

A HIGH-ORDER PARALLEL NEWTON-KRYLOV FLOW SOLVER FOR THE EULER EQUATIONS

by

Sydney Clement Dias

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

Copyright © 2009 by Sydney Clement Dias

Abstract

A high-order parallel Newton-Krylov flow solver for the Euler equations

Sydney Clement Dias

Master of Applied Science

Graduate Department of Aerospace Engineering

University of Toronto

2009

This work presents a parallel Newton-Krylov flow solver employing high-order finite-difference schemes to solve the steady three-dimensional Euler equations on a structured multi-block mesh. The fluxes are discretized using summation-by-parts operators, and interface and boundary conditions are implemented using simultaneous approximation terms. Functionals, drag and lift, are calculated using Simpson's rule. The code is verified using the method of manufactured solutions, Ringleb flow and a subsonic/supersonic vortex. The code is validated using the ONERA M6 wing. This work demonstrates code verification using order of accuracy studies that involve functionals as well. It also highlights the importance of mesh smoothness in achieving the prescribed order of accuracy. Finally, it demonstrates that high-order methods in conjunction with a parallel Newton-Krylov solver can be efficient and robust.

Acknowledgements

I would like to thank

- my supervisor, Prof. David W. Zingg, for his supervision and foresight. His experience, criticisms and attention to detail have greatly helped shape this work. His commitment and dedication to the research community are admirable.
- committee members, Prof. Alis Ekmekci, Prof. C.P.T. Groth and Prof. J. R. R. A. Martins for their comments and suggestions.
- Jason E. Hicken, for helping me get started with Diablo and explaining his implementation of summation-by-parts operators and simultaneous approximation terms. His suggestions, encouragement and friendship are very much valued and appreciated.
- Timothy Leung, for helping me get acquainted with the lab in my first days and for the many interesting lunches. His friendship is very much valued and appreciated.
- David Kam, for helping me get acquainted with the lab in my first days.
- Ken Mattson, for explaining his implementation of dissipation operators.
- my friends, for their support and encouragement.
- most importantly, my mom, dad and sister, for their never ending love, support and encouragement.

I gratefully acknowledge funding from the Mathematics of Information Technology and Complex Systems (MITACS), Bombardier Aerospace, Ontario Graduate Scholarship (OGS), the Canada Research Chairs program, and the University of Toronto.

Contents

Abstract	ii
Acknowledgements	iii
List of Tables	vi
List of Figures	vii
Nomenclature	x
1 Introduction	1
1.1 Motivation	1
1.2 Background	1
1.3 High-order methods	4
1.4 Objective	4
2 Governing Equations	5
2.1 Euler equations	5
2.1.1 Quasi-one-dimensional Euler equations	5
2.1.2 Three-dimensional Euler equations	6
2.2 Equation of state	9
3 Algorithm	10
3.1 Dimensionless variables	10
3.2 Spatial discretization	10
3.2.1 Finite-difference schemes	11
3.2.2 Grid metrics	12
3.2.3 Artificial dissipation	13

3.2.4	Interfaces	16
3.2.5	Boundary conditions	17
3.2.6	Stability	19
3.3	Time marching	20
3.3.1	Relaxation	20
3.3.2	Preconditioner	21
3.3.3	Local time stepping	22
3.3.4	Reordering of unknowns	22
3.3.5	Matrix-free GMRES	22
3.3.6	Start-up phase	22
3.3.7	Inexact-Newton phase	23
3.4	Force Integration	24
4	Results and Discussion	26
4.1	Test cases	26
4.1.1	Quasi-one-dimensional nozzle	26
4.1.2	Method of manufactured solutions	27
4.1.3	Ringleb flow	29
4.1.4	Vortex	30
4.2	Preservation of free stream flow	33
4.3	Matrix and scalar dissipation	34
4.4	Order of accuracy	37
4.5	Functionals	37
4.5.1	Integration method	37
4.5.2	Integration method and flow solver	46
4.6	Efficiency	46
4.7	Mesh smoothness	49
4.8	Parameter study	52
4.8.1	ILU fill parameter	52
4.8.2	Dissipation lumping factor	53
4.8.3	Relaxation	57
4.8.4	Lagged Jacobian update	61
4.8.5	Boundary condition and interface weighting constant	63
4.9	ONERA M6 Wing	63

5	Conclusions and Recommendations	71
5.1	Conclusions	71
5.2	Recommendations	72
A	ONERA M6 wing data	74
B	High-order finite-difference operators	84
B.1	Third-order method	84
B.2	Fourth-order method	85
	References	87

List of Tables

4.1	Constants and form of the solution for three-dimensional flow using the method of manufactured solutions	28
4.2	Residual norms for $N=33$	34
4.3	Residual norms for $N=129$	34
4.4	Order of accuracy for different test cases, slopes of lines of best fit	42
4.5	Slopes of the lines of best fit for the functional, f	45
4.6	Slopes of the lines of best fit for lift integration ($C_L S$) using both the integration method and the flow solver	49
4.7	Slopes of the lines of best fit for drag integration ($C_D S$) using both the integration method and the flow solver	49
4.8	Effect of the boundary and interface weighting constant τ_{bc} on the drag coefficient	63
4.9	Effect of the boundary and interface weighting constant τ_{bc} on the lift coefficient	63
4.10	Convergence summary for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$	66
4.11	Scalability of the flow solver	66
4.12	Lift and drag coefficients for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$	68
A.1	Lift and drag coefficients for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$	75
A.2	Convergence summary for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$	76
A.3	Lift and drag coefficients for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$	79
A.4	Convergence summary for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$	80

List of Figures

1.1	Optimization process	2
1.2	Structured mesh around a blended wing body configuration	3
2.1	Curvilinear mesh around an airfoil	7
3.1	Distribution of the diagonal of B showing the interior and transition regions	15
3.2	Figure showing two blocks with boundaries and an interface	17
3.3	Figure showing the surface of a wing for which the lift and drag are to be calculated	24
4.1	Cross-sectional area variation for the subsonic and transonic cases	27
4.2	Domain and solution for the three-dimensional method of manufactured so- lutions	29
4.3	Domain and exact solution for Ringleb flow	31
4.4	Vortex	31
4.5	Domain and exact solution for the vortex problem	32
4.6	Grid used to verify the grid metrics	33
4.7	Matrix and scalar dissipation	35
4.8	Matrix and scalar dissipation - Quasi-one-dimensional nozzle - Transonic case - 2 blocks \times 33 nodes	36
4.9	Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 2 nd order method	38
4.10	Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 3 rd order method	39
4.11	Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 4 th order method	40

4.12	Convergence histories - Matrix and scalar dissipation - ONERA M6 wing $M=0.84, \alpha=3.06^\circ$ with preconditioner lumping factor $\sigma = 7$	41
4.13	Order of accuracy for different test cases	43
4.14	Surface used to test the order of accuracy of the integration method	44
4.15	Order of accuracy for functional, f	45
4.16	Order of accuracy for $C_L S$	47
4.17	Order of accuracy for $C_D S$	48
4.18	Efficiency study	50
4.19	Grids used to test mesh smoothness	51
4.20	Order of accuracy depending on the grid smoothness	52
4.21	Effect of the fill level, k , for the ILU(k) factorization during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$	53
4.22	Effect of the fill level, k , for the ILU(k) factorization during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$	54
4.23	Effect of the fill level, k , for the ILU(k) factorization during the inexact- Newton phase for $M=0.699$ & $\alpha = 3.06^\circ$	55
4.24	Effect of the fill level, k , for the ILU(k) factorization during the inexact- Newton phase for $M=0.84$ & $\alpha = 3.06^\circ$	56
4.25	Effect of the constant σ used to lump the first and high-order dissipation coefficients for $M=0.699$ & $\alpha = 3.06^\circ$	57
4.26	Effect of the constant σ used to lump the first and high-order dissipation coefficients for $M=0.84$ & $\alpha = 3.06^\circ$	58
4.27	Effect of over- and under-relaxation during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$	59
4.28	Effect of over- and under-relaxation during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$	60
4.29	Effect of the update frequency of the Jacobian during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$	61
4.30	Effect of the update frequency of the Jacobian during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$	62
4.31	Mesh for the ONERA M6 wing	64
4.32	Convergence histories for the ONERA M6 wing - $M=0.699, \alpha=3.06^\circ$	67
4.33	Scalability of the flow solver	67

4.34	Pressure coefficient at different cross sections for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	69
4.35	Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	70
A.1	Convergence histories for the ONERA M6 wing - $M=0.3$, $\alpha=4^\circ$	74
A.2	Pressure coefficient at different for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	77
A.3	Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	78
A.4	Convergence histories for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$	81
A.5	Pressure coefficient at different cross sections for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	82
A.6	Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.	83

Nomenclature

α	angle of attack
$\Delta\hat{Q}$	solution update vector
$\Delta\xi, \Delta\eta, \Delta\zeta$	step in ξ, η, ζ directions, respectively
Δt	time step
$\Delta x, \Delta y, \Delta z$	step in x, y, z directions, respectively
$\delta_\xi^s, \delta_\eta^s, \delta_\zeta^s$	symmetric dissipation operators
ϵ	internal energy
γ	ratio of specific heats, $\gamma = 1.4$ for air
$\hat{\psi}, \psi$	source term for the Euler equations
κ	dissipation coefficient
κ_2	first-order dissipation coefficient
Λ	eigenvalue matrix of inviscid flux Jacobian
$\lambda_1, \lambda_2, \lambda_3$	eigenvalues of inviscid flux Jacobian
ω	inner iteration solve tolerance
$\partial_x F, F_x$	partial derivative of some function F with respect to x
π	constant, $\pi = 3.14159265358979323846$
ρ	density
σ	depending on context,

	<ul style="list-style-type: none"> • spectral radius of flux Jacobian • factor to lump the dissipation coefficient with the first order dissipation coefficient in the preconditioner matrix
$\sigma_{j+\frac{1}{2}}, A_{j+\frac{1}{2}}$	a forward average operator, can either represent a simple average or a Roe average
τ	depending on context, <ul style="list-style-type: none"> • time • factor by which the SAT's are scaled • drop tolerance to switch from approximate Newton to inexact Newton phase
θ	relaxation factor
Υ	pressure switch
ξ, η, ζ	curvilinear coordinates
$O(\Delta x)$	quantifies the mesh spacing and is given by, $O(\Delta x) = \frac{1}{N-1}$ where N is the number of nodes in the x coordinate direction
A	inviscid flux Jacobian
a	depending on context, <ul style="list-style-type: none"> • sound speed • parameter in startup phase
A^T	transpose of the matrix A
A^{-1}	inverse of the matrix A
$a_\infty, \rho_\infty, p_\infty, V_\infty$	free stream variables
b	startup phase parameter
B, \mathcal{B}	transition matrix for the dissipation operator
bc	boundary condition vector

C_L, C_D	lift and drag coefficients
C_p	depending on context, <ul style="list-style-type: none"> • pressure coefficient • specific heat capacity at constant pressure
C_v	specific heat capacity at constant volume
$\tilde{D}_+, \tilde{D}_-, \tilde{D}_3$	forward and backward difference operators
\tilde{D}_2, \tilde{D}_3	operators that are consistent approximations of $\frac{\partial^2}{\partial \xi^2}$ and $\frac{\partial^3}{\partial \xi^3}$
$D, D_\xi, D_\eta, D_\zeta, \delta_\xi^a, \delta_\eta^a, \delta_\zeta^a$	antisymmetric first derivative finite-difference operators
$D = \text{diag}[a, b, c, \dots]$	diagonal matrix with entries a, b, c, \dots along the diagonal, may be a block diagonal matrix if a, b, c, \dots are matrices
$\hat{E}_\infty, \hat{F}_\infty, \hat{G}_\infty$	free stream fluxes
$E, F, G, \hat{E}, \hat{F}, \hat{G}$	vectors containing inviscid fluxes
$E, F, G, \hat{E}, \hat{F}, \hat{G}$	vectors containing inviscid fluxes over the whole domain
e	total energy
H, \mathcal{H}	symmetric positive definite matrix used to define the finite difference operator
$I_\xi, I_\eta, I_\zeta, I_5, I$	identity matrices
J	metric Jacobian
j, k, m	node indices
$A \otimes B$	Kronecker product of matrices A and B , $C = A \otimes B \in M^{mp \times nq}$ is defined by $C_{p(i-1)+k, q(j-1)+l} = A_{i,j} B_{k,l}$
k	ILU fill level
l	length scale used to non-dimensionalise variables, typically the root chord length

