{\partial PVars}{\partial CVars}$ was for the unlimited case. The computation of the remaining terms and the assembly of $\frac{\partial RecCoef}{\partial PVars}$ is performed “on the fly” while iterating over Gauss points:

- $\frac{\partial RecCoef}{\partial \Phi}$ term is easily found based on the linear relationship in Equation 3.19.

- $\frac{\partial RecCoef}{\partial RecCoef_{unlmt}}$ term is also evident from Equation 3.19.
- $\frac{\partial RecCoef_{unlmt}}{\partial PVars}$ is the pseudoinverse of the reconstruction matrix pre-computed in Equation 3.9.

The limiter value Φ computed using our scheme or that of Venkatakrishnan, requires the control volume average value, the neighboring maximum and minimum control volume average values, and the unlimited reconstructed value of the solution at the Gauss points. During the limiting procedure, the minimum value of Φ from all the Gauss points is selected. For the purposes of the Jacobian, the choice of which Gauss point produced the smallest value of Φ is “frozen”. Similarly, the control volume which contains the minimum or maximum control volume average is also preselected. The $\frac{\partial \Phi}{\partial PVars}$ term can therefore be decomposed as:

$$\begin{aligned} \frac{\partial \Phi}{\partial PVars} = & \frac{\partial \Phi}{\partial CVave} \frac{\partial CVave}{\partial PVars} + \frac{\partial \Phi}{\partial NeighMin} \frac{\partial NeighMin}{\partial PVars} \\ & + \frac{\partial \Phi}{\partial NeighMax} \frac{\partial NeighMax}{\partial PVars} + \frac{\partial \Phi}{\partial RecVal} \frac{\partial RecVal}{\partial PVars} \end{aligned}$$

where $CVave$, $NeighMin$, and $NeighMax$ are simply select components of $PVar$ and $RecVal$ is the unlimited reconstructed value of the solution at the Gauss point producing the smallest value of Φ . The components $\frac{\partial \Phi}{\partial CVave}$, $\frac{\partial \Phi}{\partial NeighMin}$, $\frac{\partial \Phi}{\partial NeighMax}$, and $\frac{\partial \Phi}{\partial RecVal}$ are obtained by computing the derivative of the limiter function. The component $\frac{\partial RecVal}{\partial PVars}$ is one row of $\frac{\partial RecSol}{\partial RecCoef_{unlmt}} \frac{\partial RecCoef_{unlmt}}{\partial PVars}$ and is found as discussed in the previous section.

3.4.3 Preconditioning

Since the rate of convergence of the GMRES method is strongly dependent on the condition number of the matrix, preconditioning is used to alter the spectrum and hence accelerate the convergence rate of the iterative technique. Left preconditioning is applied by modifying the linear system to be solved

such that

$$Ax = b$$

becomes

$$M^{-1}Ax = M^{-1}b$$

where M^{-1} is an approximate inverse of the preconditioning matrix $M \approx A$. A common approach [3, 19] is to use the flux Jacobian of the first-order scheme for M and to use ILU decomposition to form the approximate inverse. This approach has the advantages of being easier to compute and requiring less memory than using the full-order accurate Jacobian. To form the preconditioning matrix M , a procedure similar to that presented in Subsection 3.4.2 is used except that the terms $\frac{\partial Flux}{\partial RecSol} \frac{\partial RecSol}{\partial RecCoe}$ are eliminated. This method is used for the matrix-free results presented in the present work.

Since the high-order Jacobian already needs to be computed in the matrix-explicit method, its ILU decomposition can easily be used as a preconditioner. The increase in memory use can be partially mitigated by using a lower level of fill; as we will show, our results demonstrate that even with low levels of fill, the matrix-explicit method is much better conditioned than the matrix-free method.

3.4.4 Startup

Since Newton-like quadratic convergence cannot be expected during initial timestepping, computing the full-order Jacobian or using a matrix-free method is unnecessarily expensive at that stage. Instead, the first-order Jacobian is used on the left-hand side and the full-order residual is used on the right-hand side. In the present work, we use timestepping with the first-order Jacobian for the first 10 and 50 iterations for subsonic and transonic cases respectively. We then switch to using either matrix-free GMRES or full-order matrix-explicit GMRES.

3.5 Results

3.5.1 Test Cases

Two two-dimensional inviscid flows will be used to evaluate the performance of the proposed scheme. First, subsonic flow at Mach 0.2 around a multi-element airfoil with zero angle of attack is considered. The mesh consists of 4913 vertices and is shown in Figure 3.1. The pressure from the steady-state solution using the second- and fourth-order schemes is shown in Figure 3.2; the lower dissipation of the fourth-order solution is evident from the larger extent of peak high and low pressure areas.

Second, transonic flow around a NACA 0012 airfoil at Mach 0.8 and an angle of attack of 1.25 degrees is also considered. To prevent oscillations in the solution the limiter described in Chapter 2 is used. The mesh consists of 4656 vertices and is shown in Figure 3.3. The pressure along the upper and lower airfoil surfaces is plotted in Figure 3.4. The benefit of the high-order scheme is evident from the sharper shock on the lower surface.

To study the effects of the various methods and parameters discussed in Sections 3.3 and 3.4 we will use what we propose as the best method as a base scheme and compare it with variations in each aspect of the solution strategy. Therefore, unless otherwise noted, the globalization scheme used is a line search method with a local residual-dependent timestep ($c_2 = 1.1$). The initial timestep is based on a CFL number of 0.1. The startup phase consists of 10 iterations for the subsonic case and 50 iterations for the transonic case. During this phase, the Jacobian of the residual is approximated by that of the first-order scheme for both preconditioning and the matrix-vector products needed in the GMRES method. Preconditioning during the startup phase is always achieved using ILU with zero levels of fill (ILU(0)). During the second phase, the full-order exact Jacobian is formed and used for preconditioning and matrix-explicit GMRES. Preconditioning is achieved using ILU(0). The linear system at each iteration is solved to a relative residual drop of 10^{-3} .

3.5.2 Globalization Strategy

3.5.2.1 Line Search

To study the effects of adding the line search step to each timestep, we compare it to a more standard backward-Euler approach. Specifically, we compare the line search algorithm described in Subsection 3.3.2 using the timestep evolution in Equation 3.16 with the backward-Euler approach of Equation 3.13 using the standard SER timestep control of Equation 3.15. In both cases the local timestep is derived from these values using the local residual as in Equation 3.18. The results are presented in the form of residual convergence plots in Figures 3.5 and 3.6. For the backward-Euler method without line search we use values of c_2 of 1.0 and 1.1. Since the line search method is expected to be more robust to larger values of c_2 we use values of 1.1 and 1.2. The results indicate that there is little difference in the number of iterations required for convergence for the subsonic flow. However, for transonic flow, the method without line search can fail to converge. Specifically it can enter a cycle where the solution alternates between two states. The line search algorithm provides the necessary robustness to avoid this pitfall. The results also illustrate that the robustness of the line search remains unaffected by the choice of parameter c_2 .

3.5.2.2 Local Timestep

Next, we consider three methods of deriving the local timesteps Δt_i from the global timestep τ computed using our modified SER method in Equation 3.16. The simplest method is to use a global timestep. The second method is to compute a local timestep using the local CFL condition for each control volume. Finally, the local residual based method of Equation 3.18 is also considered. The resulting convergence plots are shown in Figures 3.7 and 3.8. In the subsonic case, the use of the local residual based timestep results in more rapid convergence than with either CFL based local timestepping or global timestepping. For transonic flow, this trend is even more pronounced, as the non-residual-based methods fail to converge within 350 iterations.

Since it is impossible to study all permutations of parameters these results should only be viewed as a representative sample. With the careful selection of startup CFL number and other hand tuning of the startup procedure, convergence is often possible with the methods that failed to converge in these examples. However the gains in robustness and performance demonstrated from the use of the line search method and of residual based local timestep are representative of our general experience.

3.5.3 Solving the Linear System

3.5.3.1 Cost of Jacobian Evaluation

Having determined how to minimize the number of linear systems to solve to converge the nonlinear residual, we now turn our attention to the efficiency of the linear solver itself. We begin by comparing the relative CPU time needed to compute a flux integral, a first-order Jacobian, and a high-order Jacobian. The results are given in Tables 3.1 and 3.2. The large increase in computational time needed for flux and Jacobian evaluations for the third-order scheme relative to the second-order scheme can in large part be attributed to the doubling of the number of Gauss points. Since the fourth-order scheme uses the same number of Gauss points as the third-order scheme, a smaller increase in flux and Jacobian evaluation times is observed. For flux evaluations, this increase is purely due to the cost of the reconstruction procedure. For the Jacobian, the increased cost of matrix-matrix products needed for its assembly is also important.

3.5.3.2 Quality of Preconditioning

Next, we compare the relative effectiveness of the ILU decomposition of the first-order and high-order Jacobians. The average number of inner GMRES iterations needed per Newton iteration to obtain a relative residual drop of 10^{-3} is shown in Table 3.3. The results also include the case where the full LU decomposition of the first-order Jacobian is used to precondition the high-order matrix-free scheme. Since doing the full LU decomposition is prohibitively expensive in both time and memory, this is included solely to

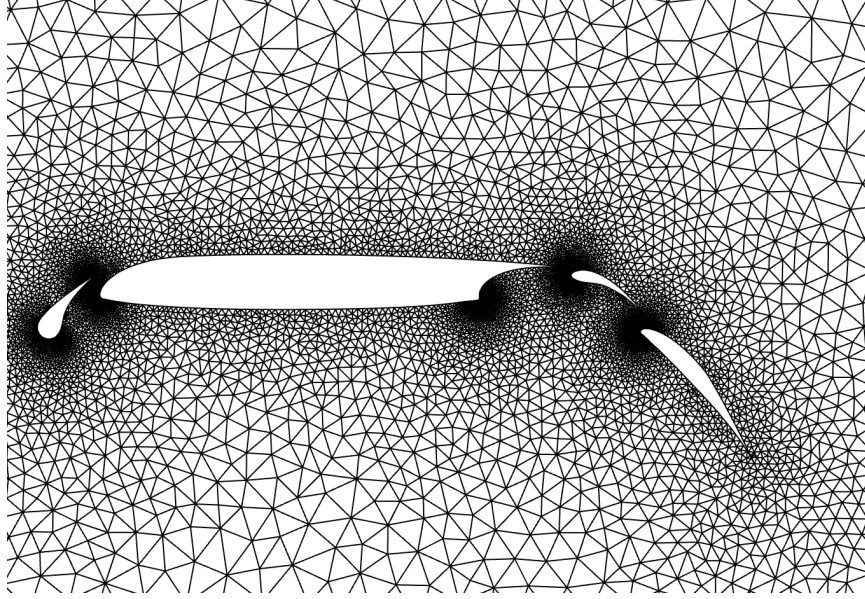


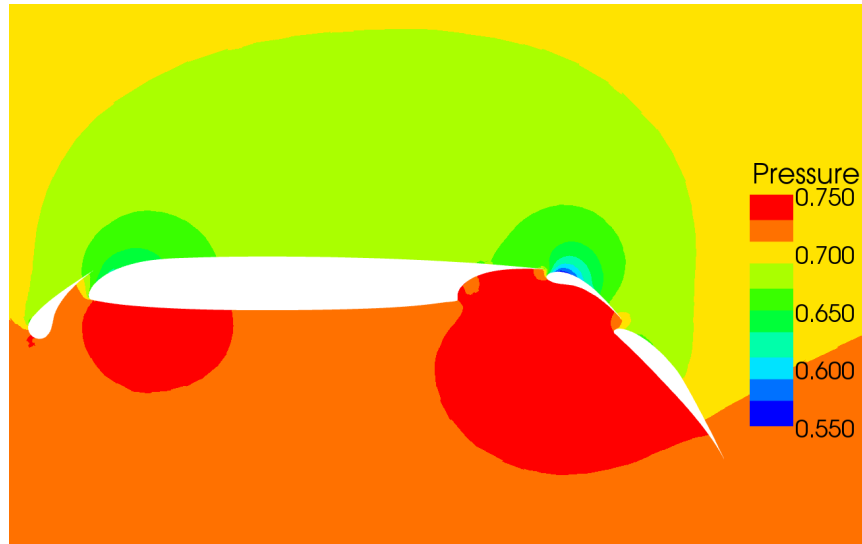
Figure 3.1: Multi-element airfoil mesh consisting of 4913 vertices

Order	Flux	1st Order Jacobian		Full Order Jacobian	
	seconds	seconds	relative to flux	seconds	relative to flux
1	0.0116	0.217	18.7		
2	0.0194	0.217	11.2	0.346	17.8
3	0.0399	0.217	5.4	1.122	28.1
4	0.0568	0.217	3.8	1.219	21.5

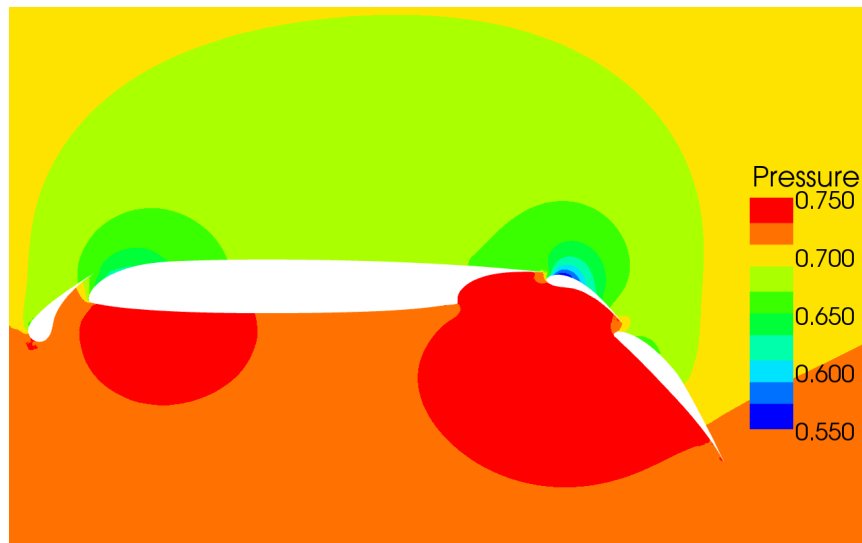
Table 3.1: Relative computational time of flux and Jacobian evaluation without limiter for subsonic case

Order	Flux	1st Order Jacobian		Full Order Jacobian	
	seconds	seconds	relative to flux	seconds	relative to flux
1	0.0109	0.205	18.8		
2	0.0356	0.205	5.8	0.375	10.5
3	0.0733	0.205	2.8	1.623	22.1
4	0.1137	0.205	1.8	2.241	19.7