M	depending on context, <ul style="list-style-type: none"> • Mach number • any ordinary matrix • preconditioner matrix
$\hat{n}^{(\xi\eta)}, \hat{n}^{(\eta\zeta)}, \hat{n}^{(\zeta\xi)}$	unit vectors normal to the, $\xi\eta$, $\eta\zeta$, and $\zeta\xi$ planes, respectively
n	iteration number
n_ξ, n_η, n_ζ	number of nodes in the ξ , η and ζ directions, respectively
P	matrix used to define the finite-difference operator such that $P + P^T = \text{diag}[-1, 0, 0, \dots, 0, 0, 1]$
p	pressure
Q, \hat{Q}	vectors containing conservative flow variables
Q, \hat{Q}	vectors containing conservative flow variables over the whole domain
Q_{bc}	conservative flow variables that need to be imposed at the boundaries
$R, R(\hat{Q})$	depending on context, <ul style="list-style-type: none"> • residual vector • Gas constant
S	depending on context, <ul style="list-style-type: none"> • cross section area for the quasi-one-dimensional nozzle • surface area of a wing
$\hat{t}^{(\xi)}, \hat{t}^{(\eta)}, \hat{t}^{(\zeta)}$	unit vectors tangential to the ξ , η and ζ directions, respectively
T	depending on context, <ul style="list-style-type: none"> • transformation matrix to convert the velocity components into tangential and normal components • temperature

t	time
U, V, W	velocity components for the curvilinear coordinate system
u, v, w	Cartesian velocity components
V_{T1}, V_{T2}, V_N	tangential and normal velocity components
$\hat{x}, \hat{y}, \hat{z}$	unit vectors for Cartesian coordinates
$\tilde{x}, \tilde{y}, \tilde{z}, \tilde{u}, \tilde{v}, \tilde{w}, \tilde{e}, \tilde{t}$	represents non-dimensional variables
X	eigenvector matrix of inviscid flux Jacobian
x, y, z	Cartesian position coordinates
$x_{j,k,m}, A_{j,k,m}$	value of variable x or matrix A at node with indices j, k, m

Chapter 1

Introduction

1.1 Motivation

Fuel consumption and emissions are two major issues that are kept in mind when designing aircraft today. The two issues are not mutually exclusive. Rising fuel costs have caused major airlines to cut operations and in extreme cases shut down. Therefore there is a great need to find alternate fuel sources and design aircraft that have a lower fuel consumption per passenger-km. Since alternate fuel sources are not feasible in the near future, there is a greater emphasis on the latter.

Aircraft emissions include CO_2 , H_2O , NO_x , hydrocarbons, CO , soot and SO_2 . Not all of these contribute to climate change. However, the impact of aircraft emissions on climate change is significant and continues to increase. Therefore there is also the need to reduce emissions per passenger-km [5, 55].

In order to meet these design requirements, several research groups around the world are developing numerical methods to optimize aircraft that will aid aircraft manufacturers [18, 44, 24, 48, 83, 82, 46, 45, 84, 81, 63]. The process for aerodynamic shape optimization is shown in Figure 1.1, and the aim of this work is to develop an efficient and robust flow solver which will then be used as a platform for aerodynamic shape optimization.

1.2 Background

There are basically three design tools available to aerospace engineers for the design of aircraft:

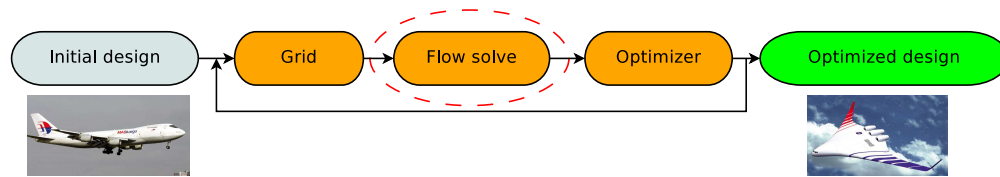


Figure 1.1: Optimization process

- flight tests,
- wind tunnel tests, and
- computational fluid dynamics (CFD).

Flight tests are expensive to set up and thus one needs to focus on wind tunnel tests and CFD. Wind tunnel tests, as the name suggests, use an actual fluid. CFD on the other hand is as good as the fluid model you use. However, CFD, if used knowledgeably, is inexpensive and numerical optimization methods can be applied to aid in the design process. Although one cannot rely on CFD alone, in recent years it has come to play a major role in the design of cost-effective and high-performance aircraft [26].

The following steps as outlined by Lomax et al. are employed in the use of CFD [32]:

1. Problem specification and geometry preparation
2. Selection of governing equations
3. Gridding strategy
4. Numerical method
5. Interpretation of results

The problem is first specified. For example, one may wish to find the induced drag on a wing. Secondly, one needs to choose the appropriate governing equations. The Navier-Stokes equations simulate all continuum fluid flow effects. However, if one wishes to simulate turbulence effects directly then one would need an excessively high mesh resolution, and so turbulence models are used instead. Once the flow models have been selected, appropriate grids are generated. Grids generally come in two varieties, structured and unstructured. Structured grids are beneficial in that they normally use computer resources efficiently and produce the smallest error for a given mesh density. On the other hand, complex body shapes

can be easily dealt with using unstructured grids. Additionally, mesh adaptation techniques are easily applied to unstructured grids. Complex body shapes for structured grids may be dealt with using a multi-block gridding strategy. A structured grid for a blended wing body configuration is shown in Figure 1.2.

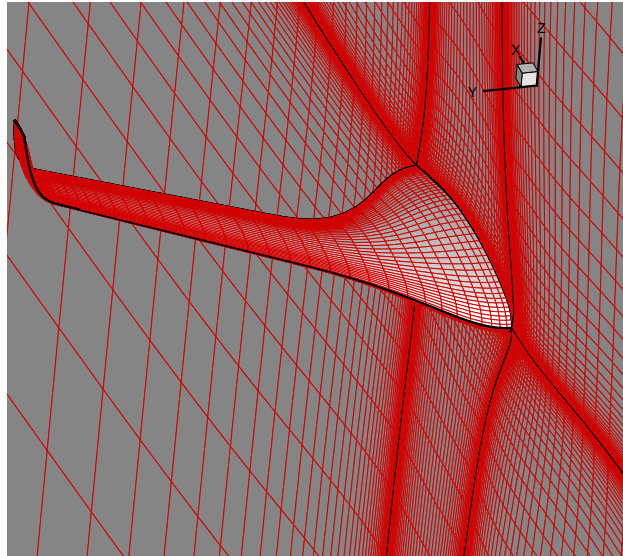


Figure 1.2: Structured mesh around a blended wing body configuration

The numerical method which is the subject of this work can be divided into

1. Spatial discretization
2. Time marching
3. Postprocessing, such as force and moment calculations

The spatial discretization schemes available are finite-difference, finite-volume and finite-element methods. Finite-difference schemes may only be used with structured grids. Time-marching methods may be classified into implicit and explicit schemes. Explicit schemes are cheaper per iteration but require more iterations as they have small stability bounds. Compared to a purely explicit scheme, implicit methods are better for extremely stiff problems since a larger time step may be used. Postprocessing would normally involve drag and lift calculations.

In the last step of the CFD process the data may be assessed against other databases and an assessment made of the errors in the flow solution; such as by calculating the flow adjoint

[76]. If the solution is not satisfactory, one would go through the entire process again, i.e. use a different flow model, a different grid, etc.

1.3 High-order methods

The focus of this work is to develop an efficient and robust numerical method, in particular the focus shall be on high-order finite-difference schemes. The order of accuracy of the spatial discretization is normally second-order, i.e. the leading error term in the solution is of order $O(\Delta x^2)$, where Δx is the mesh spacing. Spatial discretization that is higher than second-order is referred to as high-order spatial discretization.

Flow solvers have been developed with high-order finite-element, finite-volume and finite-difference schemes. For example, Gooch et al. [40, 41, 38, 39, 42] and Vaassen et al. [75] have developed high-order finite-volume schemes, Darmofal et al. [16, 17, 14, 15] have developed high-order finite-element schemes, and Zingg et al. [12, 86] and Svard et al. [72, 35] have developed high-order finite-difference schemes.

The reason high-order methods are being pursued is because compared to a second-order scheme they generally produce a smaller error per node/cell with a small increase in computational cost per node/cell [12, 41, 17] and are thus efficient. Hence, using a high-order method, one can obtain a solution of a required accuracy using fewer nodes/cells than what would be required for a second-order method. Additional benefits of using coarser grids enabled by the use of high-order methods are also seen in aerodynamic shape optimization, such as mesh movement, etc.

1.4 Objective

The objective of this work is to develop a parallel Newton-Krylov flow solver employing high-order finite-difference schemes to solve the steady three-dimensional Euler equations on a structured multi-block mesh. The fluxes will be discretized using summation-by-parts operators, and boundary and interface conditions will be implemented using simultaneous approximation terms. High-order integration shall also be developed as a part of this work. The solver thus developed shall be verified using several test cases such as the method of manufactured solutions and Ringleb flow and will be validated using the ONERA M6 wing.

Chapter 2

Governing Equations

This chapter lays out the equations that are used to model the physical fluid flow in subsequent chapters. It starts by discussing the quasi-one-dimensional Euler equations and then moves on to discuss the three-dimensional Euler equations. Finally it talks about how the three-dimensional Euler equations can be extended to a curvilinear coordinate system.

2.1 Euler equations

The Euler equations describe continuum, inviscid, compressible fluid flow with no heat conduction. The validity of continuum fluid flow depends on the Knudsen number, Kn . The Knudsen number is the ratio of the mean free path of the molecules of the fluid to some characteristic length scale. If $Kn \ll 1$ we can assume continuum flow. The validity of inviscid flow depends on the Reynolds number, Re . The Reynolds number is the ratio of the inertial stresses to the viscous stresses. Inviscid flow may be assumed if the Reynolds number is high. However, this assumption fails in the vicinity of solid boundaries where viscous effects dominate. For instance if one wishes to study drag on a wing, inviscid flow equations allow prediction of only induced drag and wave drag. The choice to model compressibility effects depends on the Mach number, M . The Mach number is the ratio of the local flow speed to the local sound speed. If $M_\infty > 0.3$, compressibility effects become important [78].

2.1.1 Quasi-one-dimensional Euler equations

Quasi-one-dimensional flow is generally used to solve nozzle problems. The assumption is that the flow is uniform along the cross-section of the nozzle. For this to be true, the cross

sectional area (S) must vary slowly along the axis. Hence all physical flow quantities are functions of the position (x) along the axis of the nozzle. The quasi-one-dimensional Euler equations in conservative form for flow with density ρ , velocity u , pressure p , internal energy e and cross-sectional area (S) are given by

$$\partial_t \mathbf{Q} + \partial_x \mathbf{F} = \psi \quad (2.1)$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho S \\ \rho u S \\ e S \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho u S \\ (\rho u^2 + p) S \\ u(e + p) S \end{bmatrix} \quad \psi = \begin{bmatrix} 0 \\ p \frac{dS}{dx} \\ 0 \end{bmatrix}$$

The three equations are the conservation of mass, the conservation of momentum and the conservation of energy, respectively [29].

2.1.2 Three-dimensional Euler equations

The three-dimensional Euler equations for a Cartesian coordinate system are given by

$$\partial_t \mathbf{Q} + \partial_x \mathbf{E} + \partial_y \mathbf{F} + \partial_z \mathbf{G} = 0 \quad (2.2)$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ u(e + p) - p \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho w v \\ v(e + p) - p \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ w(e + p) - p \end{bmatrix}$$

The equations comprise the conservation of mass, a vector equation for the conservation of momentum in each of the three coordinate directions, and the conservation of energy.

Generalized coordinate system

For most applications, the grids on which the solution is to be carried out do not conform to a standard Cartesian coordinate system as shown in Figure 2.1, so it is necessary to transform

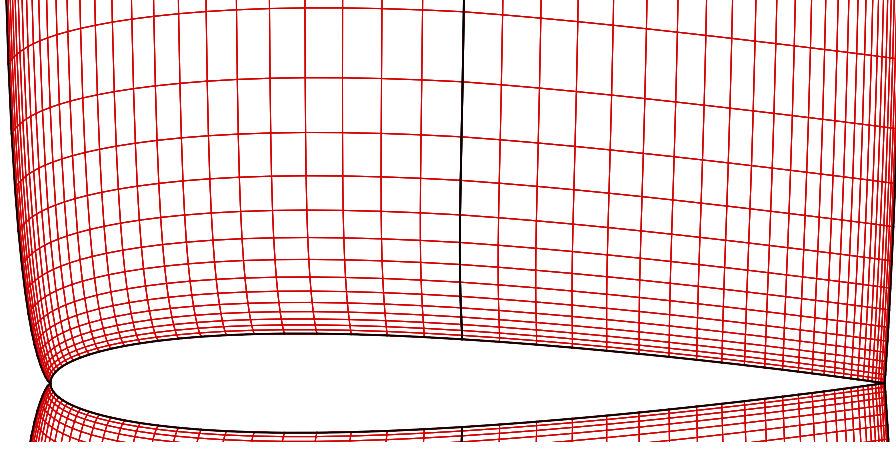


Figure 2.1: Curvilinear mesh around an airfoil

the equations in (2.2) to a generalized curvilinear coordinate system. For this the following transformations are used:

$$\tau = t \quad (2.3)$$

$$\xi = \xi(x, y, z, t) \quad (2.4)$$

$$\eta = \eta(x, y, z, t) \quad (2.5)$$

$$\zeta = \zeta(x, y, z, t) \quad (2.6)$$

where ξ , η , ζ and τ correspond to the new coordinate system. Now if a one-to-one transformation is assumed, the inverse transformations exist as well. Using these transformations we can obtain relations for the grid metrics:

$$\begin{aligned} \xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta) & \xi_y &= J(z_\eta x_\zeta - z_\zeta x_\eta) & \xi_z &= J(x_\eta y_\zeta - y_\eta x_\zeta) \\ \eta_x &= J(z_\xi y_\zeta - y_\xi z_\zeta) & \eta_y &= J(x_\xi z_\zeta - z_\xi x_\zeta) & \eta_z &= J(y_\xi x_\zeta - x_\xi y_\zeta) \\ \zeta_x &= J(y_\xi z_\eta - z_\xi y_\eta) & \zeta_y &= J(z_\xi x_\eta - x_\xi z_\eta) & \zeta_z &= J(x_\xi y_\eta - y_\xi x_\eta) \\ \xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z & \eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z & \zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z \end{aligned} \quad (2.7)$$

where J is the metric Jacobian given by

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi$$

Equation (2.2) with respect to the transformed coordinate system is given by

$$\begin{aligned}
& \partial_\tau \mathbf{Q} + \xi_t \partial_\xi \mathbf{Q} + \eta_t \partial_\eta \mathbf{Q} + \zeta_t \partial_\zeta \mathbf{Q} \\
& + \xi_x \partial_\xi \mathbf{E} + \eta_x \partial_\eta \mathbf{E} + \zeta_x \partial_\zeta \mathbf{E} \\
& + \xi_y \partial_\xi \mathbf{F} + \eta_y \partial_\eta \mathbf{F} + \zeta_y \partial_\zeta \mathbf{F} \\
& + \xi_z \partial_\xi \mathbf{G} + \eta_z \partial_\eta \mathbf{G} + \zeta_z \partial_\zeta \mathbf{G} = 0
\end{aligned} \tag{2.8}$$

By adopting a generalized coordinate system, the conservative form of the Euler equations has given way to a weak conservative form because the grid metrics appear as coefficients in (2.8). To obtain the conservative form, a set of identities known as the metric invariants must be satisfied. The metric invariants are as follows:

$$\begin{aligned}
& (\xi_x)_\xi + (\eta_x)_\eta + (\zeta_x)_\zeta = 0 \\
& (\xi_y)_\xi + (\eta_y)_\eta + (\zeta_y)_\zeta = 0 \\
& (\xi_z)_\xi + (\eta_z)_\eta + (\zeta_z)_\zeta = 0 \\
& (1/J)_\tau + (\xi_t)_\xi + (\eta_t)_\eta + (\zeta_t)_\zeta = 0
\end{aligned} \tag{2.9}$$

The conservative form can be obtained by making the following substitution for each of the partial derivatives in (2.8),

$$J^{-1} \xi_x \partial_\xi = \partial_\xi \left(\frac{\xi_x}{J} \mathbf{E} \right) - \mathbf{E} \partial_\xi (J^{-1} \xi_x) \tag{2.10}$$

Using the metric invariants (2.9) and collecting like partial derivatives, the conservative form of the Euler equations for a generalized curvilinear coordinate system is obtained [59, 77]:

$$\partial_\tau \hat{\mathbf{Q}} + \partial_\xi \hat{\mathbf{E}} + \partial_\eta \hat{\mathbf{F}} + \partial_\zeta \hat{\mathbf{G}} = 0 \tag{2.11}$$

where

$$\hat{\mathbf{Q}} = J^{-1} \mathbf{Q}$$

$$\hat{\mathbf{E}} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U(e + p) - \xi_t p \end{bmatrix} \quad \hat{\mathbf{F}} = J^{-1} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V(e + p) - \eta_t p \end{bmatrix} \quad \hat{\mathbf{G}} = J^{-1} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W(e + p) - \zeta_t p \end{bmatrix}$$

and the contravariant velocities are given by,

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w$$

2.2 Equation of state

Looking at (2.1), (2.2) and (2.11), we see that number of unknowns is one more than the number of equations. Hence another equation is required to form a closed set. This last equation is the equation of state. We use the ideal gas approximation

$$p = \rho R T \tag{2.12}$$

For a calorically perfect gas, i.e. the specific heats C_v and C_p are constant. With $\epsilon = C_v T$, $\gamma = \frac{C_p}{C_v}$, $C_p = C_v + R$ and $e = \rho\epsilon + \frac{1}{2}(u^2 + v^2 + w^2)$, we obtain our last equation

$$p = (\gamma - 1) \left(e - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right) \tag{2.13}$$

Since this thesis deals with aerodynamic applications, it suffices to use $\gamma=1.4$ for air.

Chapter 3

Algorithm

In this chapter the algorithm used to solve the governing equations shall be described. The chapter begins by introducing the spatial discretization scheme and then moves on to talk about the time-marching technique employed.

3.1 Dimensionless variables

In this work dimensionless variables are used for aerodynamic applications. Most of the variables are scaled by the free stream conditions (∞). They are given by [85]:

$$\begin{aligned} \tilde{x} &= \frac{x}{l} & \tilde{y} &= \frac{y}{l} & \tilde{z} &= \frac{z}{l} \\ \tilde{u} &= \frac{u}{a_\infty} & \tilde{v} &= \frac{v}{a_\infty} & \tilde{w} &= \frac{w}{a_\infty} \\ \tilde{t} &= \frac{ta_\infty}{l} & \tilde{e} &= \frac{e}{\rho_\infty a_\infty^2} \end{aligned}$$

where a is the speed of sound, and the length scale l for a wing is normally chosen to be the chord length. Henceforth x refers to \tilde{x} and so on.

3.2 Spatial discretization

The three-dimensional domain is discretized using structured grids. Additionally, the domain is broken down into a number of blocks to enable the solver to deal with complex geometries. Also a multi-block domain facilitates parallelization [19].

For the purposes of this work assume that the domain is decomposed into N blocks. Each

block consists of n_ξ , n_η and n_ζ nodes in the ξ , η and ζ directions, respectively.

3.2.1 Finite-difference schemes

Operators that satisfy the summation-by-parts (SBP) property are used. For a spatial discretization operator $D = H^{-1}P$, applied to vectors u and v such that $\xi \in [a, b]$, the SBP property is defined as [30, 13, 70, 34]:

$$\langle u, Dv \rangle + \langle Dv, u \rangle = uv|_a^b \quad (3.1)$$

where the inner product is defined by:

$$\langle u, v \rangle = u^T H v \quad (3.2)$$

and H is a positive definite matrix. SBP operators are used in this work because they provide an energy estimate. This shall become apparent in the section on stability.

SBP operators come in two varieties, using diagonal and non-diagonal norms. In this work diagonal norm operators are used because they guarantee stability when using a curvilinear coordinate system in three dimensions [51, 52, 13, 71]. As an example of SBP operators, H and P for a second-order spatial discretization of the first derivative are shown below [70, 34]:

$$H = \begin{pmatrix} \frac{1}{2} & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \frac{1}{2} \end{pmatrix} \quad P = \frac{1}{2\Delta\xi} \begin{pmatrix} -1 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 1 \end{pmatrix}$$

H and P for high-order schemes are given in Appendix B. We notice that they are centered differences (anti-symmetric) and would therefore require dissipation (symmetric), which shall be discussed later.

Now the Euler equations (2.11) become,

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\eta^a \hat{F} + \delta_\zeta^a \hat{G} = 0 \quad (3.3)$$

The antisymmetric operators are defined by,

$$\begin{aligned}\delta_\xi^a &= D_\xi \otimes I_\eta \otimes I_\zeta \otimes I_5 \\ \delta_\eta^a &= I_\xi \otimes D_\eta \otimes I_\zeta \otimes I_5 \\ \delta_\zeta^a &= I_\xi \otimes I_\eta \otimes D_\zeta \otimes I_5\end{aligned}$$

where the operator \otimes is the Kronecker product for matrices such that if $A \in M^{m \times n}$ and $B \in M^{p \times q}$ then $C = A \otimes B \in M^{mp \times nq}$ is defined by $C_{p(i-1)+k, q(j-1)+l} = A_{i,j} B_{k,l}$. I_ξ , I_η , I_ζ and I_5 are identity matrices of dimensions $n_\xi \times n_\xi$, $n_\eta \times n_\eta$, $n_\zeta \times n_\zeta$ and 5×5 respectively. D_ξ , D_η and D_ζ are finite-difference operators of dimensions $n_\xi \times n_\xi$, $n_\eta \times n_\eta$ and $n_\zeta \times n_\zeta$ respectively [72].

3.2.2 Grid metrics

The grid metrics (2.7) are calculated using the spatial discretization outlined in the previous section with $\Delta\xi = \Delta\eta = \Delta\zeta = 1$. However, if the grid metrics are calculated as shown in (2.7) then the metric invariants (2.9) are not necessarily satisfied. This means that a uniform free stream could fail to satisfy the Euler equations (2.11). This phenomenon could eventually result in an error creeping into our solution. There are two ways this problem can be overcome. One way is to subtract the free stream from the overall equations to ensure that a free stream is a solution to the Euler equations [60],

$$\partial_\tau \hat{Q} + \partial_\xi (\hat{E} - \hat{E}_\infty) + \partial_\eta (\hat{F} - \hat{F}_\infty) + \partial_\zeta (\hat{G} - \hat{G}_\infty) = 0 \quad (3.4)$$

In the second approach the grid metrics are modified analytically to ensure the metric invariants are satisfied. The approach implemented in this work exploits the fact that when finite-differences are used, the product rule is not satisfied numerically. Thus the metrics are calculated using a conservative form. For instance [21, 74]:

$$\xi_x = J[(y_\eta z)_\zeta - (y_\zeta z)_\eta] \quad (3.5)$$

Similar formulas can be obtained for the other metric terms. For this method to work, the finite-difference scheme used to calculate the metrics must be the same as that used to calculate the flux terms. One can now easily verify that the metric invariants are satisfied

by substituting the modified metrics into equation (2.9).