Table 3.2: Relative computational time of flux and Jacobian evaluation with limiter for transonic case



(a) Second-order scheme



(b) Fourth-order scheme

Figure 3.2: Pressure contours for subsonic flow over a multi-element airfoil

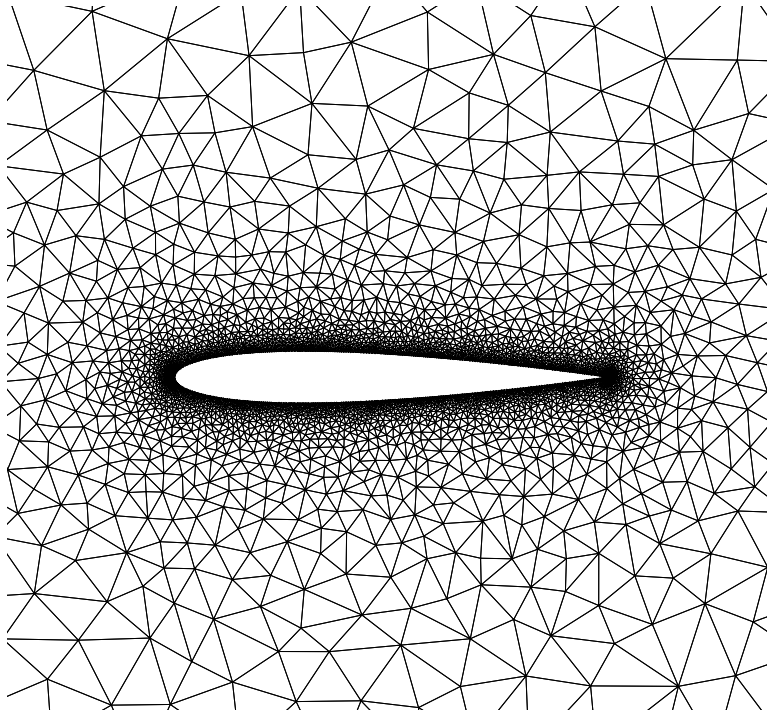


Figure 3.3: NACA 0012 airfoil mesh consisting of 4656 vertices

demonstrate the upper bounds on the quality of preconditioning that can be achieved when using the first-order Jacobian. The results indicate that the first-order Jacobian is a reasonably good preconditioner for the second-order scheme if ILU with enough levels of fill is used. However, this preconditioner does a poor job for the third- and fourth-order schemes. Even with a full LU decomposition, the number of GMRES iterations required to solve the fourth-order case remains high. Since the matrix-free method must be used when the high-order Jacobian is not available, this large number of GMRES iterations results in a large number of residual evaluations. The costs of these residual evaluations exceed the relative additional cost of computing the high-order Jacobian. Using the full-order Jacobian for the preconditioner, the convergence properties of the high-order schemes is comparable to that of the second-order scheme. In all cases the transonic test case requires fewer GMRES iterations per Newton iteration indicating that the linear systems are better conditioned. Due to the high nonlinearity of the shock, the adaptive timestep method yields smaller timesteps for the transonic case than the subsonic case. This likely explains why the linear system is better conditioned in the transonic case. However, since these smaller timesteps also result in a larger number of Newton iterations, the total number of GMRES iterations for the transonic case exceeds that of the subsonic case.

3.5.3.3 Overall Memory and Computational Cost

The major components contributing to the memory requirement for both matrix-explicit and matrix-free methods are:

- *The pseudoinverse of the reconstruction matrix.* To avoid solving the least-squares problem at each flux evaluation, these matrices need to be precomputed and stored for each control volume.
- *The Jacobian.* For the matrix-free scheme, this will always be the first-order Jacobian. For the matrix-explicit scheme, the high-order Jacobian will inevitably have more fill and require more storage.
- *ILU decomposition of the Jacobian.* The memory required for this

3.5. Results

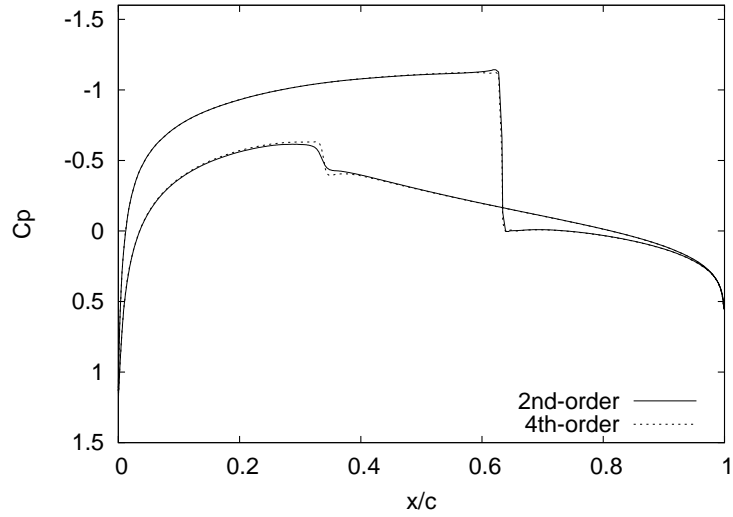


Figure 3.4: Surface pressure coefficient for transonic flow over a NACA 0012 airfoil

Order	First-Order Jacobian (MF)			High-Order Jacobian (ME)	
	ILU(1)	ILU(4)	LU	ILU(0)	ILU(1)
2	85.5	49.7	29.2	54.2	33.7
3	81.3	55.0	38.8	34.0	23.4
4	152.4	105.4	95.7	32.6	23.0

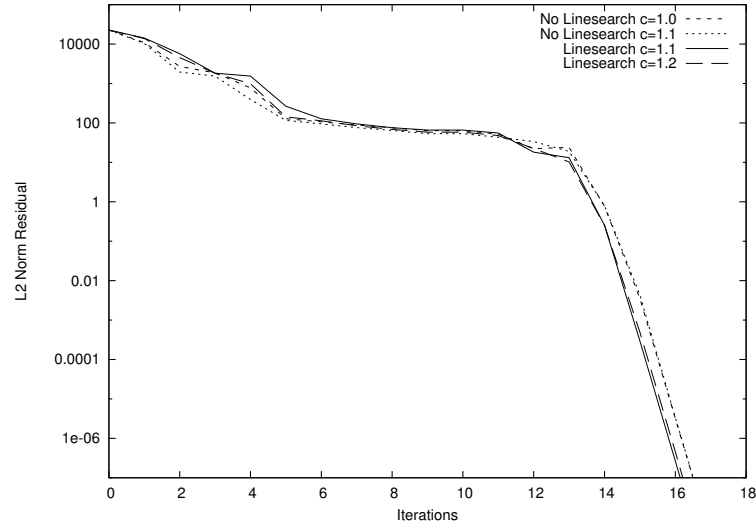
(a) Subsonic case

Order	First-Order Jacobian (MF)			High-Order Jacobian (ME)	
	ILU(1)	ILU(4)	LU	ILU(0)	ILU(1)
2	41.9	28.6	26.8	25.2	13.3
3	40.9	32.9	29.5	10.9	7.1
4	58.9	47.5	44.9	20.6	11.9

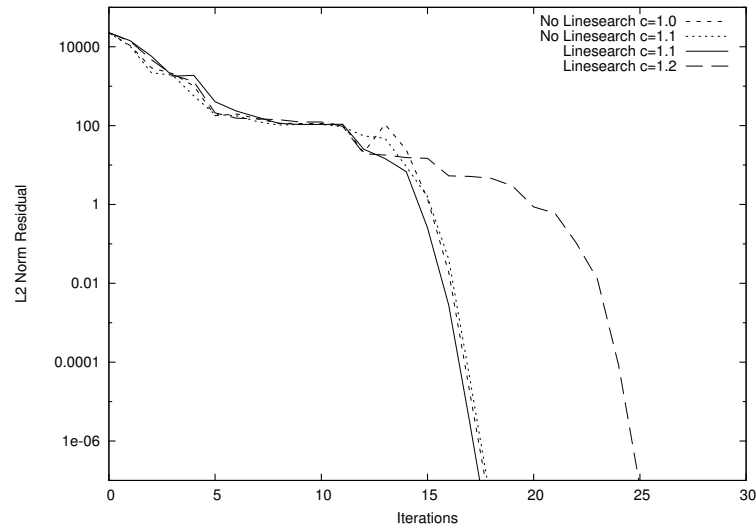
(b) Transonic case

Table 3.3: Average number of inner GMRES iterations per Newton iteration

3.5. Results



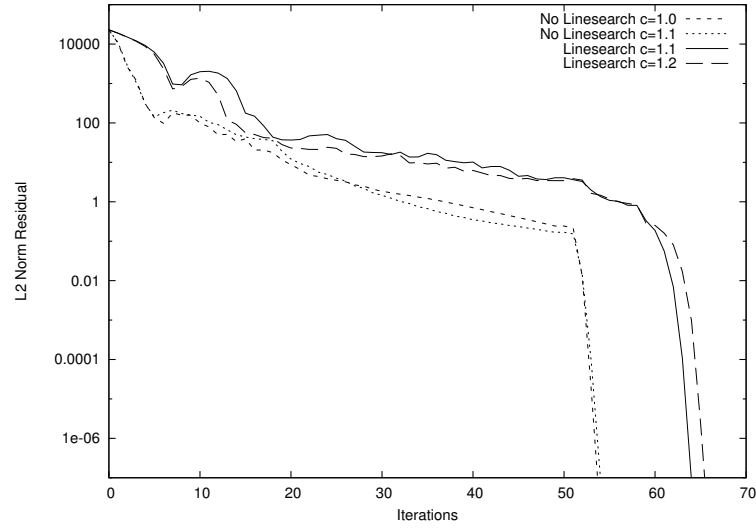
(a) Second-order scheme for subsonic case



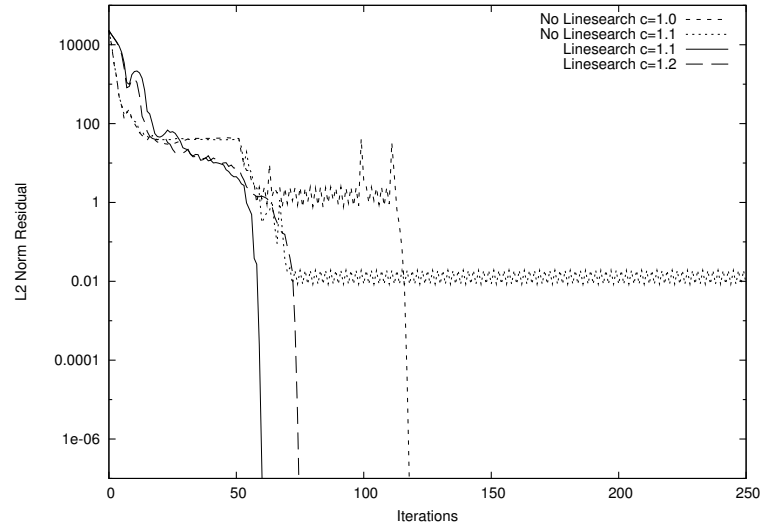
(b) Fourth-order scheme for subsonic case

Figure 3.5: Convergence plots comparing the line search method with a backward-Euler method for subsonic case

3.5. Results



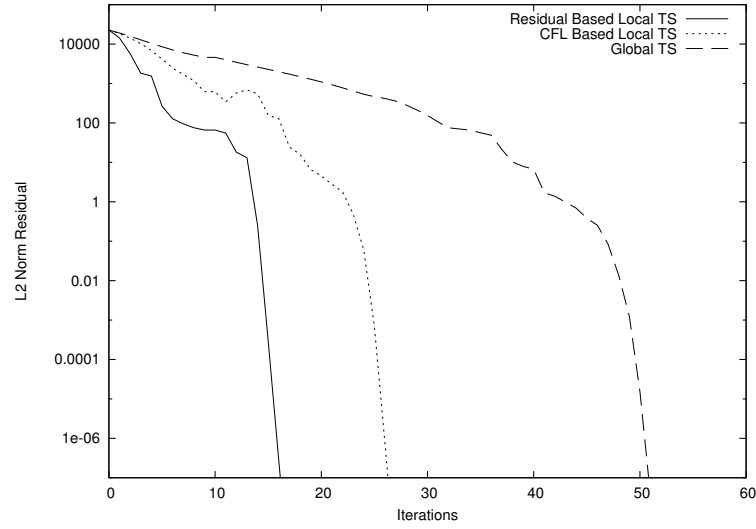
(a) Second-order scheme for transonic case



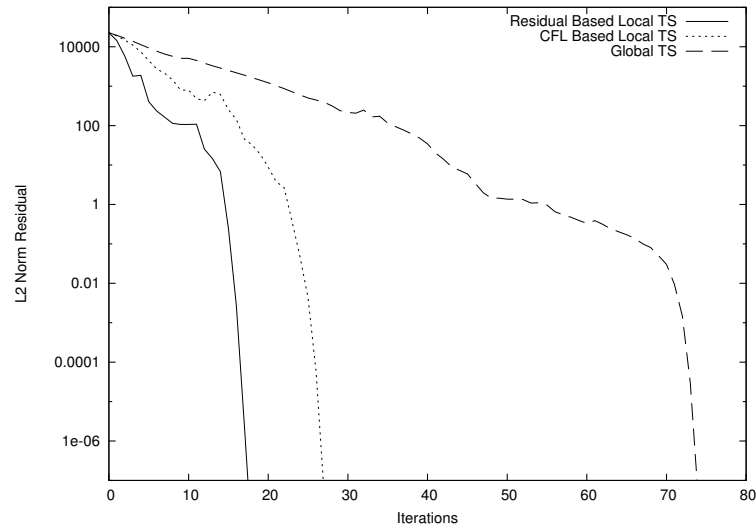
(b) Fourth-order scheme for transonic case

Figure 3.6: Convergence plots comparing the line search method with a backward-Euler method for transonic case