3.2.3 Artificial dissipation

Nonlinear convection problems such as the Euler equations that use central finite-differences require the use of artificial dissipation to damp out high frequency modes. In this work the dissipation operator was implemented so as that it meets the following criteria, as stated by Mattson et al. [36] :

1. Efficiently reduce spurious oscillations.
2. Order of accuracy should be the same or higher than the finite-difference scheme.
3. Computational work should be comparable to the finite-difference scheme, i.e. the stencil size of the dissipation operator should be similar to the finite-difference scheme, if not the same.
4. The dissipation operator should provide an energy estimate.

Two forms of dissipation were implemented, scalar and matrix dissipation. Errors in the solution with scalar dissipation tend to be higher because it uses dissipation that is higher than what is actually needed. The scalar dissipation model along the ξ direction is given by [59, 25, 58, 53]:

$$\delta_{\xi}^s \hat{Q} = \mathcal{H}^{-1} \tilde{\mathcal{D}}_- J^{-1} \sigma \left(\epsilon^{(2)} \tilde{\mathcal{D}}_+ - \epsilon \tilde{\mathcal{D}}_-^{-1} \tilde{\mathcal{D}}_p^T \mathcal{B} \tilde{\mathcal{D}}_p \right) J \hat{Q} \quad (3.6)$$

where the operators are given by,

$$\begin{aligned} \mathcal{H} &= H \otimes I_{\eta} \otimes I_{\zeta} \otimes I_5 \\ \tilde{\mathcal{D}}_- &= \tilde{D}_- \otimes I_{\eta} \otimes I_{\zeta} \otimes I_5 \\ \tilde{\mathcal{D}}_+ &= \tilde{D}_+ \otimes I_{\eta} \otimes I_{\zeta} \otimes I_5 \\ \tilde{\mathcal{D}}_p &= \tilde{D}_p \otimes I_{\eta} \otimes I_{\zeta} \otimes I_5 \\ \mathcal{B} &= B \otimes I_{\eta} \otimes I_{\zeta} \otimes I_5 \\ \sigma &= \text{diag}(\sigma_{1+\frac{1}{2},1,1}, \dots, \sigma_{j+\frac{1}{2},k,m}, \dots, \sigma_{n_{\xi},n_{\eta},n_{\zeta}}) \otimes I_5 \\ \epsilon^{(2)} &= \text{diag}(\epsilon_{1+\frac{1}{2},1,1}^{(2)}, \dots, \epsilon_{j+\frac{1}{2},k,m}^{(2)}, \dots, \epsilon_{n_{\xi},n_{\eta},n_{\zeta}}^{(2)}) \otimes I_5 \\ \epsilon &= \text{diag}(\epsilon_{1+\frac{1}{2},1,1}, \dots, \epsilon_{j+\frac{1}{2},k,m}, \dots, \epsilon_{n_{\xi},n_{\eta},n_{\zeta}}) \otimes I_5 \end{aligned}$$

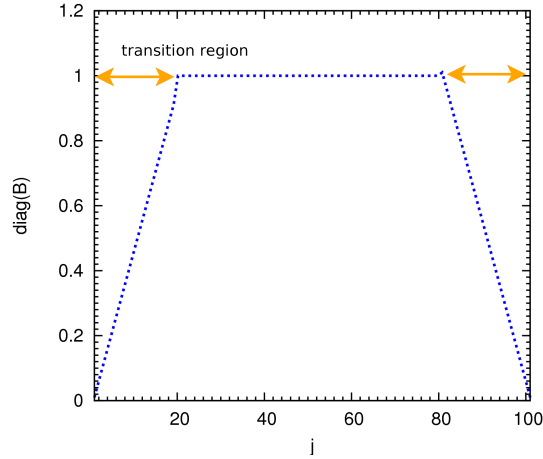


Figure 3.1: Distribution of the diagonal of B showing the interior and transition regions

transition region is the region where the diagonal of B increases from its boundary value to unity in the interior. The transition region is given by the percentage of total nodes that are to be included in this transition region. A typical distribution for the diagonal of B , including the transition region, is shown in Figure 3.1. The remaining variables are defined as

$$\begin{aligned} \Upsilon_{j,k,m} &= \frac{|p_{j+1,k,m} - 2p_{j,k,m} + p_{j-1,k,m}|}{|p_{j+1,k,m} + 2p_{j,k,m} + p_{j-1,k,m}|} \\ \epsilon &= \max(0, \kappa - \epsilon_2) \\ \epsilon_{j,k,m}^{(2)} &= \kappa_2 \max[\Upsilon_{j+1,k,m}, \Upsilon_{j,k,m}, \Upsilon_{j-1,k,m}] \\ \sigma &= |U| + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} + |V| + a\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2} + |W| + a\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2} \end{aligned}$$

Matrix dissipation is given by,

$$\delta_\xi^s \hat{Q} = \mathcal{H}^{-1} \tilde{\mathcal{D}}_- J^{-1} |A| \left(\epsilon^{(2)} \tilde{\mathcal{D}}_+ - \epsilon \tilde{\mathcal{D}}_-^{-1} \tilde{\mathcal{D}}_p^T \mathcal{B} \tilde{\mathcal{D}}_p \right) J \hat{Q} \quad (3.7)$$

where $|A|$ is block diagonal matrix given by,

$$|A| = \text{diag} \left(|A|_{1+\frac{1}{2},1,1}, \dots, |A|_{j+\frac{1}{2},k,m}, \dots, |A|_{n_\xi, n_\eta, n_\zeta} \right)$$

In this context, $A_{j+\frac{1}{2},k,m}$ is taken to mean A calculated for a Roe averaged state, and $|A|$ is given by [73]:

$$|A| = X |\Lambda| X^{-1} \quad (3.8)$$

where X is the eigenvector matrix of the flux Jacobian, and

$$|\Lambda| = \begin{pmatrix} |\lambda_1| & 0 & 0 & 0 & 0 \\ 0 & |\lambda_2| & 0 & 0 & 0 \\ 0 & 0 & |\lambda_3| & 0 & 0 \\ 0 & 0 & 0 & |\lambda_3| & 0 \\ 0 & 0 & 0 & 0 & |\lambda_3| \end{pmatrix}$$

where

$$\lambda_1 = U + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}$$

$$\lambda_2 = U - a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}$$

$$\lambda_3 = U$$

$$|\lambda_1| = \max(|\lambda_1|, V_n \rho)$$

$$|\lambda_2| = \max(|\lambda_2|, V_n \rho)$$

$$|\lambda_3| = \max(|\lambda_3|, V_l \rho)$$

$$\rho = |U| + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}$$

Typical values for κ_2 , κ , V_n and V_l are 1, 0.04, 0.25 and 0.25, respectively.

On adding the dissipation terms to the semi-discrete Euler equations (3.3), we get,

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} = 0 \quad (3.9)$$

3.2.4 Interfaces

The block interfaces are dealt with using simultaneous approximation terms (SAT's) [8, 7, 49, 19]. SAT's are a penalty method which, when implemented with SBP operators, provide an energy estimate. For convenience we shall consider an interface in the ξ direction. It is quite easy to obtain similar results for the other two directions.

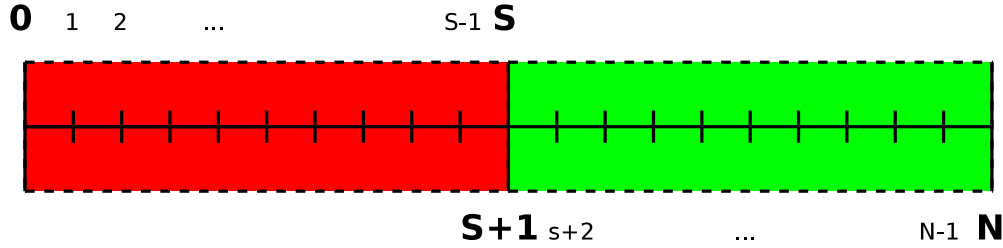


Figure 3.2: Figure showing two blocks with boundaries and an interface

Consider that in the ξ direction the domain has been split into two blocks with boundaries at $j = 0$ and $j = N$ and an interface at $j = s$ and $j = s + 1$, as shown in Figure 3.2. The semi-discrete Euler equations with interface SAT's for the left and right blocks are given by

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} = bc_s \quad (3.10)$$

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} = bc_{s+1} \quad (3.11)$$

where the SAT penalty terms are:

$$bc_s = -\tau_{bc} \mathcal{H}^{-1} J^{-1} \frac{(|A| - A)}{2} (e_s \otimes (Q_s - Q_{s+1}))$$

$$bc_{s+1} = -\tau_{bc} \mathcal{H}^{-1} J^{-1} \frac{(|A| + A)}{2} (e_{s+1} \otimes (Q_{s+1} - Q_s))$$

with

$$|A| = \text{diag} \left(|A|_{1,1,1}, \dots, |A|_{j,k,m}, \dots, |A|_{n_\xi, n_\eta, n_\zeta} \right)$$

$$A = \text{diag} \left(A_{1,1,1}, \dots, A_{j,k,m}, \dots, A_{n_\xi, n_\eta, n_\zeta} \right)$$

A is the flux Jacobian and $|A|$ is given by equation (3.8). The variables $e_s = [0, 0, \dots, 1]^T$ and $e_{s+1} = [1, 0, \dots, 0]^T$ are column vectors of dimensions $n_\xi \times 1$, where n_ξ is the number of points in the ξ direction, which maybe different for each block. Q_s and Q_{s+1} are column vectors of dimensions $5n_\eta n_\zeta \times 1$. τ_{bc} is a scalar that scales the interface terms.

3.2.5 Boundary conditions

Like the interfaces, the boundary conditions have been implemented using SAT's [8, 7, 49, 19, 33]. For convenience, the boundary conditions are shown for the ξ direction and can

be easily extended to the other two directions. For the domain shown in Figure 3.2, the semi-discrete Euler equations with boundary conditions are given by

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} = bc_0 \quad (3.12)$$

$$\partial_\tau \hat{Q} + \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} = bc_N \quad (3.13)$$

where

$$bc_0 = -\tau_{bc} \mathcal{H}^{-1} J^{-1} \frac{(|A| + A)}{2} (e_0 \otimes (Q_0 - Q_{bc}))$$

$$bc_N = -\tau_{bc} \mathcal{H}^{-1} J^{-1} \frac{(|A| - A)}{2} (e_N \otimes (Q_N - Q_{bc}))$$

$e_0 = [1, 0, \dots, 0]^T$ and $e_N = [0, 0, \dots, 1]^T$ are column vectors of dimensions $n_\xi \times 1$, where n_ξ is the number of points in the ξ direction, which maybe different for each block. Q_0 , Q_N and Q_{bc} are column vectors of dimensions $5n_\eta n_\zeta \times 1$. Q_{bc} contains the flow variables that need to be imposed at the boundaries. Also one can easily see that equations (3.12) and (3.13) are a mathematical representation of Riemann boundary conditions.

Far-field boundary conditions

For the far-field boundary conditions, Q_{bc} is set to the free stream conditions. This suffices for wings because far-field boundaries are generally placed far away from the wing such that the lift and drag are constant to within a certain tolerance [72].