3.5. Results



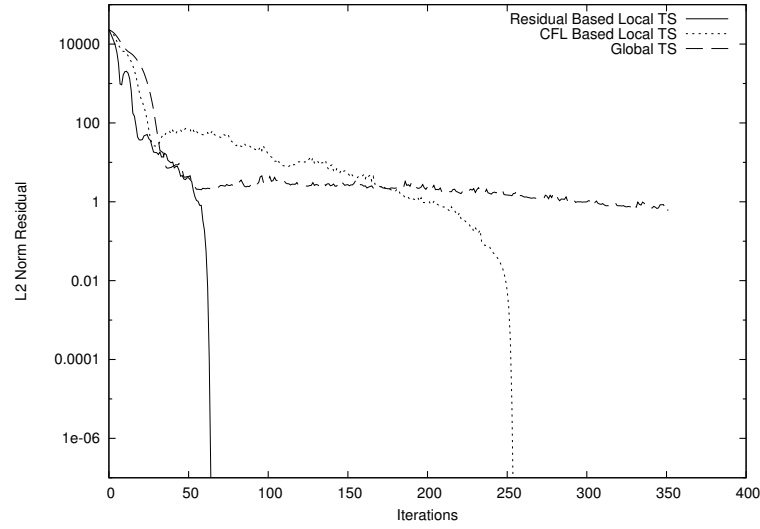
(a) Second-order scheme for subsonic case



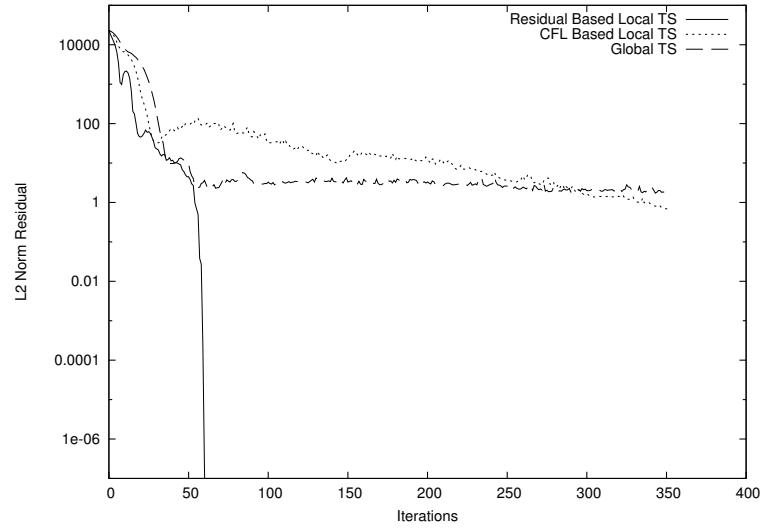
(b) Fourth-order scheme for subsonic case

Figure 3.7: Convergence plots comparing global timestepping, CFL-based local timestepping and residual-based local timestepping for subsonic case

3.5. Results



(a) Second-order scheme for transonic case



(b) Fourth-order scheme for transonic case

Figure 3.8: Convergence plots comparing global timestepping, CFL-based local timestepping and residual-based local timestepping for transonic case

depends not only on the fill of the Jacobian but also on the additional fill due to the decomposition which increases with n for ILU(n).

- *Krylov subspace.* The maximum number of inner GMRES iterations required to solve a Newton iteration and the number of solution unknowns per control volume determines the memory requirement of the Krylov solver.

The breakdown of memory requirement along with the total CPU time is shown in Tables 3.4 and 3.5. The additional memory required by the matrix-explicit scheme is due to the increased fill of the Jacobian and resulting preconditioning matrix. However, this is partially offset by the lower fill ratio of the ILU decomposition and the reduced memory use of the Krylov solver. For the subsonic case, the matrix-explicit method preconditioned with ILU(0) requires less than 50% more memory than the matrix-free method with ILU(4) for second- and fourth-order schemes while for the third-order scheme the difference is about 100% more memory. For the transonic case, the matrix-explicit second-order method requires 31% more memory than the matrix-free method, while for the third- and fourth-order scheme, the difference is closer to 100%.

Order	Scheme	Precon- ditioner	Recon- -struction	Jacobian	ILU Matrix	Krylov Subsp.	Total (floats)	Total (bytes)	Time (seconds)	Time (res eval)
2	ME	ILU(0)	11.7	299.2	299.2	252	862.1	6897	10.2	526
2	ME	ILU(1)	11.7	299.2	480.3	156	947.2	7577	9.5	490
2	MF	ILU(1)	11.7	109	144.1	488	752.9	6023	21.8	1124
2	MF	ILU(4)	11.7	109	300.4	252	673.2	5385	15.2	784
3	ME	ILU(0)	88	585	585	172	1430	11440	16.7	419
3	ME	ILU(1)	88	585	1011.7	124	1808.6	14469	15.7	393
3	MF	ILU(1)	88	109	144.1	476	817.2	6537	35	877
3	MF	ILU(4)	88	109	300.4	316	813.5	6508	26.2	657
4	ME	ILU(0)	179.2	615.2	615.2	164	1573.7	12589	19.7	347
4	ME	ILU(1)	179.2	615.2	1087.6	108	1990	15920	20.9	368
4	MF	ILU(1)	179.2	109	144.1	808	1240.4	9923	92.1	1621
4	MF	ILU(4)	179.2	109	300.4	572	1160.7	9285	66.5	1171

Table 3.4: Memory in number of floating point numbers per control volume along with run time in seconds and equivalent residual evaluations for subsonic case

Order	Scheme	Precon- ditioner	Recon- struction	Jacobian	ILU Matrix	Krylov Subsp.	Total (floats)	Total (bytes)	Time (seconds)	Time (res eval)
2	ME	ILU(0)	11.7	297.2	297.2	120	726.1	5809	27.5	772
2	ME	ILU(1)	11.7	297.2	470.2	72	851.1	6809	27.1	761
2	MF	ILU(1)	11.7	108.7	142.6	216	479	3832	47.2	1326
2	MF	ILU(4)	11.7	108.7	291.7	144	556.1	4449	40.2	1129
3	ME	ILU(0)	88	580.2	580.2	116	1364.3	10915	184.5	2517
3	ME	ILU(1)	88	580.2	986	52	1706.2	13649	156.4	2134
3	MF	ILU(1)	88	108.7	142.6	268	607.3	4858	179.1	2443
3	MF	ILU(4)	88	108.7	291.7	220	708.4	5668	155.9	2127
4	ME	ILU(0)	179.2	616.7	616.7	100	1512.6	12101	53.5	471
4	ME	ILU(1)	179.2	616.7	1073.4	56	1925.3	15403	54.8	482
4	MF	ILU(1)	179.2	108.7	142.6	316	746.5	5972	101.6	894
4	MF	ILU(4)	179.2	108.7	291.7	256	835.6	6685	88.1	775

Table 3.5: Memory use in number of floating point numbers per control volume along with run time in seconds and equivalent residual evaluations for transonic case

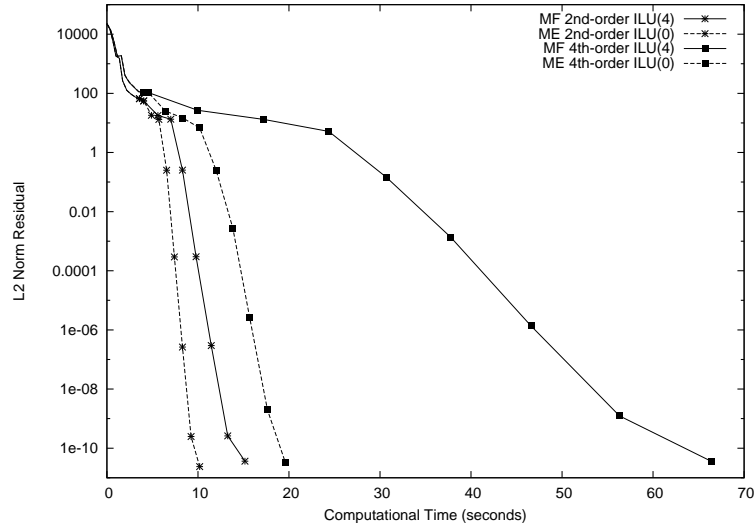
Next we consider the overall computational time required to converge to the solution. The results presented in Tables 3.4 and 3.5 present both the CPU time in seconds on a single core of an Intel Core 2 processor as well as a multiple of the cost of a residual evaluation. Despite the fact that no attempt to hand tune the parameters of the globalization strategy was made, the runtime in terms of equivalent residual evaluations is competitive with that reported elsewhere. In particular, Blanco and Zingg report [5] that the Newton-Krylov algorithm applied to a second-order accurate matrix-dissipation scheme required the equivalent of 660 residual evaluation to converge for the same transonic case. However due to the differences in the discretization scheme and the mesh used, too much emphasis should not be placed on such direct comparisons.

The results indicate that in terms of the memory to computational time trade-off the matrix-explicit method with ILU(0) preconditioning and the matrix-free method with ILU(4) preconditioning represent, in most cases, the best trade-offs. Figure 3.9 shows the residual as a function of time for these two schemes for the second- and fourth-order schemes. On these plots the startup procedure is shown with a line while the main stage is shown with a line and a point marker at each iteration. In all cases the matrix-explicit method outperforms the matrix-free method. The difference is most substantial for the subsonic case where the startup phase represents a smaller portion of the total time and where the conditioning of the linear system is poorer. In this case, the matrix-explicit method provides a 38% and 70% reduction in CPU time for the second- and fourth-order schemes respectively. For the transonic case, the matrix-explicit method provides a 32% and 39% reduction in CPU time for second- and fourth-order schemes respectively.

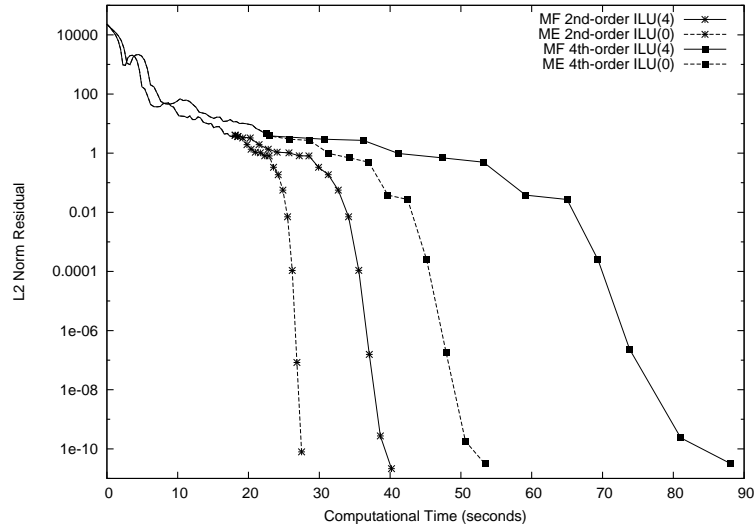
3.6 Conclusion

We have shown that the most commonly used globalization technique may fail to converge, particularly for difficult cases such as the fourth-order solution of transonic flow. This was successfully remedied by the addition of a line search

3.6. Conclusion



(a) Subsonic test case



(b) Transonic test case

Figure 3.9: Residual versus computational time for matrix-explicit method and matrix-free method

3.6. Conclusion

method and the use of the result of the line search to adapt the timestep for the next iteration. A new method of computing the local-timestep from the local residual was shown to dramatically improve robustness and performance of both the second- and fourth-order schemes relative to the use of a global timestep or a local timestep based on the CFL number.

A method for efficiently computing the exact Jacobian of the high-order scheme was devised. The incomplete factorization of this Jacobian was shown to be a more effective preconditioner than the factorization of the first-order Jacobian. The difference is particularly important for the fourth-order scheme. In addition the use of this exact Jacobian makes the use of Frechet derivatives and their associated residual evaluations unnecessary in the GMRES method. These factors lead to a reduction in computational time needed for convergence relative to the matrix-free scheme. The difference is particularly important for the fourth-order scheme where a 70% and 39% reduction of computational time is achieved for the subsonic and transonic cases respectively. These reductions in computational time come at the cost of increased memory consumption of 36% and 81% for subsonic and transonic cases respectively.

Although these proposed methods help reduce the computational time of both the second- and fourth-order schemes, they have a greater effect on the latter. This helps reduce the gap between the computational effort needed for the fourth-order scheme and the second-order scheme on the same grid thereby making the high-order schemes more attractive. With the use of the proposed globalization techniques and the matrix-explicit method, the fourth-order scheme is only a factor of 1.9 more expensive than the second-order scheme. When compared to the matrix-free second-order scheme, the matrix-explicit fourth-order scheme requires only a factor of 1.1 to 1.3 more computational time and about three times as much memory. Therefore the fourth-order matrix-explicit method is less expensive in time and memory than the second-order matrix-free scheme applied to a mesh with one additional 4 : 1 refinement step.

3.7 Bibliography

- [1] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. In *Lecture Series 1994-05*, Rhode-Saint-Genèse, Belgium, Mar. 1994. von Karman Institute for Fluid Dynamics.
- [2] T. J. Barth and P. O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, Jan. 1990.
- [3] T. J. Barth and S. W. Linton. An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. AIAA paper 95-0221, Jan. 1995.
- [4] C. Bischof, L. Roh, and A. Mauer. ADIC — An extensible automatic differentiation tool for ANSI-C. Preprint ANL/MCS-P626-1196, Argonne National Laboratory, 1996.
- [5] M. Blanco and D. W. Zingg. A Newton-Krylov algorithm with a loosely-coupled turbulence model for aerodynamic flows. AIAA Paper 2006-0691. Presented at the 44th AIAA Aerospace Sciences Meeting, Jan. 2006.
- [6] S. De Rango and D. W. Zingg. Higher-order spatial discretization for turbulent aerodynamic computations. *American Institute of Aeronautics and Astronautics Journal*, 39(7):1296–1304, July 2001.
- [7] M. Delanaye and J. A. Essers. Quadratic-reconstruction finite volume scheme for compressible flows on unstructured adaptive grids. *American Institute of Aeronautics and Astronautics Journal*, 35(4):631–639, Apr. 1997.
- [8] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization*. Prentice-Hall, 1983.
- [9] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1983.

3.7. Bibliography

- [10] W. Gropp, D. Keyes, L. C. McInnes, and M. D. Tidriri. Globalized Newton-Krylov-Schwarz Algorithms and Software for Parallel Implicit CFD. *International Journal of High Performance Computing Applications*, 14(2):102–136, 2000.
- [11] H. Jiang and P. A. Forsyth. Robust linear and nonlinear strategies for solution of the transonic Euler equations. *Computers and Fluids*, 24(7):753 – 770, 1995.
- [12] C. T. Kelley and D. E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, 35:508–523, 1998.
- [13] W. A. Mulder and B. van Leer. Experiments with implicit upwind methods for the Euler equations. *Journal of Computational Physics*, 59:232–246, June 1985.
- [14] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured GMRES algorithm for inviscid compressible flows. In *Proceedings of the Seventeenth AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2005.
- [15] A. Nejat and C. Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. *Journal of Computational Physics*, 227(4):2366–2386, 2008.
- [16] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2592–2609, 2008.
- [17] C. F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.
- [18] R. P. Pawlowski, J. N. Shadid, J. P. Simonis, and H. F. Walker. Globalization techniques for Newton–Krylov methods and applications to the

3.7. Bibliography

- fully coupled solution of the Navier–Stokes equations. *SIAM Review*, 48(4):700–721, 2006.
- [19] A. Pueyo and D. W. Zingg. Improvements to a Newton-Krylov solver for aerodynamic flows. Thirty-sixth AIAA Aerospace Sciences Meeting, Jan. 1998. AIAA Paper 98-0619.
- [20] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [21] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3):856–869, July 1986.
- [22] R. S. Tuminaro, H. F. Walker, and J. N. Shadid. On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *Journal of Computational Physics*, 180(2):549 – 558, 2002.
- [23] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady-state solutions. AIAA paper 93-0880, Jan. 1993.
- [24] D. Zingg, S. De Rango, M. Nemec, and T. Pulliam. Comparison of several spatial discretizations for the Navier-Stokes equations. *Journal of Computational Physics*, 160:683–704, 2000.

Chapter 4

Extension to Viscous Flows[†]

4.1 Introduction

Unstructured grid finite-volume methods have seen wide acceptance in recent times due to their efficient handling of complex geometries. Though third-order schemes for the advective fluxes have been known [2] for almost 20 years, they have seen little use. The number of details which must be properly implemented, such as the correct treatment of curved boundaries, is likely partially to blame. Furthermore, the application of convergence acceleration techniques is also more complicated than for the second-order schemes. Despite this we have recently had success in demonstrating the accuracy of third- and fourth-order schemes for the Euler equations [15]. Just as importantly, we have had success in devising convergence acceleration techniques for these schemes [11, 13, 14]. The result is that we have successfully demonstrated that third- and fourth-order schemes can be more computationally efficient at solving the Euler equations for a given level of accuracy than second-order schemes.

We now aim to extend this work to the Navier-Stokes equations. Though other researchers [3, 5] have previously used third-order discretizations for the advective terms, they used traditional second-order discretizations for the viscous terms. The present work is an extension of work previously carried out [16] demonstrating the fourth-order accurate solution of the advection-diffusion equation. Although the simulation of compressible turbulent flows for aerodynamics applications is our ultimate goal, the present work is

[†]A version of this chapter will be submitted for publication. C. Michalak and C. Ollivier-Gooch. A high-order accurate unstructured finite-volume method for the Navier-Stokes equations.

restricted to subsonic laminar compressible flows.

In Section 4.3 we demonstrate how k -exact reconstruction including boundary conditions can be efficiently carried out. The evaluation of advective and diffusive fluxes using this reconstruction is addressed Section 4.4. The matrix-explicit Newton-Krylov method used to converge the solution to steady state is presented in 4.5. Finally, in Section 4.6, we present numerical experiments demonstrating the accuracy of the method and its computational efficiency.

4.2 Governing Equations

The two-dimensional compressible Navier-Stokes equations for a control volume V_i can be written in integral form as

$$\frac{d}{dt} \int_{V_i} U dA + \oint_{\Omega_i} (F - G) dS = 0 \quad (4.1)$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad F = \begin{bmatrix} \rho u_n \\ \rho u u_n + P \hat{n}_x \\ \rho v u_n + P \hat{n}_y \\ (E + P) u_n \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ \tau_{xx} \hat{n}_x + \tau_{yx} \hat{n}_y \\ \tau_{xy} \hat{n}_x + \tau_{yy} \hat{n}_y \\ (u \tau_{xx} + v \tau_{xy} - q_x) \hat{n}_x + (u \tau_{yx} + v \tau_{yy} - q_y) \hat{n}_y \end{bmatrix}$$

where U is the solution vector in conservative variables, F and G are the inviscid and viscous flux vectors normal to the control volume boundary, \hat{n}_x and \hat{n}_y are the unit normal components, and $u_n = u \hat{n}_x + v \hat{n}_y$. The strain tensor and heat flux vector are

$$\tau = \mu \begin{bmatrix} 2u_x & u_y + v_x \\ v_x + u_y & 2v_y \end{bmatrix} - \frac{2}{3} \mu \begin{bmatrix} u_x + v_y & 0 \\ 0 & u_x + v_y \end{bmatrix}$$

$$q = \frac{1}{(\gamma - 1)} \frac{\mu}{Pr} \nabla T$$

where μ is the viscosity and T is the temperature.

4.3 Solution Reconstruction

The basis of the present work is k -exact least-squares reconstruction [1] which has been demonstrated to achieve high-order accuracy for the solution of the Euler equation [15]. This approach has previously been extended to include the diffusive fluxes of the advection-diffusion equation [16]. Our current approach is similar to this previous work, but includes some additional efficiency considerations, particularly with regards to the handling of boundary condition enforcement. Though the flux is evaluated in the conserved variables $(\rho, \rho u, \rho v, E)$, as is common we reconstruct the primitive variables (ρ, u, v, P) .

4.3.1 Reconstruction for Interior Control Volumes

In the finite-volume method, the domain is tessellated into non-overlapping control volumes. Although our approach is equally applicable to vertex- and cell-centered control volume discretizations, only results using vertex-centered grids will be presented herein. Each control volume V_i has a geometric reference point \vec{x}_i . This reference point is taken as the cell centroid for cell-centered control volumes and the vertex for vertex-centered control volumes. For any smooth function $U(\vec{x})$ and its control volume averaged values \bar{U}_i , the k -exact least-squares reconstruction uses a compact stencil in the neighborhood of control volume i to compute an expansion $U_i^R(\vec{x} - \vec{x}_i)$ that conserves the mean in control volume i and reconstructs exactly polynomials of degree $\leq k$ (equivalently, $U_i^R(\vec{x} - \vec{x}_i) - U(\vec{x}) = \mathcal{O}(\Delta x^{k+1})$).

Conservation of the mean requires that the average of the reconstructed function R_i and the original function U over control volume i be the same:

$$\frac{1}{V_i} \int_{V_i} U_i^R(\vec{x} - \vec{x}_i) dA = \frac{1}{V_i} \int_{V_i} U(\vec{x}) dA \equiv \bar{U}_i. \quad (4.2)$$

4.3. Solution Reconstruction

The expansion $U_i^R(\vec{x} - \vec{x}_i)$ can be written as:

$$\begin{aligned}
U_i^R(\vec{x} - \vec{x}_i) = & U|_{\vec{x}_i} + \frac{\partial U}{\partial x}\bigg|_{\vec{x}_i} (x - x_i) + \frac{\partial U}{\partial y}\bigg|_{\vec{x}_i} (y - y_i) \\
& + \frac{\partial^2 U}{\partial x^2}\bigg|_{\vec{x}_i} \frac{(x - x_i)^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_{\vec{x}_i} ((x - x_i)(y - y_i)) \\
& + \frac{\partial^2 U}{\partial y^2}\bigg|_{\vec{x}_i} \frac{(y - y_i)^2}{2} + \dots
\end{aligned} \tag{4.3}$$

Taking the control volume average of this expansion over control volume i and equating it to the mean value gives

$$\begin{aligned}
\bar{U}_i = & U|_{\vec{x}_i} + \frac{\partial U}{\partial x}\bigg|_{\vec{x}_i} \bar{x}_i + \frac{\partial U}{\partial y}\bigg|_{\vec{x}_i} \bar{y}_i \\
& + \frac{\partial^2 U}{\partial x^2}\bigg|_{\vec{x}_i} \frac{\bar{x}^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_{\vec{x}_i} \bar{xy} + \frac{\partial^2 U}{\partial y^2}\bigg|_{\vec{x}_i} \frac{\bar{y}^2}{2} + \dots
\end{aligned} \tag{4.4}$$

where

$$\bar{x^n y^m}_i \equiv \frac{1}{A_i} \int_{V_i} (x - x_i)^n (y - y_i)^m dA. \tag{4.5}$$

are control volume moments.

k th-order accuracy requires that we compute the $(k - 1)$ th derivatives by minimizing the error in predicting the mean value of the reconstructed function for control volumes in the stencil of neighboring control volumes $\{V_j\}_i$. In other words, we aim to minimizing the difference between the control volume average \bar{U}_j and the average of the reconstruction R_i over control volume j . The minimum number of control volumes in the reconstruction stencil is equal to the number of derivative terms to be approximated. In practice, to improve conditioning, we increase this minimum to 4, 10, and 16 for second-, third-, and fourth-order schemes respectively. The stencil is constructed by first considering the first-neighbors of the control volume. If additional neighbors are needed they are added in order of topological proximity. All neighbors at given topological distance are added at once. The resulting stencil is therefore topologically centered but may exceed the minimum specified size. The mean value, for a single control volume V_j , of

the reconstructed function U_i^R is

$$\begin{aligned}
 \frac{1}{A_j} \quad & * -4ex \int_{V_j} U_i^R(\vec{x} - \vec{x}_i) dA = U|_{\vec{x}_i} \\
 & + \frac{\partial U}{\partial x} \Big|_{\vec{x}_i} \left(\frac{1}{A_j} \int_{V_j} (x - x_i) dA \right) \\
 & + \frac{\partial U}{\partial y} \Big|_{\vec{x}_i} \left(\frac{1}{A_j} \int_{V_j} (y - y_i) dA \right) \\
 & + \frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} \left(\frac{1}{2A_j} \int_{V_j} (x - x_i)^2 dA \right) \\
 & + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} \left(\frac{1}{A_j} \int_{V_j} (x - x_i)(y - y_i) dA \right) \\
 & + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} \left(\frac{1}{2A_j} \int_{V_j} (y - y_i)^2 dA \right) + \dots
 \end{aligned} \tag{4.6}$$

To avoid computing moments of each control volume in $\{V_j\}_i$ about \vec{x}_i , replace $x - x_i$ and $y - y_i$ with $(x - x_j) + (x_j - x_i)$ and $(y - y_j) + (y_j - y_i)$, respectively. Expanding and integrating, we obtain

$$\begin{aligned}
 \frac{1}{A_j} \quad & * -4ex \int_{V_j} U_i^R(\vec{x} - \vec{x}_i) = U|_{\vec{x}_i} + \frac{\partial U}{\partial x} \Big|_{\vec{x}_i} (\bar{x}_j + (x_j - x_i)) \\
 & + \frac{\partial U}{\partial y} \Big|_{\vec{x}_i} (\bar{y}_j + (y_j - y_i)) \\
 & + \frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} \frac{\bar{x}_j^2 + 2\bar{x}_j(x_j - x_i) + (x_j - x_i)^2}{2} \\
 & + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} (\bar{x}_j \bar{y}_j + \bar{x}_j(y_j - y_i) + (x_j - x_i)\bar{y}_j + (x_j - x_i)(y_j - y_i)) \\
 & + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} \frac{\bar{y}_j^2 + 2\bar{y}_j(y_j - y_i) + (y_j - y_i)^2}{2} + \dots
 \end{aligned} \tag{4.7}$$

This equation is written for every control volume in the stencil. These equations combined with the mean constraint form a constrained least-squares system of the form:

4.3. Solution Reconstruction

$$\begin{aligned}
 & \begin{bmatrix} 1 & \bar{x} & \bar{y} & \bar{x^2} & \overline{xy} & \bar{y^2} & \cdots \\ w_{i1} & w_{i1}\widehat{x}_{i1} & w_{i1}\widehat{y}_{i1} & w_{i1}\widehat{x^2}_{i1} & w_{i1}\widehat{xy}_{i1} & w_{i1}\widehat{y^2}_{i1} & \cdots \\ w_{i2} & w_{i2}\widehat{x}_{i2} & w_{i2}\widehat{y}_{i2} & w_{i2}\widehat{x^2}_{i2} & w_{i2}\widehat{xy}_{i2} & w_{i2}\widehat{y^2}_{i2} & \cdots \\ w_{i3} & w_{i3}\widehat{x}_{i3} & w_{i3}\widehat{y}_{i3} & w_{i3}\widehat{x^2}_{i3} & w_{i3}\widehat{xy}_{i3} & w_{i3}\widehat{y^2}_{i3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\widehat{x}_{iN} & w_{iN}\widehat{y}_{iN} & w_{iN}\widehat{x^2}_{iN} & w_{iN}\widehat{xy}_{iN} & w_{iN}\widehat{y^2}_{iN} & \cdots \end{bmatrix} \begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i \\
 &= \begin{pmatrix} \overline{U}_i \\ w_{i1}\overline{U}_1 \\ w_{i2}\overline{U}_2 \\ w_{i3}\overline{U}_3 \\ \vdots \\ w_{iN}\overline{U}_N \end{pmatrix} \tag{4.8}
 \end{aligned}$$

where the first row is the mean constraint, and the geometric terms are

$$\begin{aligned}
 \widehat{x^n y^m}_{ij} &\equiv \frac{1}{A_j} \int_{V_j} ((x - x_j) + (x_j - x_i))^n ((y - y_j) + (y_j - y_i))^m dA \\
 &= \sum_{l=0}^m \sum_{k=0}^n \frac{m!}{l! (m-l)!} \frac{n!}{k! (n-k)!} (x_j - x_i)^k (y_j - y_i)^l \overline{x^{n-k} y^{m-l}}_j
 \end{aligned}$$

and w , the weights, can be chosen to emphasize geometrically nearby data

$$w_{ij} = \frac{1}{|\vec{x}_j - \vec{x}_i|^n}. \tag{4.9}$$

where n is typically chosen to be 0, 1, or 2. Although the use of $n = 0$ has been shown to lead to some accuracy issues for highly anisotropic grids [9], in our experience it is the best choice for successful convergence of the scheme and is therefore used in the present work.

Since the matrix in Equation 4.8 depends purely on geometric terms it is possible to reduce the reconstruction step at each flux evaluation to a matrix-vector multiplication by precomputing a pseudoinverse during a

preprocessing stage. To obtain this form, we first analytically eliminate the mean constraint from Equation 4.8. Next, we compute the pseudoinverse of the resulting unconstrained least-squares problem. For this we use singular-value decomposition (see, for instance, [6] for details). The mean constraint can then be reintroduced to yield a matrix, which we store for each control volume, that can be used to compute the reconstruction coefficients using a single matrix-vector multiplication operation of the form

$$\begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i = A \begin{pmatrix} \bar{U}_i \\ \bar{U}_1 \\ \bar{U}_2 \\ \bar{U}_3 \\ \vdots \\ \bar{U}_N \end{pmatrix}. \quad (4.10)$$

Since the matrix A is based on purely geometric terms, the same pseudoinverse matrix can be used to reconstruct all the flow variables in the control volume which helps minimize memory use. Furthermore, this reconstruction form helps in constructing the exact flux Jacobian which was shown in Chapter 3 to be helpful in obtaining rapid convergence of high-order accurate schemes.