Wall boundary conditions

The wall boundary condition is implemented by setting the velocity of the flow normal to the wall to zero. The velocity components u , v and w need to be transformed from the Cartesian coordinate system into a coordinate system with two vectors parallel to the surface and one vector normal to the surface. The vectors parallel to ξ , η and ζ are given by [27, 47]:

$$\hat{t}^{(\xi)} = \frac{x_\xi \hat{x} + y_\xi \hat{y} + z_\xi \hat{z}}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} \quad \hat{t}^{(\eta)} = \frac{x_\eta \hat{x} + y_\eta \hat{y} + z_\eta \hat{z}}{\sqrt{x_\eta^2 + y_\eta^2 + z_\eta^2}} \quad \hat{t}^{(\zeta)} = \frac{x_\zeta \hat{x} + y_\zeta \hat{y} + z_\zeta \hat{z}}{\sqrt{x_\zeta^2 + y_\zeta^2 + z_\zeta^2}} \quad (3.14)$$

The vectors normal to ξ and η , η and ζ and ξ and ζ are given by [27, 47],

$$\hat{n}^{(\xi\eta)} = \frac{\zeta_x \hat{x} + \zeta_y \hat{y} + \zeta_z \hat{z}}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \quad \hat{n}^{(\eta\zeta)} = \frac{\xi_x \hat{x} + \xi_y \hat{y} + \xi_z \hat{z}}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}} \quad \hat{n}^{(\xi\zeta)} = \frac{\eta_x \hat{x} + \eta_y \hat{y} + \eta_z \hat{z}}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}} \quad (3.15)$$

If the wall surface is normal to η and ξ , the tangential velocities V_{T1} and V_{T2} and the normal velocity V_N are given by [27, 47]:

$$\begin{bmatrix} V_{T1} \\ V_{T2} \\ V_N \end{bmatrix} = T^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where $T = [\vec{t}^{(\xi)} | \vec{t}^{(\eta)} | \vec{n}^{(\xi\eta)}]$ and is explicitly given by:

$$T = \begin{pmatrix} \frac{x_\xi}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} & \frac{x_\eta}{\sqrt{x_\eta^2 + y_\eta^2 + z_\eta^2}} & \frac{\zeta_x}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \\ \frac{y_\xi}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} & \frac{y_\eta}{\sqrt{x_\eta^2 + y_\eta^2 + z_\eta^2}} & \frac{\zeta_y}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \\ \frac{z_\xi}{\sqrt{x_\xi^2 + y_\xi^2 + z_\xi^2}} & \frac{z_\eta}{\sqrt{x_\eta^2 + y_\eta^2 + z_\eta^2}} & \frac{\zeta_z}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \end{pmatrix}$$

V_N can now be set to zero and the velocities transformed back to the Cartesian coordinate system. The density remains unchanged and the new energy is given by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2} \rho (V_{T1}^2 + V_{T2}^2) \quad (3.16)$$

This may be done because $\|T\|_2 = 1$, and thus Q_{bc} for a wall boundary condition is obtained.

3.2.6 Stability

The equations derived thus far do not guarantee stability for the three-dimensional Euler equations. However, when the same formulation is applied to the linear convection equation, it provides an energy estimate and thus guarantees stability. Stability of the linear convection equation is necessary but not sufficient to guarantee stability for the three-dimensional Euler equations [30, 13, 36, 71, 34].

3.3 Time marching

Using the spatial discretization scheme from the previous section we obtain

$$\partial_\tau \hat{Q} = -R(\hat{Q}) \quad (3.17)$$

where $R(\hat{Q})$ is known as the residual and is given by

$$R(\hat{Q}) = \delta_\xi^a \hat{E} + \delta_\xi^s \hat{Q} + \delta_\eta^a \hat{F} + \delta_\eta^s \hat{Q} + \delta_\zeta^a \hat{G} + \delta_\zeta^s \hat{Q} - bc$$

Until now no assumptions have been made about the time accuracy of the system and thus one could easily use the RK4 method with fourth-order spatial discretization to obtain a fourth-order accurate method in time and space [35]. This work is concerned with steady flows and thus the equation we aim to solve is

$$R(\hat{Q}) = 0 \quad (3.18)$$

An iterative method is used to solve equation (3.18), which can be solved using the implicit Euler method through local linearization of $R(\hat{Q})$ [32]:

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}} \right)^n \right] \Delta \hat{Q}^n = -R^n \quad (3.19)$$

where the superscript n is the outer iteration number. If we let $\Delta t \rightarrow \infty$, Newton's method, which converges quadratically, is obtained. However, if the initial guess is not close to the solution, then Newton's method can fail. Hence, the solution process in this work uses two stages, a start-up phase to find a suitable iterate and an inexact-Newton phase. Before we describe the two phases, we shall lay down some of the background.

3.3.1 Relaxation

For each solution update, we have the following equation

$$\hat{Q}^{n+1} = \hat{Q}^n + \theta \Delta \hat{Q}^n \quad (3.20)$$

where θ is the relaxation factor. If $\theta < 1$ then it is known as under relaxation, and if $\theta > 1$ is known as over relaxation. Under relaxation helps in preventing solution updates during the start-up phase that would cause the solution to become non-physical.

3.3.2 Preconditioner

In this work we utilize GMRES [64, 66]. The performance of GMRES can be improved if the system is preconditioned. The system in equation (3.19) when right preconditioned with M becomes

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}} \right)^n \right] M^{-1} M \Delta \hat{Q}^n = -R^n \quad (3.21)$$

From equation (3.21), we see that M^{-1} needs to be calculated. This is accomplished by performing an incomplete lower-upper factorization (ILU) of fill number k such that

$$M = LU + error \quad (3.22)$$

where L is a lower triangular matrix, and U is an upper triangular matrix [65]. As the fill level k is increased, the factorization becomes more accurate. The optimal fill level shall be investigated as a part of this work.

We choose M to be the ILU factorization of the first-order approximate Jacobian. This is preferred because the resultant ILU factors are better conditioned, and the first-order Jacobian requires less memory and is faster to calculate and factorize [57, 40, 41, 38, 39].

The high-order dissipation coefficient ϵ is lumped with the first-order dissipation coefficient $\epsilon^{(2)}$ in the preconditioner using σ [19, 57, 47]:

$$\epsilon^{(2)} = \epsilon^{(2)} + \sigma \epsilon \quad (3.23)$$

Optimal values for σ shall be investigated as a part of this work.

Also the additive Schwarz and approximate Schur parallel preconditioners have been used [19, 65].

3.3.3 Local time stepping

To speed up convergence a spatially varying time step is used [19],

$$\Delta t_{j,k,m} = \frac{\Delta t_{ref}^n}{J_{j,k,m} (1 + \sqrt[3]{J_{j,k,m}})} \quad (3.24)$$

This approximately represents a constant CFL number. The reference time step Δt_{ref} is chosen differently for the start-up and inexact-Newton phases.

3.3.4 Reordering of unknowns

The nodes are reordered using the reverse Cuthill-McKee (RCM) method. This reduces the bandwidth of the system and also improves the conditioning of the ILU factors [57, 47, 11]. The root is chosen to be the node with indices (n_ξ, n_η, n_ζ) on each block, which lies on the downstream boundary of the block.

3.3.5 Matrix-free GMRES

GMRES requires matrix vector products. Because of the size and the difficulty in calculating the higher-order Jacobian, we utilize matrix-free GMRES. In matrix-free GMRES, the matrix-vector products are approximated with a finite-difference [4, 57, 40, 41, 38, 39, 19, 47]:

$$\frac{\partial R(\hat{Q})}{\partial \hat{Q}} v \approx \frac{R(\hat{Q} + \epsilon v) - R(\hat{Q})}{\epsilon} \quad (3.25)$$

where

$$\epsilon = \sqrt{\frac{N\delta}{v^T v}}$$

where N is the number of unknowns, and $\delta = 10^{-13}$ [19].

3.3.6 Start-up phase

The idea behind using the start-up phase is to go through the transient as quickly as possible, in other words, to provide a suitable iterate so that we can switch to the inexact-Newton phase. Since we are performing defect correction we modify equation (3.19) and solve the

following equation [19, 57, 40, 41, 38, 39]:

$$\left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}} \right)_1 \right] \Delta \hat{Q} = -R \quad (3.26)$$

where $\left(\frac{\partial R}{\partial \hat{Q}} \right)_1$ is the first-order Jacobian. Equation (3.26) is not solved exactly but to a relative tolerance of ω such that [19]:

$$\left\| R + \left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}} \right)_1 \right] \Delta \hat{Q} \right\|_2 \leq \omega \|R\|_2 \quad (3.27)$$

The parameter ω is normally chosen to be 0.5. Also since factorizing is expensive, we use a lagged Jacobian update [19] i.e. we update the Jacobian every couple of iterations. The reference time step is chosen such that [19]:

$$\Delta t_{ref}^n = a(b)^n \quad (3.28)$$

Typical values for a and b are 0.01 and 1.3, respectively.

3.3.7 Inexact-Newton phase

The switch to the inexact-Newton phase is made once the residual has fallen below a certain value specified by τ such that [19]:

$$\|R^n\|_2 \leq \tau \|R^0\|_2 \quad (3.29)$$

The parameter τ is chosen to be $\frac{1}{15}$. Equation (3.19) is now solved with a tolerance ω such that [19]:

$$\left\| R - \left[\frac{I}{\Delta t} + \left(\frac{\partial R}{\partial \hat{Q}} \right) \right] \Delta \hat{Q} \right\|_2 \leq \omega \|R\|_2 \quad (3.30)$$

where

$$\omega_n = \max \left(0.01, \omega_{n-1}^{\frac{1+\sqrt{5}}{2}} \right) \quad (3.31)$$

The reference time step is chosen such that [19]:

$$\Delta t_{ref}^n = \max \left(\alpha \left(\frac{\|R^n\|_2}{\|R^0\|_2} \right)^{-\beta}, \Delta t_{ref}^{n-1} \right) \quad (3.32)$$

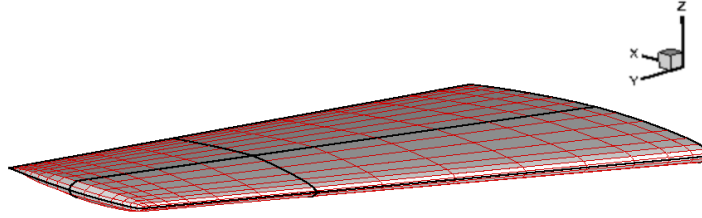


Figure 3.3: Figure showing the surface of a wing for which the lift and drag are to be calculated

where

$$\alpha = a(b)^n \left(\frac{\|R^k\|_2}{\|R^0\|_2} \right)^\beta \quad (3.33)$$

and k is the index of the last start-up phase iteration, and $\beta = 2$.

3.4 Force Integration

Consider the wing shown in Figure 3.3. The chord of the wing is aligned with the x axis and the span of the wing is aligned with the y axis. Now if the angle of attack is α , and we assume that the coordinates ξ and η are parallel to the surface, then the lift and drag coefficients are given by

$$C_D = \frac{1}{S} \oint C_p \vec{r} \cdot d\vec{S} \quad (3.34)$$

$$C_L = \frac{1}{S} \oint C_p \vec{s} \cdot d\vec{S} \quad (3.35)$$

where

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty V_\infty^2}$$

$$\vec{r} = \sin \alpha \hat{x} + \cos \alpha \hat{z}$$

$$\vec{s} = \cos \alpha \hat{x} - \sin \alpha \hat{z}$$

$$d\vec{S} = \hat{n}^{(\xi\eta)} |\vec{p}^{(\xi)} \times \vec{p}^{(\eta)}| d\xi d\eta$$

where S is the surface area, and $\hat{n}^{(\xi\eta)}$ is the surface normal vector and is defined in (3.15). The vectors $\vec{t}^{(\xi)} = x_\xi \hat{x} + y_\xi \hat{y} + z_\xi \hat{z}$ and $\vec{t}^{(\eta)} = x_\eta \hat{x} + y_\eta \hat{y} + z_\eta \hat{z}$ are parallel to the surface and are similar to (3.14) except they are not unit vectors. Using $\vec{t}^{(\xi)} \times \vec{t}^{(\eta)} = \zeta_x \hat{x} + \zeta_y \hat{y} + \zeta_z \hat{z}$, we can simplify $d\vec{S}$ to,

$$d\vec{S} = (\zeta_x \hat{x} + \zeta_y \hat{y} + \zeta_z \hat{z}) d\xi d\eta$$

The integrals in equation (3.35) are calculated using Simpson's rule which is a fourth-order accurate method given by [6]:

$$\int \int f(x, y) dx dy = \frac{1}{9} \Delta x \Delta y \sum_{i=0}^M \sum_{j=0}^N \alpha_i \alpha_j f(i \Delta x, j \Delta y) \quad (3.36)$$

where

$$\alpha_1 = 1, \alpha_2 = 4, \alpha_3 = 2, \alpha_4 = 4, \dots, \alpha_{N-3} = 4, \alpha_{N-2} = 2, \alpha_{N-1} = 4, \alpha_N = 1$$

Chapter 4

Results and Discussion

This chapter demonstrates the algorithm described in the previous chapter for various test cases. It begins by describing tests such as a quasi one-dimensional nozzle, the method of manufactured solutions (MMS), Ringleb flow and supersonic and subsonic vortices. Using these tests we can rigorously test the algorithm. Matrix and scalar dissipation are compared early on and then order of accuracy studies that include functionals are performed. Finally, the algorithm is validated for the ONERA M6 wing, at subsonic and transonic Mach numbers.