4.3.2 Boundary Condition Enforcement

In general, boundary conditions can be enforced either through special flux functions or by modifying the reconstruction step. In practice, we use special flux functions for boundary conditions of the inviscid flux and for maintaining the adiabatic condition at walls. We modify the reconstruction step to maintain the no-slip condition at the wall by enforcing Dirichlet boundary conditions for the components of velocity u and v . However, our approach to boundary condition enforcement is general enough to be applied to any Dirichlet or Neumann boundary condition. In previous work with the advection-diffusion equation [16] boundary conditions were enforced by introducing additional constraints on the least-squares problem in Equation

4.3. Solution Reconstruction

4.8. This resulted in the boundary conditions being satisfied exactly at the Gauss integration points. In the present work we modify this approach by instead augmenting the least-squares problem with additional reconstruction goals. While this does not ensure that the boundary conditions are met exactly, it does guarantee that they are satisfied within the truncation error of the scheme and it simplifies the reconstruction process.

Suppose that the solution must satisfy a Dirichlet boundary condition such that along part of the boundary $\partial\Omega_1$ the reconstruction approximately satisfies $R(\vec{x} - \vec{x}_i) = f_1(\vec{x})$. Rather than attempting to satisfy this condition everywhere along the boundary, we consider only the solution at each Gauss integration point \vec{x}_g on the boundary. In terms of the reconstruction polynomial coefficients, this reconstruction goal can be expressed as

$$\begin{aligned} f_1(\vec{x}_g) = & U|_{\vec{x}_i} + \frac{\partial U}{\partial x}\bigg|_{\vec{x}_i} (x_g - x_i) + \frac{\partial U}{\partial y}\bigg|_{\vec{x}_i} (y_g - y_i) \\ & + \frac{\partial^2 U}{\partial x^2}\bigg|_{\vec{x}_i} \frac{(x_g - x_i)^2}{2} + \frac{\partial^2 U}{\partial x \partial y}\bigg|_{\vec{x}_i} (x_g - x_i)(y_g - y_i) \\ & + \frac{\partial^2 U}{\partial y^2}\bigg|_{\vec{x}_i} \frac{(y_g - y_i)^2}{2} + \dots \end{aligned} \quad (4.11)$$

The boundary condition can be incorporated into Equation 4.8 by adding a row to the matrix for each boundary Gauss point giving

$$\begin{bmatrix} 1 & \bar{x} & \bar{y} & \bar{x}^2 & \bar{x}\bar{y} & \bar{y}^2 & \dots \\ w_{ia} & w_{ia}\delta x_{ia} & w_{ia}\delta y_{ia} & w_{ia}\delta x_{ia}^2 & w_{ia}\delta x_{ia}\delta y_{ia} & w_{ia}\delta y_{ia}^2 & \dots \\ w_{ib} & w_{ib}\delta x_{ib} & w_{ib}\delta y_{ib} & w_{ib}\delta x_{ib}^2 & w_{ib}\delta x_{ib}\delta y_{ib} & w_{ib}\delta y_{ib}^2 & \dots \\ w_{i1} & w_{i1}\widehat{x}_{i1} & w_{i1}\widehat{y}_{i1} & w_{i1}\widehat{x}_{i1}^2 & w_{i1}\widehat{x}_{i1}\widehat{y}_{i1} & w_{i1}\widehat{y}_{i1}^2 & \dots \\ w_{i2} & w_{i2}\widehat{x}_{i2} & w_{i2}\widehat{y}_{i2} & w_{i2}\widehat{x}_{i2}^2 & w_{i2}\widehat{x}_{i2}\widehat{y}_{i2} & w_{i2}\widehat{y}_{i2}^2 & \dots \\ w_{i3} & w_{i3}\widehat{x}_{i3} & w_{i3}\widehat{y}_{i3} & w_{i3}\widehat{x}_{i3}^2 & w_{i3}\widehat{x}_{i3}\widehat{y}_{i3} & w_{i3}\widehat{y}_{i3}^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ w_{iN} & w_{iN}\widehat{x}_{iN} & w_{iN}\widehat{y}_{iN} & w_{iN}\widehat{x}_{iN}^2 & w_{iN}\widehat{x}_{iN}\widehat{y}_{iN} & w_{iN}\widehat{y}_{iN}^2 & \dots \end{bmatrix} \begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i$$

$$= \begin{pmatrix} \frac{\bar{U}_i}{w_{ia}f_1(\vec{x}_a)} \\ w_{ib}f_1(\vec{x}_b) \\ w_{i1}\bar{U}_1 \\ w_{i2}\bar{U}_2 \\ w_{i3}\bar{U}_3 \\ \vdots \\ w_{iN}\bar{U}_N \end{pmatrix} \quad (4.12)$$

where the weights for the Gauss point rows can be computed as

$$w_{ig} = c_b \frac{1}{|\vec{x}_g - \vec{x}_i|^n} \quad (4.13)$$

and

$$\begin{aligned} \delta x_{ig} &\equiv x_g - x_i \\ \delta y_{ig} &\equiv y_g - y_i \end{aligned}$$

where n is the same value used in Equation 4.9 and c_b determines how strongly the boundary condition is enforced. Except where otherwise noted, we use $c_b = 6$. Neumann boundary conditions can be applied in an analogous way. If the reconstruction polynomial is to approximate the boundary condition $\frac{\partial U_i^R(\vec{x}-\vec{x}_i)}{\partial n} = f_2(\vec{x})$ we simply add reconstruction goals at the boundary Gauss points of the form

$$\begin{aligned} f_2(\vec{x}_g) &= \nabla U_i^R(\vec{x}_g - \vec{x}_i) \cdot \hat{n} \\ &= n_x \left(\frac{\partial U}{\partial x} \Big|_{\vec{x}_i} + \frac{\partial^2 U}{\partial x^2} \Big|_{\vec{x}_i} (x_g - x_i) + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} (y_g - y_i) + \dots \right) + \\ &\quad n_y \left(\frac{\partial U}{\partial y} \Big|_{\vec{x}_i} + \frac{\partial^2 U}{\partial x \partial y} \Big|_{\vec{x}_i} (x_g - x_i) + \frac{\partial^2 U}{\partial y^2} \Big|_{\vec{x}_i} (y_g - y_i) + \dots \right) \end{aligned}$$

where \hat{n} is the unit boundary normal.

As for the interior case, the pseudoinverse of the constrained least-squares

4.3. Solution Reconstruction

problem of Equation 4.12 can be computed to reduce the reconstruction to a matrix-vector product of the form

$$\begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i = \begin{bmatrix} A_b & A_i \end{bmatrix} \begin{pmatrix} f_1(\vec{x}_a) \\ f_1(\vec{x}_b) \\ \bar{U}_i \\ \bar{U}_1 \\ \bar{U}_2 \\ \bar{U}_3 \\ \vdots \\ \bar{U}_N \end{pmatrix},$$

where the component A_b of the pseudoinverse has as many columns as there are boundary Gauss points. Since the value of $f_1(\vec{x})$ does not change at each iteration, the boundary reconstruction step can be further optimized by precomputing a vector

$$v_b = A_b \begin{pmatrix} f_1(\vec{x}_a) \\ f_1(\vec{x}_b) \end{pmatrix}.$$

The reconstruction can then be found by

$$\begin{pmatrix} U \\ \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial x^2} \\ \frac{\partial^2 U}{\partial x \partial y} \\ \frac{1}{2} \frac{\partial^2 U}{\partial y^2} \\ \vdots \end{pmatrix}_i = A_i \begin{pmatrix} \bar{U}_i \\ \bar{U}_1 \\ \bar{U}_2 \\ \bar{U}_3 \\ \vdots \\ \bar{U}_N \end{pmatrix} + v_b. \quad (4.14)$$

Therefore the computational cost of reconstruction at the boundaries is almost identical to that of the interior. However since the different flow variables do not need to satisfy the same boundary conditions, the pseudoinverse matrix

A_i and vector v_b will need to be stored independently for each flow variable. Since only a small fraction of control volumes are adjacent to the boundary, this is not a major issue.

Boundary conditions which linearly couple flow variables or their derivatives can also be implemented. This is most easily accomplished by considering the constrained least-squares problem which simultaneously solves the reconstruction polynomial for all flow variables at a control volume. However this causes the memory and computational cost of boundary control volume reconstruction to increase quadratically with the number of flow variables. Therefore we elect to only simultaneously solve the reconstruction for the flow variables which are coupled by boundary conditions and independently solve those that are not.

4.4 Flux Integration

Once the reconstruction polynomial coefficients are computed as in Section 4.3, the solution and its gradients can easily be obtained at any Gauss integration point along the control volume boundary. In the present work, we use a vertex-centered centroidal-dual control volume which is obtained by connecting mesh face centroids with cell centroids. This results in two control volume edges per face in the primal mesh. Gauss integration is carried out on these dual edges using one Gauss point per control volume edge for the second-order scheme and two Gauss points for the third-, and fourth-order schemes. For the third-, and fourth-order schemes, it is well known that Gauss points must be located on the curved boundary rather than on the straight edge connecting vertices. Recently, it has been shown that for vertex-centered schemes the boundary face centroid used for the construction of the dual mesh must also be correctly located on the boundary, even for second-order schemes [4]. For this reason, we use curved boundary information for determining Gauss point locations, weights, and normals for all orders of accuracy.

The inviscid component of the flux is computed by evaluating the solution at the Gauss point using the reconstruction from both incident control

volumes as input to Roe’s approximate Riemann solver [19]. The viscous flux requires not only the solution, but also the solution gradients. For this we use the average gradient obtained from the reconstruction of the two incident control volumes. For a p -order accurate reconstruction scheme, fluxes that depend only on the solution will be p -order accurate. Since the solution gradient is only $(p - 1)$ -order accurate, the accuracy of the viscous flux is an order of accuracy lower than that of the inviscid flux. However, previous work [16] has analytically demonstrated that the even-order discretization of the Laplace equation on equilateral triangular grids benefits from error cancellation leading to an overall scheme that is p -order accurate. Numerical results in the same work confirmed the p -order accuracy of the overall scheme for the advection-diffusion equation on an isotropic unstructured grid. At the onset of this work it was not known whether this could also be expected of the Navier-Stokes equations on an unstructured grid.

4.5 Solution Convergence

The solution is converged to steady state using a Newton-GMRES [20] algorithm. The exact full-order Jacobian is used for forming the preconditioning matrix and for performing the matrix-vector products needed by the GMRES algorithm. The preconditioner used is ILU with one level of fill for the second-order scheme and zero levels of fill for the third- and fourth-order schemes. As in our work with the Euler equation, reproduced as Chapter 3 in this thesis, the globalization technique used is a combination of a local timestepping scheme with a line search algorithm. The local timestep is a function of a global timestep and the relative size of the local residual. The global timestep is increased as the solution converges using the successive evolution-relaxation (SER) [12] scheme. Convergence is facilitated by scaling the local timestep by the size of the local residual from the previous iteration. The line search algorithm is used to detect when an exceedingly large timestep is taken which may adversely affect the nonlinear convergence of the scheme. When this occurs, the subsequent timestep size is reduced.

In the manuscript presented in Chapter 3, it was shown that the exact

high-order accurate flux Jacobian for the Euler equation could be found using the chain rule

$$\frac{\partial FluxInt}{\partial CVars} = \frac{\partial FluxInt}{\partial Flux} \frac{\partial Flux}{\partial RecSol} \frac{\partial RecSol}{\partial RecCoef} \frac{\partial RecCoef}{\partial PVars} \frac{\partial PVars}{\partial CVars}$$

where *FluxInt* is the flux integral, *Flux* are the numerical fluxes, *RecSol* are the reconstructed solutions at Gauss points, *RecCoef* are the reconstruction coefficients, *PVars* are the control volume averages of the primitive variables, and *CVars* are the control volume averages of the conserved variables. In that work, a method to efficiently compute these terms was also presented. For viscous equations the flux depends not only on the solution but also its gradient. To accommodate this, we simply need to replace *RecSol*, the vector of reconstructed solutions at Gauss points, with a larger vector *RecSolGrad* which contains the solution and its two gradient terms. The term $\frac{\partial RecSolGrad}{\partial RecCoef}$ becomes the sensitivity of the reconstructed solution and solution gradients to the reconstruction coefficients, and the term $\frac{\partial Flux}{\partial RecSolGrad}$ becomes the Jacobian of the inviscid and viscous fluxes with respect to the reconstructed solution and its gradients. With these changes, the method to efficiently assemble the Euler flux Jacobian described in Chapter 3 can readily be used to assemble the Navier-Stokes flux Jacobian. The sparsity pattern of the assembled Jacobian matrix does not change through the addition of the viscous terms.

4.6 Results

4.6.1 Cylindrical Couette Flow

To numerically evaluate the order of accuracy of the scheme we consider flow around concentric cylinders. The outer cylinder with a radius of 2 is fixed and adiabatic while the inner cylinder has a radius of 1 and is rotated clockwise with a tangential Mach number of 0.5. Since isothermal wall conditions for the inner cylinder would be difficult to enforce with the current choice of reconstruction variables, we obtain an equivalent result by augmenting the

4.6. Results

Reconstruction Order	L_1	L_2	L_∞
2	1.60	1.59	1.37
3	2.22	2.25	2.38
4	3.54	3.57	3.26

Table 4.1: Order of error norms in velocity computed using a regression fit through data from all grids

fixed velocity wall condition with an isobaric condition. To avoid Taylor instabilities a Reynolds number of 10 based on the inner cylinder radius is used. Viscosity is considered constant for this case which allows comparison with the analytic exact solution [8]. In particular, in this case the velocity profile reduces to that of the incompressible flow with

$$v_\theta = \Omega_i r_i \frac{\frac{r_o}{r} - \frac{r}{r_o}}{\frac{r_o}{r_i} - \frac{r_i}{r_o}}$$

where v_θ is the tangential flow velocity, Ω_i is the angular velocity of the inner wall, r is the radius, and r_i and r_o and the radius of the inner and outer cylinders.

The numerical solution is carried out on a series of four meshes consisting of 219, 840, 3256, and 12812 vertices shown in Figure 4.1. Since the grids are unstructured and not aligned with the expected flow gradients and since the discretization is carried out in Cartesian coordinates, the error in the numerical solution should not have any unexpected benefits from the simplicity of the solution expressed in radial coordinates.

The L_2 norms of error in the velocity terms for the second-, third-, and fourth-order schemes is shown in Figure 4.2 and the convergence order of the error norms is shown in Table 4.1. The results indicate that the effective order of accuracy of the schemes falls between the nominal order of the flux calculation of the solution and gradient terms.

To demonstrate that the enforcement of boundary conditions using a least-squares approach is not detrimental to the overall accuracy of the scheme, we consider the effects of varying c_b in Equation 4.13. The L_2

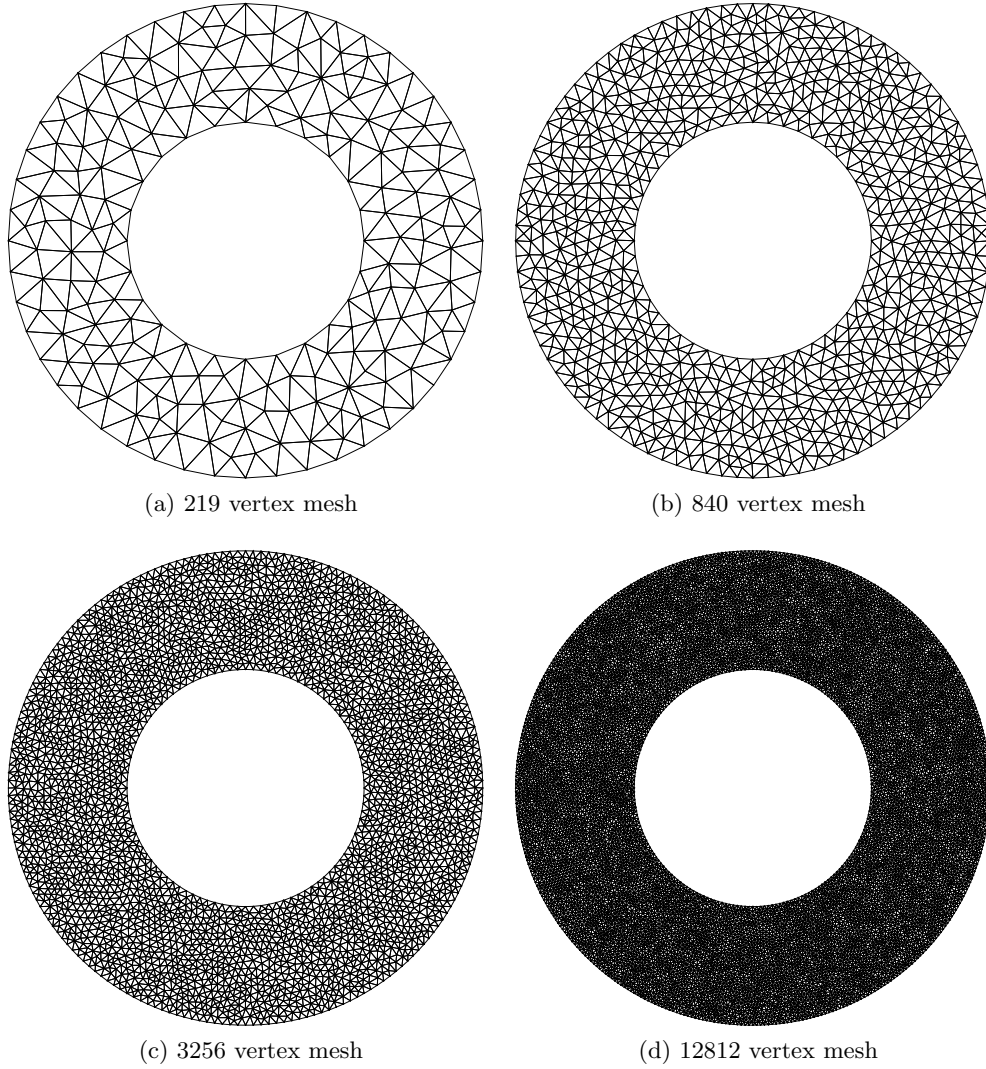


Figure 4.1: Four coarsest grids used in cylindrical Couette flow case

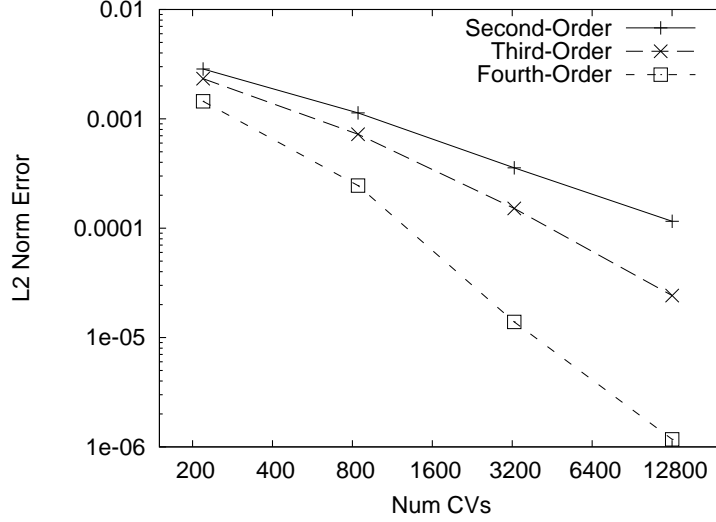


Figure 4.2: Convergence of error in velocity for cylindrical Couette flow

norm of error in the velocity terms sampled at 10,000 points along the inner cylinder boundary along with the norms of error in the interior of the domain is plotted in Figure 4.3. The results indicate that increasing the value of c_b only serves to decrease the error in the constrained terms at the boundary without affecting the overall accuracy of the solution in the interior.

4.6.2 Flow Over an Airfoil

Next we consider subsonic laminar compressible flow around an airfoil. Specifically, flow around a NACA 0012 airfoil at zero angle of attack at a Mach number of 0.5 and Reynolds number of 5000 is simulated on a series of grids. These grids were generated by adaptive refinement using the second-order scheme [17]. A total of six grids is considered with the coarsest containing 1345 vertices and the finest containing 37399 vertices. The grids are unstructured, anisotropic, and are not self-similar. The coarsest grid is shown in Figure 4.4 and a sample flow solution is given in Figure 4.5. The airfoil surface is considered adiabatic and Sutherland's law is used to compute the viscosity.

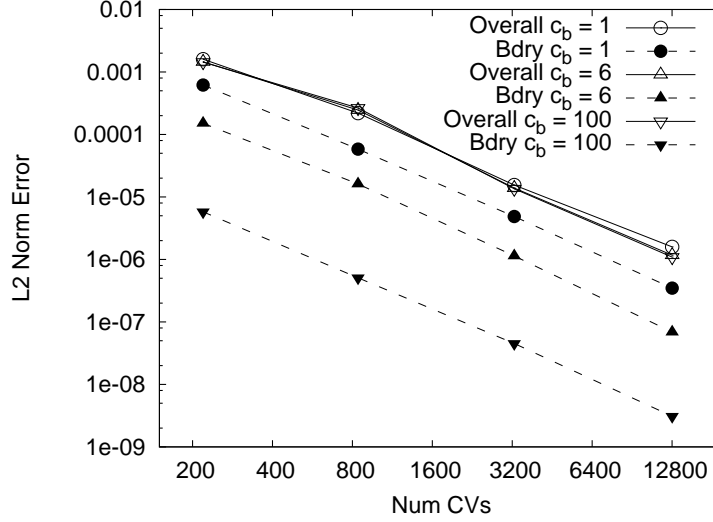
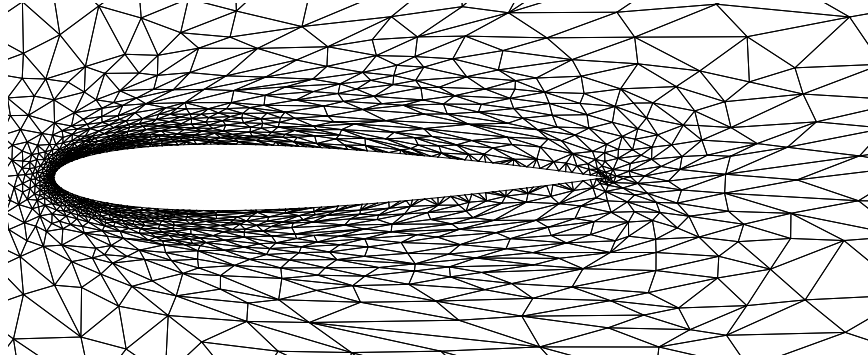


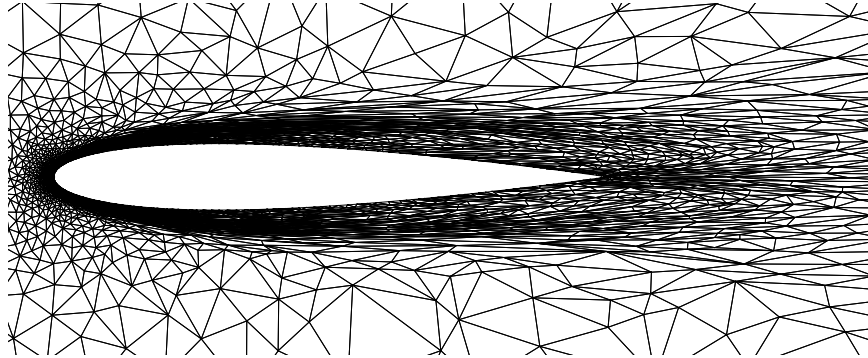
Figure 4.3: Convergence of error in velocity for the fourth-order scheme at the boundary and in the interior using different least-squares weights for the boundary condition terms

The convergence process is facilitated by using the converged solution from the previous coarser mesh as the initial solution on the next mesh. The solution is transferred by Gauss integration on the fine grid of the values obtained from the solution reconstruction on the coarse grid. Using this grid-sequencing method, the CPU time needed for a solution scales nearly linearly with the number of control volumes. The time required for a solution at each grid level is shown in Figure 4.6. These results along with the cumulative time required for the grid sequencing are shown in Table 4.2. In general, the third- and fourth-order schemes are a factor of 2.5 and 2.8, respectively, more costly than the second-order scheme. The majority of this additional cost can be attributed to the doubling in the number of Gauss points required in the high-order schemes.

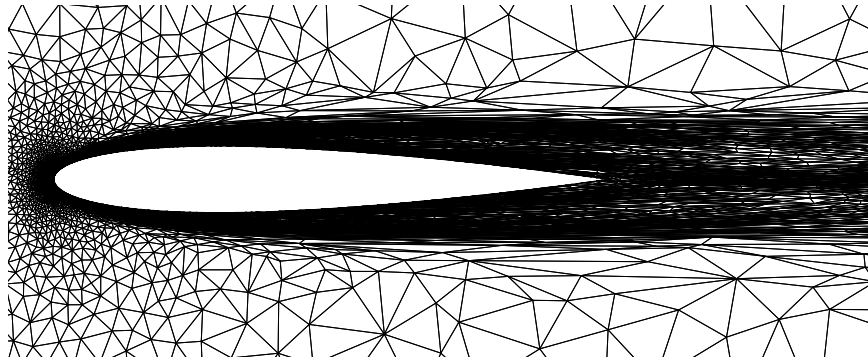
From the converged flow solution, the pressure and viscous drag coefficients are computed and the flow separation point is found. Table 4.3 demonstrates that the results are in general agreement with other published values. Next we consider the convergence of viscous drag with mesh refine-



(a) 1345 vertex grid



(b) 2681 vertex grid



(c) 5363 vertex grid

Figure 4.4: Three coarsest grids used for airfoil case

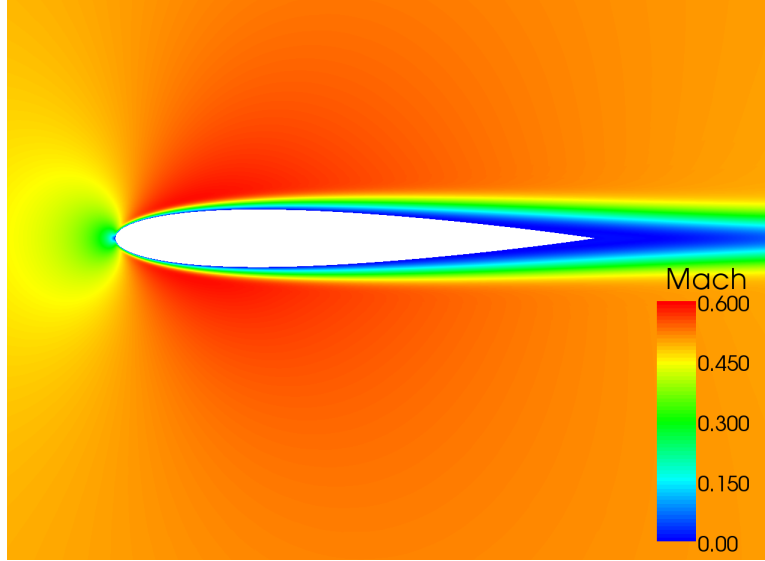


Figure 4.5: Mach number contours as computed on the 10541 vertex mesh using the fourth-order scheme

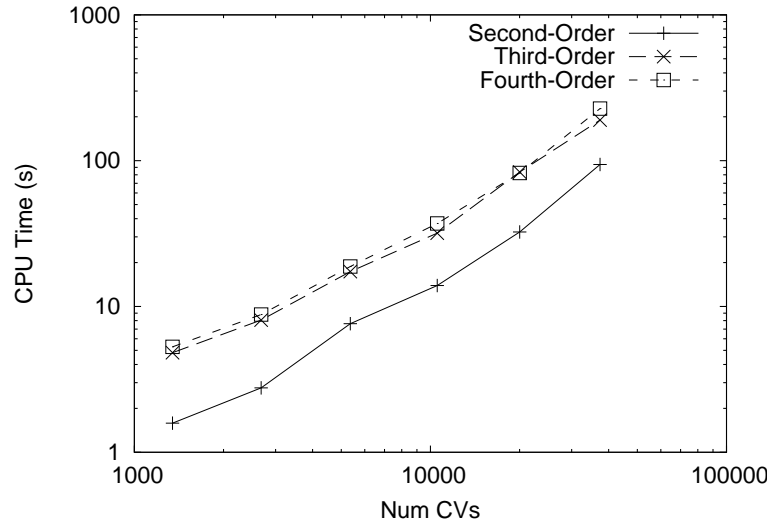


Figure 4.6: Computational time required for convergence at each grid level

4.6. Results

Number of Control Volumes	Incremental Time			Cumulative Time		
	2nd	3rd	4th	2nd	3rd	4th
1345	1.6	4.8	5.3	1.6	4.8	5.3
2681	2.8	8.1	8.8	4.4	12.9	14.1
5363	7.6	17.3	18.8	10.4	25.4	27.6
10541	13.9	31.9	37.1	21.5	49.2	55.9
20016	32.4	83.6	82.3	46.3	115.5	119.4
37399	94.2	190.0	227.4	126.6	273.6	309.7