4.1 Test cases

4.1.1 Quasi-one-dimensional nozzle

For the quasi-one-dimensional nozzle, the problem is specified by specifying the cross-sectional area variation and the boundary conditions. The cross-sectional area variation for the subsonic nozzle case is given by:

$$S(x) = 2 - \exp\left(\frac{-(x-5)^2}{2(25)^2}\right) \quad 0 \leq x \leq 10 \quad (4.1)$$

and is shown in Figure 4.1(a). This is an inverted Gaussian function. This function is used because it has an infinite number of defined derivatives. At the inlet and outlet, $P = 82,168$ Pa, $\rho = 1.0094 \text{ Kg/m}^3$ and $u = 181.35 \text{ m/s}$, which corresponds to a subsonic case. For the

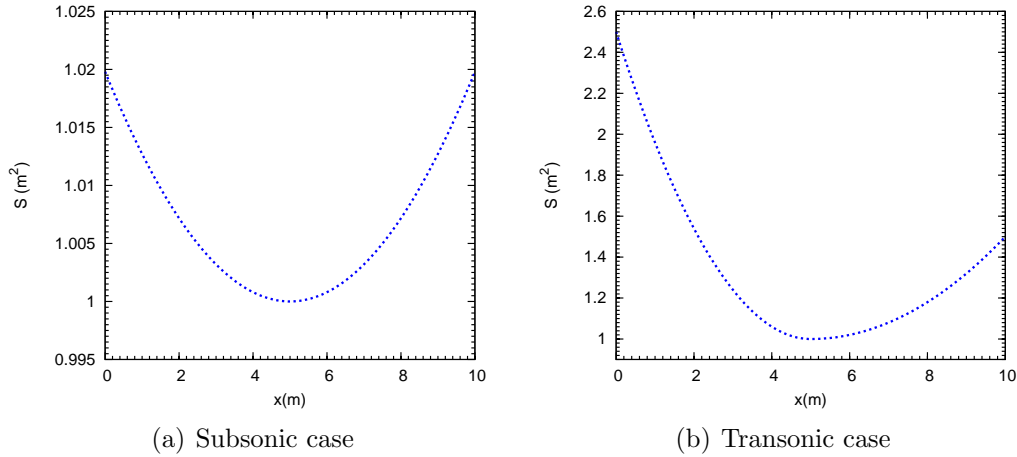


Figure 4.1: Cross-sectional area variation for the subsonic and transonic cases

transonic case, the cross-sectional area variation is given by [20]:

$$S(x) = 1 + \frac{3}{2} \left(1 - \frac{x}{5}\right)^2 \quad 0 \leq x \leq 5 \quad (4.2)$$

$$S(x) = 1 + \frac{1}{2} \left(1 - \frac{x}{5}\right)^2 \quad 5 \leq x \leq 10 \quad (4.3)$$

and is shown in Figure 4.1(b). At the inlet $P = 96,084.91$ Pa, $\rho = 1.1288 \text{ Kg/m}^3$ and $u = 82.69 \text{ m/s}$ and at the outlet $P = 84,973.84$ Pa, $\rho = 1.0261 \text{ Kg/m}^3$ and $u = 151.61 \text{ m/s}$. The boundary conditions are set by setting Q_{bc} of Section 3.2.5 to the exact solution.

4.1.2 Method of manufactured solutions

The method of manufactured solutions can be used to verify the prescribed order of accuracy for computational codes [61]. It works by manufacturing a solution by adding appropriate source terms to the governing equations. The six steps outlined by C. J. Roy et al. [62] for implementing the method of manufactured solutions for error convergence studies are:

1. Choose the form of the governing equations.
2. Choose the form of the manufactured solution.
3. Apply the governing equations to the manufactured solution to generate analytical source terms.
4. Discretize the equations and solve on multiple mesh levels using analytical boundary conditions and source terms from the manufactured solution.

Quantity ϕ	f_x	f_y	f_z	ϕ_o	ϕ_x	ϕ_y	ϕ_z	$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$	L
ρ	sin	cos	cos	1	0.1	0.15	0.15	0.75	1.0	1.0	1.0
u	sin	cos	cos	$\frac{70}{100\sqrt{14}}$	$\frac{4}{100\sqrt{14}}$	$\frac{-12}{100\sqrt{14}}$	$\frac{-8}{100\sqrt{14}}$	$\frac{5}{3}$	1.5	1.0	1.0
v	cos	sin	cos	$\frac{90}{100\sqrt{14}}$	$\frac{-20}{100\sqrt{14}}$	$\frac{4}{100\sqrt{14}}$	$\frac{3}{100\sqrt{14}}$	1.5	1.0	0.75	1.0
w	cos	sin	cos	$\frac{40}{100\sqrt{14}}$	$\frac{-2}{100\sqrt{14}}$	$\frac{2}{100\sqrt{14}}$	$\frac{3}{100\sqrt{14}}$	1.0	1.5	1.0	1.0
p	cos	sin	sin	$\frac{1}{1.4}$	$\frac{-0.3}{1.4}$	$\frac{0.2}{1.4}$	$\frac{0.05}{1.4}$	1.0	1.25	1.0	1.0

Table 4.1: Constants and form of the solution for three-dimensional flow using the method of manufactured solutions

5. Evaluate the global discretization error in the numerical solutions.
6. Determine whether or not the observed order of accuracy matches the formal order of accuracy.

In this work we shall be using the Euler equations (2.2). We shall let the form of the solution for three-dimensional flow be

$$\phi(x, y, z) = \phi_o + \phi_x f_x \left(\frac{a_{\phi x} \pi x}{L} \right) + \phi_y f_y \left(\frac{a_{\phi y} \pi y}{L} \right) + \phi_z f_z \left(\frac{a_{\phi z} \pi z}{L} \right) \quad (4.4)$$

where ϕ is ρ , u , v , w , or p . The form of the solution f_x , f_y , f_z and the constants ϕ_o , ϕ_x , ϕ_y , $a_{\phi x}$, $a_{\phi y}$, $a_{\phi z}$ and L are given in Table 4.1. These constants have been chosen in order to keep the flow physical, i.e. $p > 0$ and $\rho > 0$. Now that the form of the solution has been specified, the governing equations are applied to the manufactured solution to generate the analytical source terms. Thus equation (2.11) now becomes

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = \hat{\psi} \quad (4.5)$$

where $\hat{\psi}$ is the source term. As an example, for two-dimensional flow the source term in the equation of continuity is given by,

$$\begin{aligned} J\hat{\psi}_{continuity} = & \frac{u\rho_x a_{\rho x} \pi}{L} \cos \left(\frac{a_{\rho x} \pi x}{L} \right) + \frac{\rho u_x a_{u x} \pi}{L} \cos \left(\frac{a_{u x} \pi x}{L} \right) \\ & - \frac{v\rho_y a_{\rho y} \pi}{L} \sin \left(\frac{a_{\rho y} \pi y}{L} \right) + \frac{\rho v_y a_{v y} \pi}{L} \cos \left(\frac{a_{v y} \pi y}{L} \right) \end{aligned} \quad (4.6)$$

The other source terms can be derived similarly.

The domain for the problem is shown in Figure 4.2. The domain consists of eight blocks such that $0 \leq x \leq 1$, $0 \leq y \leq 1$ and $0 \leq z \leq 1$. Each block has a mesh density ranging from $13 \times 13 \times 13$ nodes to $65 \times 65 \times 65$ nodes. The boundary conditions are implemented by setting

Q_{bc} of Section 3.2.5 to the exact solution.

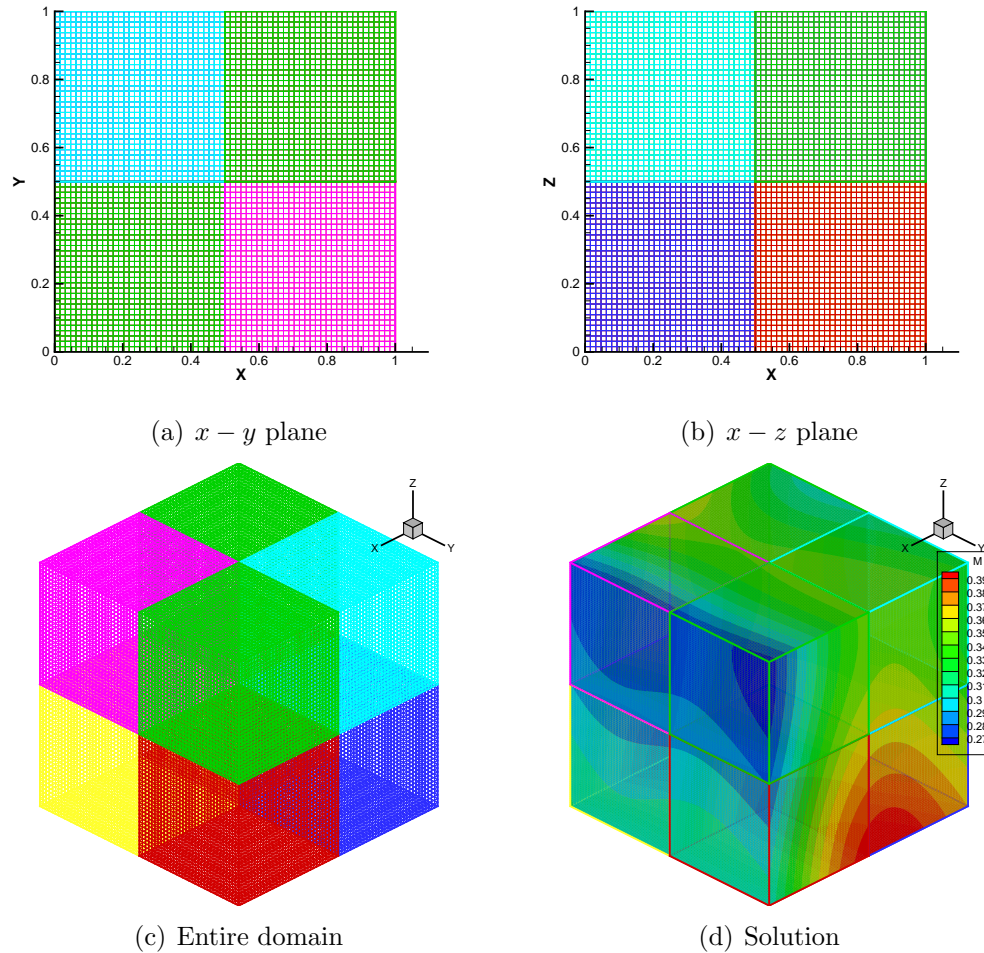


Figure 4.2: Domain and solution for the three-dimensional method of manufactured solutions

4.1.3 Ringleb flow

Ringleb flow is an exact solution to the two-dimensional Euler equations [67, 23, 10, 9]. The flow lies between two streamlines. Hence, in addition to testing the order of accuracy of the flow solver, Ringleb flow can also be used to test the wall boundary conditions. The solution

is obtained using a hodograph transformation and is given by [67, 23, 10, 9]:

$$x(q, \mu) = \frac{1}{2\rho} \left(\frac{2}{\mu^2} - \frac{1}{q^2} \right) - \frac{C}{2} \quad (4.7)$$

$$y(q, \mu) = \pm \frac{1}{\mu\rho q} \sqrt{1 - \left(\frac{q}{\mu} \right)^2} \quad (4.8)$$

$$a = \sqrt{1 - \frac{\gamma - 1}{2} q^2} \quad (4.9)$$

$$\rho = a^{\frac{2}{\gamma-1}} \quad (4.10)$$

$$C = \frac{1}{a} + \frac{1}{3a^3} + \frac{1}{5a^5} - \frac{1}{2} \log \frac{1+a}{1-a} \quad (4.11)$$

where ρ is the density, x and y are the Cartesian coordinates, q is the nondimensional velocity magnitude, a is the sound speed, and μ is the streamline constant. The velocity is nondimensionalized with respect to the stagnation sound speed. The flow angle θ is by,

$$\theta = 2\pi - \sin^{-1} \left(\frac{q}{\mu} \right) \quad (4.12)$$

Lastly, the energy can be obtained by using isentropic relationships.

The entire flow can be parameterized using μ and q . In this work μ and q were chosen such that $1 \leq \mu \leq 1.5$ and $0.5 \leq q \leq 0.75$. The flow domain and solution are shown in Figure 4.3. The domain consists of four blocks with a mesh density ranging from 13×13 nodes to 67×67 nodes per block. Inlet, outflow and wall boundary conditions are imposed as shown in Figure 4.3. The inlet and outlet boundary conditions are imposed by setting Q_{bc} of Section 3.2.5 to the exact solution. Since the flow solver in this work is three-dimensional, symmetry boundary conditions are imposed on the $x - y$ planes.

4.1.4 Vortex

The vortex is an isentropic annular flow. An exact solution is known for this flow, and it can be used to test the order of accuracy and boundary conditions. The solution is given by [1, 2, 42]:

$$\rho(r) = \rho_i \left(1 + \frac{\gamma - 1}{2} M_i^2 \left(1 - \frac{r_i^2}{r} \right) \right)^{\frac{1}{\gamma-1}} \quad (4.13)$$

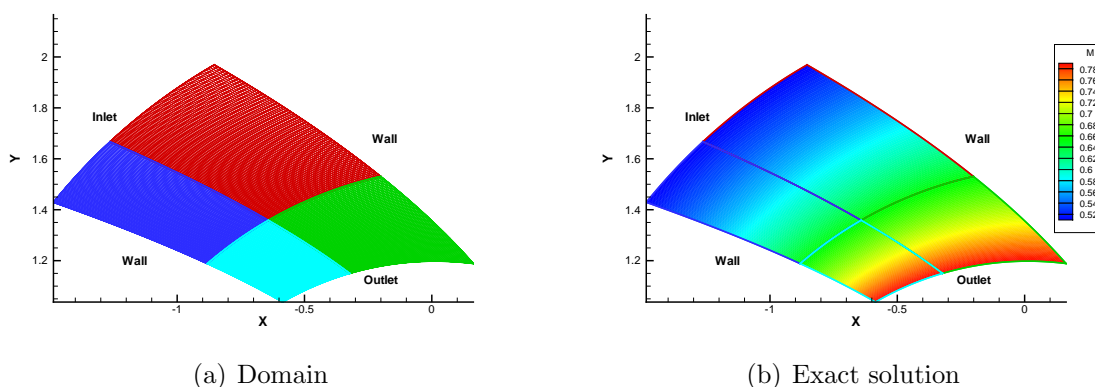


Figure 4.3: Domain and exact solution for Ringleb flow

where r is the radius, ρ is the density, M is the Mach number, and the subscript i stands for values at the inner radius shown in Figure 4.4. For this work r_i is chosen to be 1, M_i is chosen to be 2.0, ρ_i is chosen to be 1.0 and p_i is chosen to be $1/\gamma$. The other quantities may be obtained by using isentropic relationships. We set up two problems, one which has transonic flow, and another which has subsonic flow, as shown in Figure 4.5. Both domains consist of four blocks with mesh densities ranging from 13×13 nodes to 65×65 nodes per block. Inlet, outflow and wall boundary conditions are imposed as shown in Figure 4.5. The inlet and outlet boundary conditions are imposed by setting Q_{bc} of Section 3.2.5 to the exact solution. Since this is two-dimensional flow, as before, symmetry boundary conditions are imposed on the $x - y$ planes.

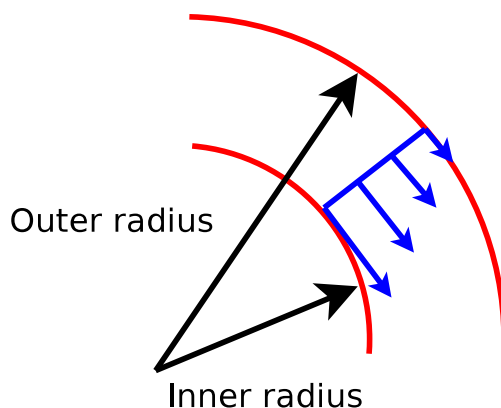
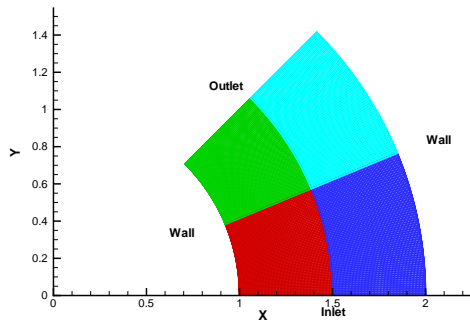
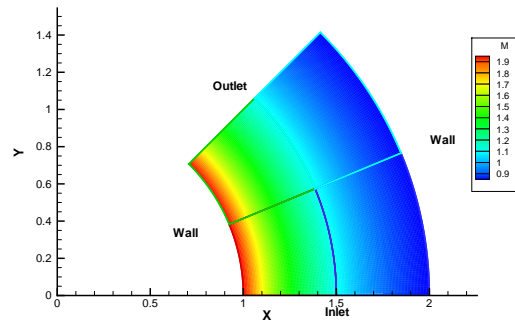


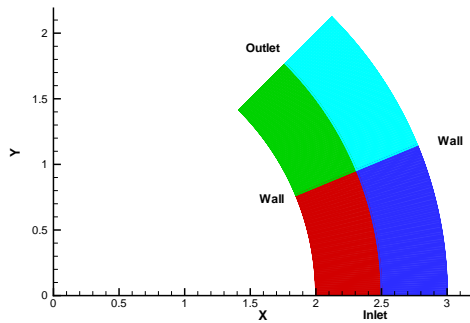
Figure 4.4: Vortex



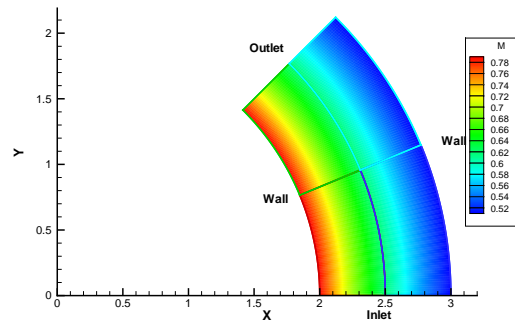
(a) Transonic vortex domain



(b) Transonic vortex exact solution



(c) Subsonic vortex domain



(d) Subsonic vortex exact solution

Figure 4.5: Domain and exact solution for the vortex problem

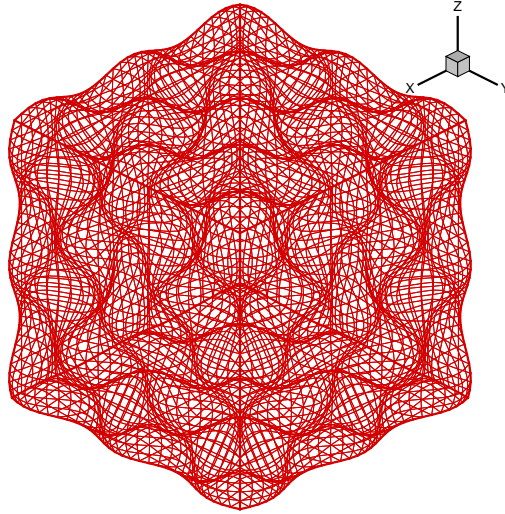


Figure 4.6: Grid used to verify the grid metrics

4.2 Preservation of free stream flow

In Section 3.2.2, a method for calculating the metrics was outlined such that the metric invariants (2.9) are satisfied. To check if the metric invariants are satisfied, we impose a free stream flow on an arbitrary geometry and calculate $\left\|R(\hat{Q})\right\|_2$ to see if the free stream is a solution. The geometry used is defined by the following functions:

$$\begin{aligned} x_{j,k,m} &= \frac{j}{5N} + \frac{1}{100} \sin \frac{10\pi k}{N} \sin \frac{10\pi m}{N} \\ y_{j,k,m} &= \frac{k}{5N} + \frac{1}{100} \sin \frac{10\pi j}{N} \sin \frac{10\pi m}{N} \\ z_{j,k,m} &= \frac{m}{5N} + \frac{1}{100} \sin \frac{10\pi j}{N} \sin \frac{10\pi k}{N} \end{aligned} \quad (4.14)$$

where N is the total number of points in each direction. The geometry for $N=33$ is shown in Figure 4.6. The data for $N=33$ and $N=129$ are shown in Tables 4.2 and 4.3 respectively. The data are shown for both scalar and matrix dissipation. We see that the residuals are machine zero and thus the free stream is preserved, hence establishing that the metric invariants are satisfied.

	Scalar dissipation	Matrix dissipation
2 nd order	0.11343E-17	0.11344E-17
3 rd order	0.29738E-16	0.29738E-16
4 th order	0.37564E-16	0.37562E-16

Table 4.2: Residual norms for $N=33$

	Scalar dissipation	Matrix dissipation
2 nd order	0.52514E-18	0.52513E-18
3 rd order	0.34625E-16	0.34625E-16
4 th order	0.48092E-16	0.48092E-16

Table 4.3: Residual norms for $N=129$

4.3 Matrix and scalar dissipation

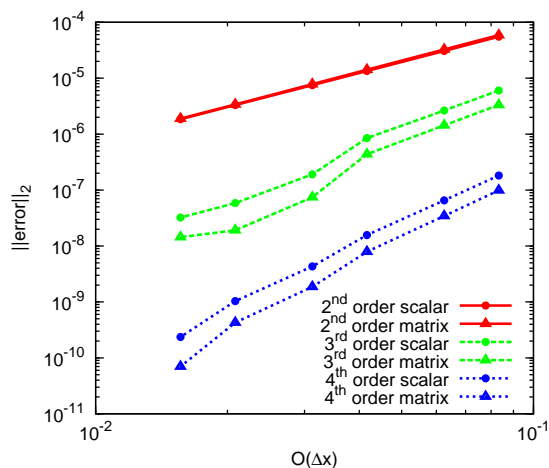
To compare matrix and scalar dissipation we look at the quasi-one-dimensional nozzle, the supersonic vortex, the subsonic vortex, the method of manufactured solutions, and a transonic flow case using the ONERA M6 wing. The results for the quasi-one-dimensional nozzle, the vortices and the method of manufactured solutions are shown in Figure 4.7. The errors for the the quasi-one-dimensional nozzle are based on the L_2 norm of the error in velocity and for the other test cases they are based on the L_2 norm of the error in density. The variable plotted on the x -axis is $O(\Delta x) = \frac{1}{N-1}$, where N is the number of nodes in the x coordinate direction.

As can be seen there is not a major advantage in using matrix dissipation for the second-order method. However, for the higher-order methods, matrix dissipation reduces the error noticeably.

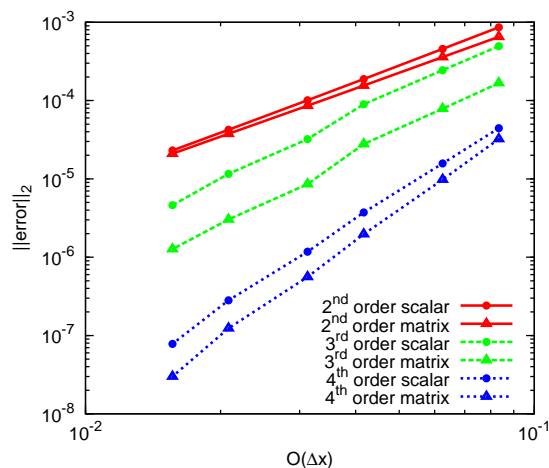
Moving to non-smooth solutions, the transonic solution for the quasi-one-dimensional nozzle is shown in Figure 4.8. From the data it can be seen that for all three methods matrix dissipation performs better near the shock as one would expect.