Table 4.2: Computational time in seconds required for convergence of airfoil test case using grid sequencing

Method	Grid Size	Viscous Drag Coefficient	Pressure Drag Coefficient	Separation Point
Second-Order	37399	0.0311	0.0237	0.768
Third-Order	37399	0.0323	0.0224	0.778
Fourth-Order	37399	0.0321	0.0225	0.789
ARC2D [7]	320 x 128	0.0321	0.0221	0.824
Mavriplis [10]	320 x 64	0.0332	0.0229	0.814
Radespiel [18]	512 x128	0.0330	0.0224	0.814

Table 4.3: Comparison of fine grid results for airfoil case

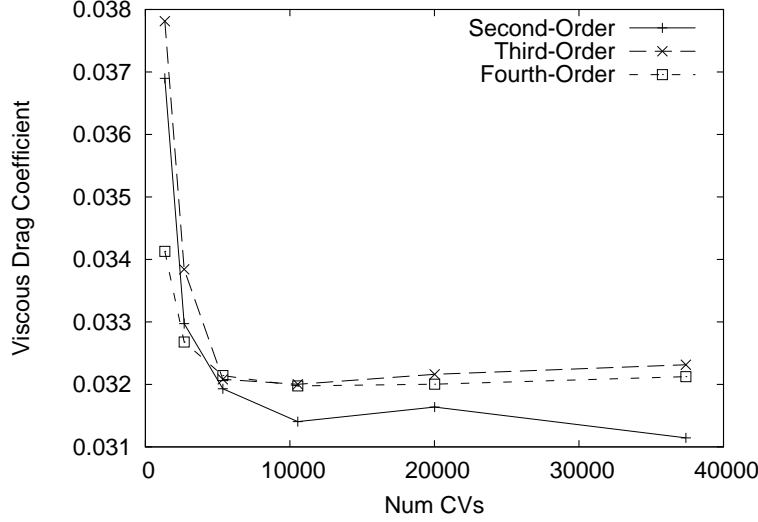


Figure 4.7: Viscous drag coefficient grid convergence

ment in Figure 4.7 and of pressure drag in Figure 4.8. We note that the fourth-order scheme exhibits excellent grid convergence of drag coefficients. In fact, both drag coefficients are within 4 counts of the converged values beginning at the third-coarsest mesh (5363 vertices) and within 2 counts of the converged values beginning at the fourth-coarsest mesh (10541 vertices). The third-order scheme also exhibits good viscous drag convergence but the pressure drag varies by 8 counts between the second-finest and finest grids. Like the third-order scheme, the second-order scheme does not exhibit grid-convergence of drag. Furthermore, the fine grid solution differs significantly from that of the third- and fourth-order schemes. Next we consider the separation point in Figure 4.9. The fourth-order scheme exhibits convergence of the separation point to within 1% of chord length beginning at the fourth-coarsest mesh while the third- and second-order schemes continue to see fluctuations of up to 5%.

Overall, the grid convergence study demonstrates that the fourth-order scheme is remarkably superior to the second- and third-order schemes. The fourth-order solution on the third-coarsest mesh is arguably superior to the second-order solution on the finest mesh and requires only $\frac{1}{5}$ the computa-

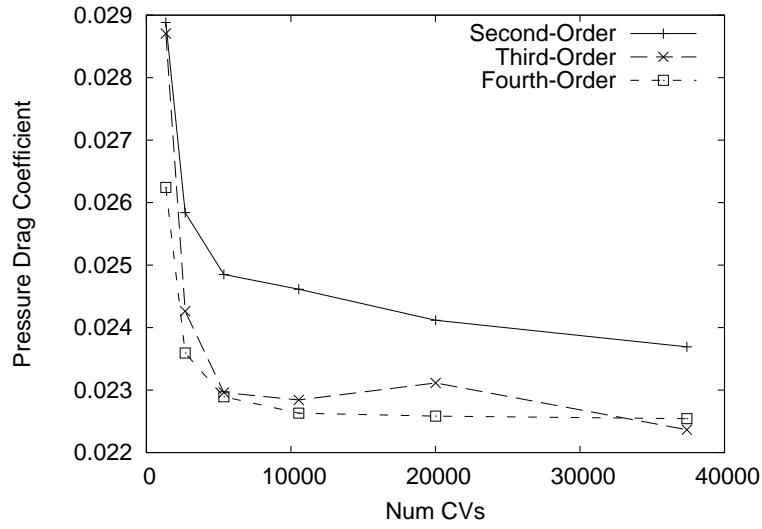


Figure 4.8: Pressure drag coefficient grid convergence

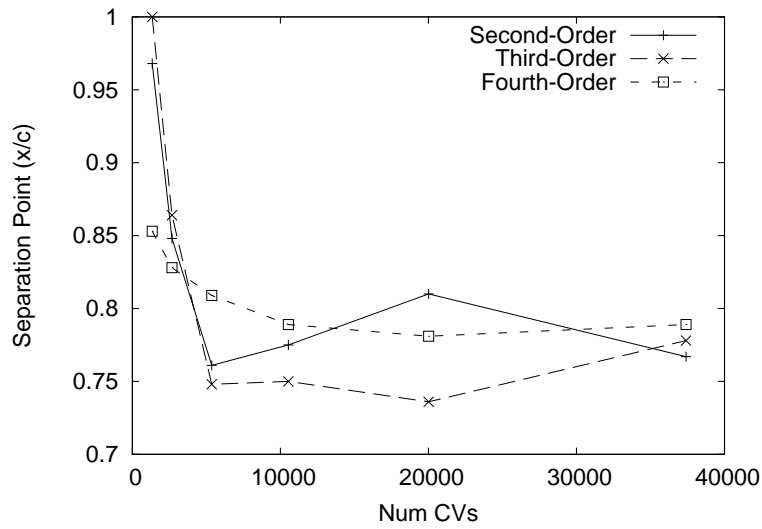


Figure 4.9: Grid convergence of the separation point on the upper airfoil surface

tional time.

4.7 Conclusion

The scheme based on a fourth-order reconstruction was shown to obtain an order of accuracy of approximately 3.5 under a mesh refinement study of the cylindrical Couette flow. The subsonic laminar airfoil test case demonstrated that this accuracy translates to rapid grid convergence of lift, drag and separation points relative to the second-order scheme. When combined with a matrix-explicit Newton-GMRES convergence acceleration technique, the fourth-order scheme was shown to be only a factor of 2.8 more computationally expensive than a second-order scheme on the same grid. The results indicate that the fourth-order scheme can be less computationally expensive for obtaining a level of accuracy typically desired for aerodynamics simulations.

Future work will examine the extension of the present scheme to turbulent flows. One significant challenge will lay in the stability of the high-order discretization of the turbulence model. Furthermore the need for curved boundary treatment for the high-order scheme combined with the highly-anisotropic grids required for efficient discretizations of turbulent flows can lead to malformed control volumes unless the interior faces are also curved.

4.8 Bibliography

- [1] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA paper 93-0668, Jan. 1993.
- [2] T. J. Barth and P. O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013, Jan. 1990.
- [3] M. Delanaye and J. A. Essers. Quadratic-reconstruction finite volume scheme for compressible flows on unstructured adaptive grids. *American Institute of Aeronautics and Astronautics Journal*, 35(4):631–639, Apr. 1997.
- [4] B. Diskin and J. Thomas. Accuracy analysis for mixed-element finite-volume discretization schemes. Technical Report 2007-08, National Institute of Aerospace, 2007.
- [5] P. Geuzaine, M. Delanaye, and J.-A. Essers. Computation of high Reynolds number flows with an implicit quadratic reconstruction scheme on unstructured grids. In *Proceedings of the Thirteenth AIAA Computational Fluid Dynamics Conference*, pages 610–619. American Institute of Aeronautics and Astronautics, 1997.
- [6] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1983.
- [7] C. F. Gooch. *Solution of the Navier-Stokes Equations on Locally-Refined Cartesian Meshes Using State-Vector Splitting*. PhD thesis, Stanford University, 1993.
- [8] C. Illingworth. Some solutions of the equations of flow of a viscous compressible fluid. *Proceedings of the Cambridge Philosophical Society*, 46:469–478, 1950.
- [9] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003.

- [10] D. J. Mavriplis, A. Jameson, and L. Martinelli. Multigrid solution of the Navier-Stokes equations on triangular meshes. ICASE Report No. 89-11, NASA Langley Research Center, 1989.
- [11] C. Michalak and C. Ollivier-Gooch. Matrix-explicit GMRES for a higher-order accurate inviscid compressible flow solver. In *Proceedings of the Eighteenth AIAA Computational Fluid Dynamics Conference*, 2007.
- [12] W. A. Mulder and B. van Leer. Experiments with implicit upwind methods for the Euler equations. *Journal of Computational Physics*, 59:232–246, June 1985.
- [13] A. Nejat and C. Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. *Journal of Computational Physics*, 227(4):2366–2386, 2008.
- [14] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227(4):2592–2609, 2008.
- [15] C. Ollivier-Gooch, A. Nejat, and C. Michalak. On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes. In *Proceedings of the Eighteenth AIAA Computational Fluid Dynamics Conference*, 2007. AIAA-2007-4464.
- [16] C. F. Ollivier-Gooch and M. Van Altena. A high-order accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.
- [17] D. Pagnutti and C. Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers and Structures*, To appear, 2009.
- [18] R. Radespiel. A cell-vertex multigrid method for the Navier-Stokes equations. NASA TM-101557, 1989.

4.8. Bibliography

- [19] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [20] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3):856–869, July 1986.

Chapter 5

Conclusion

In the previous chapters a number of specific improvements to the state of the art of the unstructured high-order finite-volume method were presented. In Chapter 2 a means of eliminating oscillations of the numerical solution near shocks while maintaining the high-order accuracy of the scheme and computational efficiency of least-squares reconstruction was devised. The explicit formation of the high-order Jacobian combined with novel globalization techniques was shown in Chapter 3 to provide significant improvement to the Newton-Krylov convergence strategy. In Chapter 4 a method to discretize diffusive fluxes of the Navier-Stokes equations and a means of enforcing the associated boundary conditions was presented and shown to attain the expected order of accuracy. In this chapter the combined significance of these works is discussed, how these results affect the relative competitiveness of the high-order finite-volume method compared with other high-order methods is addressed, and future developments are suggested.

5.1 Significance of Results

The intended combined significance of the manuscript chapters was to resolve specific limitations of the high-order least-squares reconstruction based finite-volume scheme so as to pave the way for future research aimed at exploring the feasibility and competitiveness of this method for aerodynamics simulations. The common theme of the work is the demonstration that for the simplified problem of two-dimensional inviscid and laminar flow the method can be accurate and efficient. While this does not prove that the method will be efficient or even accurate for the more difficult case of practical interest, three-dimensional turbulent flow, it provides a critical stepping stone in that

direction.

5.1.1 Limiter

The manuscript in Chapter 2 introduces a slope limiter for the least-squares reconstruction which does not inhibit high-order accuracy in smooth regions while maintaining monotonicity at shocks. Just as importantly, this is achieved in a computationally efficient manner and without degrading the steady-state convergence properties of the scheme. Previous researchers [3, 7, 17] have used a nominally high-order least-squares reconstruction together with a limiter to resolve flows with shocks. However none of these works have demonstrated or made explicit claims that the high-order accuracy of the method was maintained in smooth regions when such a limiter is used.

The broad family of essentially non-oscillatory (ENO) scheme [1, 10, 20, for instance] including the Weighted ENO (WENO) schemes [9, 11, 13, for instance] provides a viable solution to the accuracy and monotonicity challenge. However these schemes do not converge to steady state as efficiently as least-squares reconstruction schemes. Furthermore, the reconstruction step is much more costly than that of the least-squares reconstruction. A hybrid approach, named Quasi-ENO [19], has also been developed. In this approach, the reconstruction is carried out in a least-squares sense, but the weights assigned to the reconstruction goals are modified at each iteration to emphasize smooth data. Although found to be effective, this method is computationally expensive since the pseudoinverse of the least-squares matrix cannot be precomputed and stored. The cost savings enjoyed from this precomputation by regular least-squares reconstruction schemes increases dramatically as the order of accuracy, and therefore the size of the least-squares matrices, increases. For these reasons the limiter approach present in this work is more attractive than ENO-type schemes for steady-state solutions with relatively simple shock formations typically seen in transonic aerodynamics. The possible application of the limiter to unsteady problems with more complex shock formations for which ENO schemes have primarily been developed has yet to be studied.

The significance of the development of the new limiter may extend beyond high-order accurate schemes. The results in Chapter 2 demonstrated that even for second-order schemes, the new limiter provided significantly better accuracy than what is considered the de facto standard limiter [22] for unstructured finite-volume methods. The degradation of the accuracy of the solution with the use of limiters is a well-known problem. In fact, it is common practice to avoid using a limiter for cases where only weak shocks are expected. Such shocks occur, for example, in flow over a wing designed to operate efficiently in transonic regimes. The choice to avoid the use of limiters is made due to the improved convergence properties and accuracy in smooth regions of the unlimited scheme. However, this choice comes at the cost of introducing oscillations at the shock. The new limiter applied to the second-order scheme was shown to maintain the same accuracy as the unlimited scheme, therefore making the trade-off in choosing a good quality solution away from the shock at the expense of the quality of the solution at the shock unnecessary. Since the new limiter fits in essentially the same framework as other limiters, existing finite-volume codes which incorporate Venkatakrishnan's [22] or Barth-Jespersen's [3] limiter can easily be extended to accommodate it.

The extension of the limiter to turbulent viscous flows is expected to be straightforward. The only significant difference for the viscous case has to do with the no-slip wall boundary condition. The method of maintaining accuracy at stagnation points, which was developed and tested on the inviscid case, is expected to be effective at allowing the limiter to maintain accuracy near no-slip walls. In fact, the application of the limiter to viscous flows should be simpler than for inviscid flows since the special treatment of tangential flow at walls will no longer be necessary.

Likewise, the extension to three-dimensional flows is not expected to be problematic. The two-dimensional assumption is only used in the development of the special treatment of tangential flow at the wall. Therefore, for inviscid three-dimensional flows some adaptation of this method will need to take place. However for viscous three-dimensional flows the method as presented in this work should be directly applicable.

5.1.2 Globalized Matrix-Explicit Newton-GMRES

The manuscript in Chapter 3 introduces improvements to two aspects of the efficient steady-state solution process developed for the high-order finite-volume scheme. The first aspect deals with improving the robustness and efficiency of the globalization of the Newton method. The second aspect deals with the efficiency of the solution of the linear systems arising from the globalized Newton method.

As part of the improvement in the globalization, a method of combining timestepping with the line search method was proposed to improve robustness. Although both timestepping and line search methods are commonly used, this is to the author's knowledge the first study combining both methods in a CFD application. The lack of robustness of the Newton-GMRES method, particularly for flows with shocks, has prevented its rapid adoption in production codes. For high-order schemes, where the numerical method is less dissipative, the problem is exacerbated. Although far from being a full solution to this issue, the combined line search and timestepping method provides an additional tool in the arsenal that researchers and developers can use to improve the robustness of the solution process.

Another improvement to the globalization method presented in this work relates to the choice of local timestep. Selecting a timestep that depends on the relative local residual at the previous iterate was shown to be significantly more effective than using a CFL based local timestep or a global timestep. For transonic cases, in particular, the convergence of the scheme was drastically improved. The idea behind the use of the residual based timestep is simple. Regions where the residual, or flux integral, is large are changing rapidly and therefore require the use of small timesteps for the correct evolution to take place. Regions where small residuals prevail, on the other hand, benefit from large timesteps so that errors can be propagated to the boundaries quickly. An analogy to multigrid would be that the goal of the residual based local timestep scheme is to smooth high frequency error in regions where the residual at the previous iterate was large and to smooth low frequency error in regions where the residual was small. Further research is warranted to

determine if this approach is effective in reducing or eliminating the grid size dependence on the number of nonlinear iterations needed for convergence. However this is certainly not a complete replacement for multigrid since the stiffness of the resulting linear system maintains its grid size dependence. Both improvements to the globalization technique are not tied to the finite-volume method and could potentially be used with other discretizations.

Chapter 3 also demonstrates that the exact Jacobian for the high-order finite-volume method can be effectively used to improve the performance of the preconditioned GMRES linear solver. The benefit is most significant for the fourth-order scheme for which the first-order Jacobian had previously been shown to provide poor preconditioning [17]. The high-order Jacobian not only improves conditioning but also eliminates the need for residual evaluations resulting from the matrix-free GMRES method previously employed. Perhaps the most significant revelation from this work is the demonstration that the exact Jacobian, which was previously thought to be prohibitively expensive to compute, can be obtained at reasonable cost in both computational time and memory. The significance of this extends beyond its demonstrated application for efficient ILU preconditioned GMRES. For example, the availability of the explicit Jacobian introduces the possibility of using an algebraic multigrid method to solve the linear system. Another potential use of the exact Jacobian is for aerodynamic shape optimization. Adjoint-based methods have been proposed for this application [2, 12, 14]. Currently, the discrete adjoint method is favored over the continuous adjoint method since it does not require that the flux Jacobian be explicitly known. However, in light of the fact that the exact flux Jacobian was found to be relatively inexpensive to compute, the continuous adjoint methods may prove to be more efficient. This research question has begun to be addressed by another graduate student in the author's group.

5.1.3 Viscous Discretization

The extension of the least-squares based unstructured high-order finite-volume discretization to the full Navier-Stokes equations was carried out

in Chapter 4. Since, unlike the inviscid fluxes of the Euler equation, the viscous fluxes require the solution gradient in addition to the solution value, this represents an important extension to the discretization which warranted study. In addition to demonstrating that the expected order of accuracy could be achieved, it was also shown that such schemes can readily be converged through the use of the Newton-Krylov method with the help of the exact Jacobian, which can be obtained in a method similar to that described for the inviscid case in Chapter 3.

This is, to the author's knowledge, the first high-order unstructured least-squares reconstruction based finite-volume discretization to demonstrate high-order accuracy for viscous flows. However, the discontinuous Galerkin method has also been successfully used for this [4, 5, 8]. A comparative study of the accuracy and efficiency of the high-order finite-volume method with the discontinuous Galerkin method would certainly be warranted. However, since both methods are in their infancy, such a study would provide only preliminary information as to which scheme could fare better on more complex problems which are ultimately of interest.

5.2 Outlook

5.2.1 Turbulent Flows

The Reynolds averaged Navier-Stokes (RANS) equations provide a computationally feasible way of solving typical aerodynamics flows. For this application, the most common closure models are the one-equation Spalart-Allmaras [21] and two-equation $k - \omega$ and $k - \epsilon$ models [23]. These closure equations, consisting of convective, diffusive, and source terms, are difficult to solve due to their numerical stiffness. Their numerical solution must be carefully designed to maintain the positivity of turbulence variables. For this reason it is common for codes which use a second-order discretization for the RANS equations to use a first-order discretization for some or all of the terms of the turbulence model. While good results have been achieved with this method, it is anticipated that a first-order discretization will be a limiting

factor on the accuracy of the overall scheme if high-order accurate methods are used for the RANS equations. Therefore a stable high-order accurate discretization of the turbulence model needs to be developed. Although this question has not been explored for high-order accurate finite-volume schemes, it has been to be addressed in the context of the discontinuous Galerkin method [4] which may provide valuable insight.

The use of Newton-Krylov methods for the solution of turbulent flows will also require careful study. For second-order methods, the question has been addressed with some success. Due to numerical stiffness, some evidence suggests that a fully coupled approach to the solution of the turbulence and RANS equations may not provide the best convergence properties. For this reason a loosely-coupled approach has been suggested [6]. This approach is also attractive due to its reduced memory requirement resulting from the reduced block size of the Jacobian matrix. In the fourth-order case, where the use of the full-order Jacobian was shown to be important for good convergence, this savings in memory would be especially important.

Another complication is that the turbulence model equations, unlike the Navier-Stokes equations, contain source terms. The high-order Gauss integration rules for these terms on arbitrary control volumes, including those at curved boundaries, have been developed [18]. However, in their current state, these rules make no guarantee that the Gauss points will be inside the control volume itself or even within the domain boundary. For simple applications, such as initializing a solution, transferring a solution to another grid, or evaluating the error relative to an analytic solution, this is not problematic. In fact, this integration method is already being successfully used as part of the pre- and post-processing of the solution in our code. However if applied to the integration of turbulence source terms, problems may arise near the domain boundary. For control volumes near a no-slip wall some Gauss points will be located outside the domain on the other side of the wall. The evaluation of turbulence model source terms at these points is ill defined since they depend on the distance to the nearest wall and the velocity magnitude. Therefore some special methods may need to be developed for high-order integration of turbulence source terms near

wall boundaries.

The added complications of high Reynolds number flows are not limited to the issues directly related to the turbulence model. Since the boundary layer in these flows is very thin relative to the macroscopic flow properties, it is essential to use highly anisotropic grids to achieve computational efficiency. For aerodynamics applications, it is not unusual to use grids with cell aspect ratios of 10,000 : 1 in the boundary layer.

This is expected to cause problems with the stability, accuracy, and convergence properties of the scheme, all of which have also been issues with second-order schemes. For example, a relatively recent paper [16] addresses the accuracy of second-order least-squares reconstruction on anisotropic grids. Although the findings indicate that the use of distance weighing of the least-squares terms is essential for accuracy, this weighing is not typically used due to stability concerns. In practice, structured or quasi-structured grids are almost always used in the boundary layer. The resulting grid alignment with flow properties alleviates many of the accuracy and stability issues of reconstruction based finite-volume schemes. Whether these accuracy and stability issues will be exacerbated with the use of a high-order accurate reconstruction needs to be studied.

The difficulty of converging solutions to steady state using Newton-Krylov methods on anisotropic grids results in part from poor conditioning due to the strong directional coupling of flow properties. This problem will be aggravated by high-order methods since they have higher inherent numerical stiffness than second-order schemes. For second-order finite-volume methods, this has been addressed through the use of directional coarsening multigrid or the use of directional implicit smoothing [15]. Similar approaches will undoubtedly need to be considered for high-order schemes.

A final challenge of highly anisotropic grids, which is unique to high-order schemes, arises due to the treatment of curved boundaries. To obtain high-order accuracy it is essential to account for the curved boundary when locating flux integration points and when computing the control volume moments used for reconstruction. However, for highly-anisotropic grids, this may lead to ill-defined control volumes where the boundary face bulges past

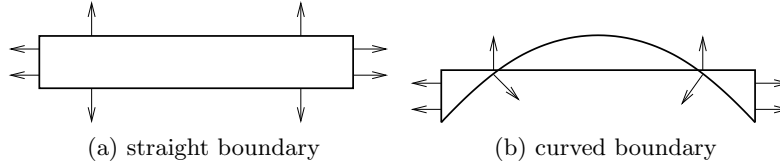


Figure 5.1: Effect of curved boundary treatment on a quadrilateral highly anisotropic boundary control volume

the interior face as seen in Figure 5.1. The solution to this problem may be to also curve the interior faces of anisotropic control volumes near the boundary. For this, a method to compute the appropriate location of the interior Gauss points used for flux integration and moment calculation needs to be devised.

5.2.2 Three Dimensional Flows

The fundamental aspects of the high-order discretization do not change substantially from the two-dimensional case to the three-dimensional case. Indeed, the discretization has already been implemented and shown to be accurate for simple problems such as the Poisson equation. However, the challenge of implementing curved boundary support has not yet been addressed. The Gauss integration rules developed for curved boundary control volumes [18] will need to be extended to the three-dimensional case. Furthermore, for turbulent flows, the schemes devised to accommodate highly anisotropic curved boundary control volumes will need to be extended to three dimensions.

In terms of computational efficiency, there is good news and bad news for the extension of the method to three dimensions. The good news is that the benefit of high-order accuracy, in terms of accuracy versus number of control volumes, should increase. For example, let's consider the results in Chapter 4 where a fourth-order solution of the airfoil problem on a mesh with 5,363 control volumes was more accurate than that of a second-order scheme on a mesh with 37,399 control volumes. To achieve similar mesh spacing in three dimensions the number of control volumes needs to be increased by a

power of $\frac{3}{2}$. Therefore, the equivalent mesh size in three dimensions would be 390,000 control volumes for the fourth-order scheme and 7,200,000 control volumes for the second-order scheme. While the two-dimensional problem required roughly 7 times fewer control volumes for the fourth-order solution, the three-dimensional problem may require 18 times fewer control volumes.

The bad news is that the computational cost of the high-order scheme relative to the second-order scheme on a per control volume basis also increases. In all likelihood the median dual vertex centered control volume, which we prefer in two dimensions due to conditioning issues, will need to be abandoned due to the large number of faces of such a tessellation in three dimensions. Even if a cell-centered approach is taken, which results in only four faces per control volume, the relative increase in cost of moving to a high-order accurate flux integration in three dimensions exceeds that of two dimensions. In two dimensions a fourth-order scheme requires twice as many Gauss integration points as a second-order scheme, while in three dimensions this factor increases to four. Furthermore, in terms of memory requirement, the additional cost of the matrix-explicit method relative to the matrix-free method will also increase. This is due to the increase in Jacobian block size from the additional velocity term, as well as to the increased flux dependency stencil which determines the matrix sparsity. Due to this plethora of differences in the three-dimensional case, it is difficult to speculate on the exact potential of high-order methods for these flows.

5.3 Bibliography

- [1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.
- [2] W. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers and Fluids*, 28(4–5):443–480, 1999.
- [3] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA paper 89-0366, Jan. 1989.
- [4] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and $k-\omega$ turbulence model equations. *Computers and Fluids*, 34(4-5):507 – 540, 2005. Residual Distribution Schemes, Discontinuous Galerkin Schemes and Adaptation.
- [5] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.
- [6] M. Blanco and D. W. Zingg. A Newton-Krylov algorithm with a loosely-coupled turbulence model for aerodynamic flows. AIAA Paper 2006-0691. Presented at the 44th AIAA Aerospace Sciences Meeting, Jan. 2006.
- [7] M. Delanaye. *Polynomial Reconstruction Finite Volume Schemes for the Compressible Euler and Navier-Stokes Equations on Unstructured Adaptive Grids*. PhD thesis, Universite de Liege, Faculte des Sciences Appliquees, 1998.
- [8] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 207(1):92 – 113, 2005.

5.3. Bibliography

- [9] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144(1):194–212, July 1998.
- [10] A. G. Godfrey, C. R. Mitchell, and R. W. Walters. Practical aspects of spatially high-order accurate methods. *American Institute of Aeronautics and Astronautics Journal*, 31(9):1634–1642, Sept. 1993.
- [11] C. Q. Hu and C. W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, Mar. 1999.
- [12] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1–4):213–237, 1998.
- [13] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, June 1996.
- [14] D. Mavriplis. A discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. AIAA Paper 2006-0050. Presented at the 44th AIAA Aerospace Sciences Meeting, Jan. 2006.
- [15] D. J. Mavriplis. On convergence acceleration techniques for unstructured meshes. Technical Report ICASE No. 98-44, Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, NASA Langley Research Center, Hampton VA 23681-2199, 1998.
- [16] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003.
- [17] A. Nejat and C. Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. *Journal of Computational Physics*, 227(4):2366–2386, 2008.

5.3. Bibliography

- [18] C. Ollivier-Gooch. On k -exact integration over arbitrary polygons and polyhedra. *ACM Trans. Math. Softw.*, In preparation, 2009.
- [19] C. F. Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6–17, 1997.
- [20] C.-W. Shu, T. A. Zang, G. Erlebacher, D. Whitaker, and S. Osher. High-order ENO schemes applied to two- and three-dimensional compressible flow. *Applied Numerical Mathematics*, 9(1):45–71, 1992.
- [21] P. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA paper 92-0439, Jan. 1992.
- [22] V. Venkatakrishnan. On the accuracy of limiters and convergence to steady-state solutions. AIAA paper 93-0880, Jan. 1993.
- [23] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, second edition, 2004.