We next look at a transonic test case using the ONERA M6 wing at a Mach number of 0.84 and an angle of attack of 3.06° . For this test case we use a grid with 96 blocks and 4913 nodes per block. The C_p plots at different sections along the span of the wing are shown in Figures 4.9, 4.10 and 4.11 for the second, third and fourth-order methods. Comparing the different C_p plots against the results on a finer mesh, one can see that matrix dissipation performs somewhat better in the vicinity of shocks.

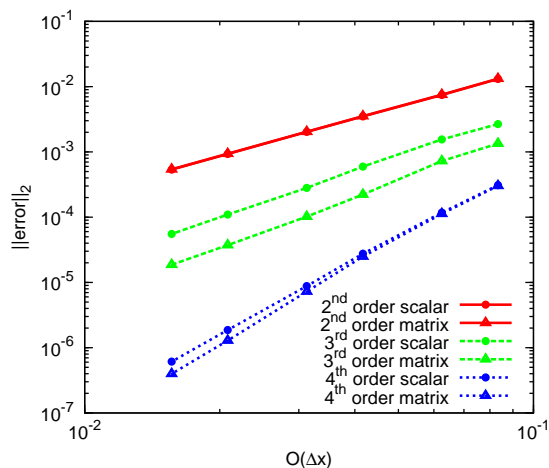
Lastly we compare the convergence histories for the transonic flow case. This is shown in Figure 4.12. Matrix dissipation performs better for all the three methods. Based on the



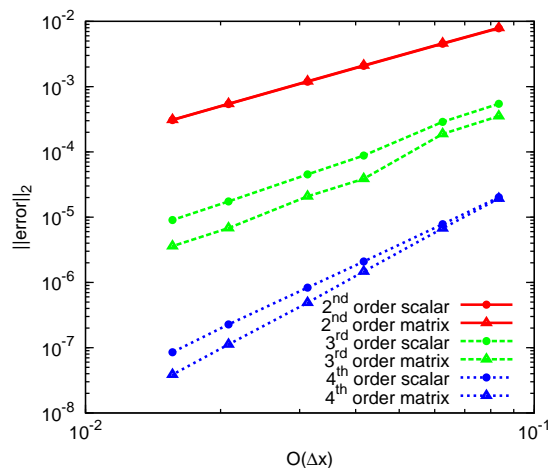
(a) Quasi-one-dimensional nozzle



(b) MMS



(c) Supersonic vortex



(d) Subsonic vortex

Figure 4.7: Matrix and scalar dissipation

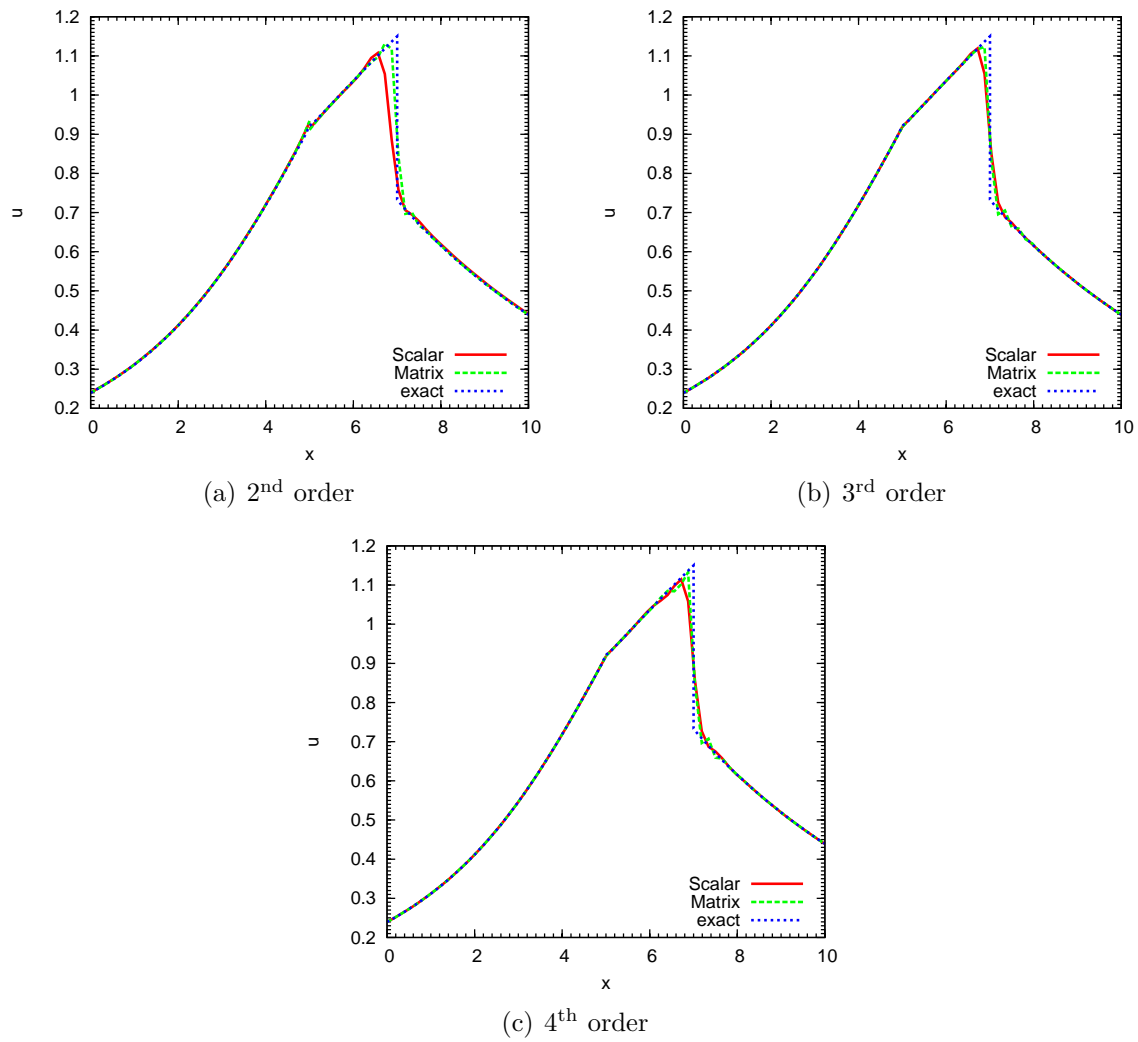


Figure 4.8: Matrix and scalar dissipation - Quasi-one-dimensional nozzle - Transonic case - 2 blocks \times 33 nodes

results in this section, only matrix dissipation shall be studied in the sections to follow.

4.4 Order of accuracy

The order of accuracy of each scheme is tested using the quasi-one-dimensional nozzle, the method of manufactured solutions, the subsonic and supersonic vortices, and Ringleb flow. The order of accuracy for the different test cases is shown in Figure 4.13, and the slopes for the lines of best fit to the data are shown in Table 4.4. The errors shown in Table 4.4 are errors in the slopes of the lines of best fit. The errors for the the quasi-one-dimensional nozzle are based on the L_2 norm of the error in velocity, and for the other test cases they are based on the L_2 norm of the error in density. It can be seen that the observed order of accuracy agrees well with the formal order of accuracy. In addition to establishing the order of accuracy, these results verify that there are no bugs in the spatial discretization.

4.5 Functionals

To obtain high-order accuracy for functionals such as drag and lift, both the flow solver and the integration method should employ high-order methods. This concept shall be examined in this section. High-order accuracy of the flow solver has been established in the previous sections. We now begin by studying the order of accuracy for the integration method alone and then look at the combination of the flow solver and the integration method.

4.5.1 Integration method

We analyze the order of accuracy of the integration method on a manufactured solution on a fictitious surface. The fictitious surface is defined by the vector \vec{r} :

$$\vec{r} = \left(1 - \frac{1}{2}z\right) \cos \theta \hat{x} + \left(1 - \frac{1}{2}z\right) \sin \theta \hat{y} + z \hat{z} \quad 0 \leq \theta \leq \frac{\pi}{2}, 0 \leq z \leq 1 \quad (4.15)$$

The functional is defined by f :

$$f = \oint C_p \hat{x} \cdot d\vec{S} \quad (4.16)$$

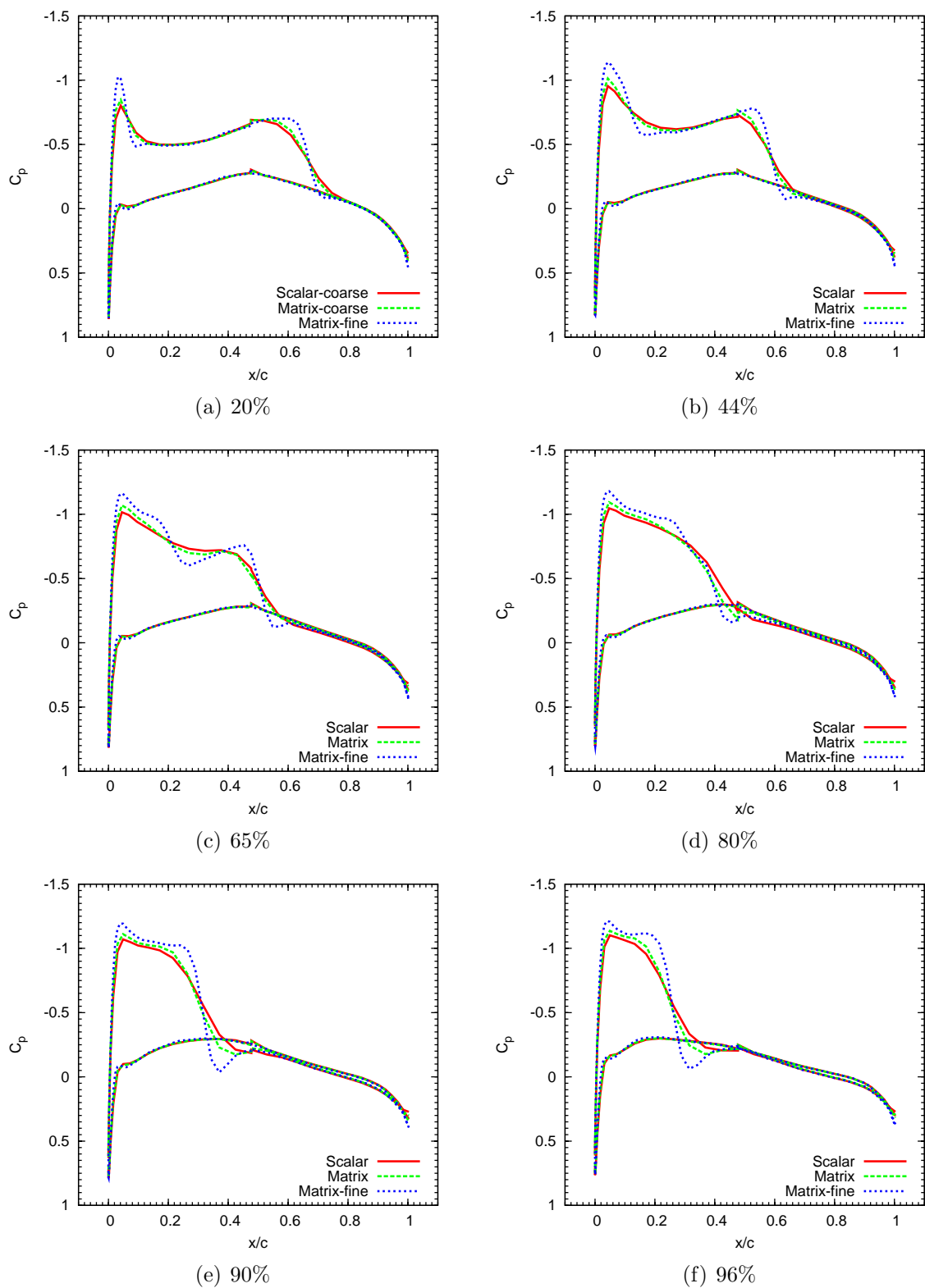


Figure 4.9: Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 2nd order method

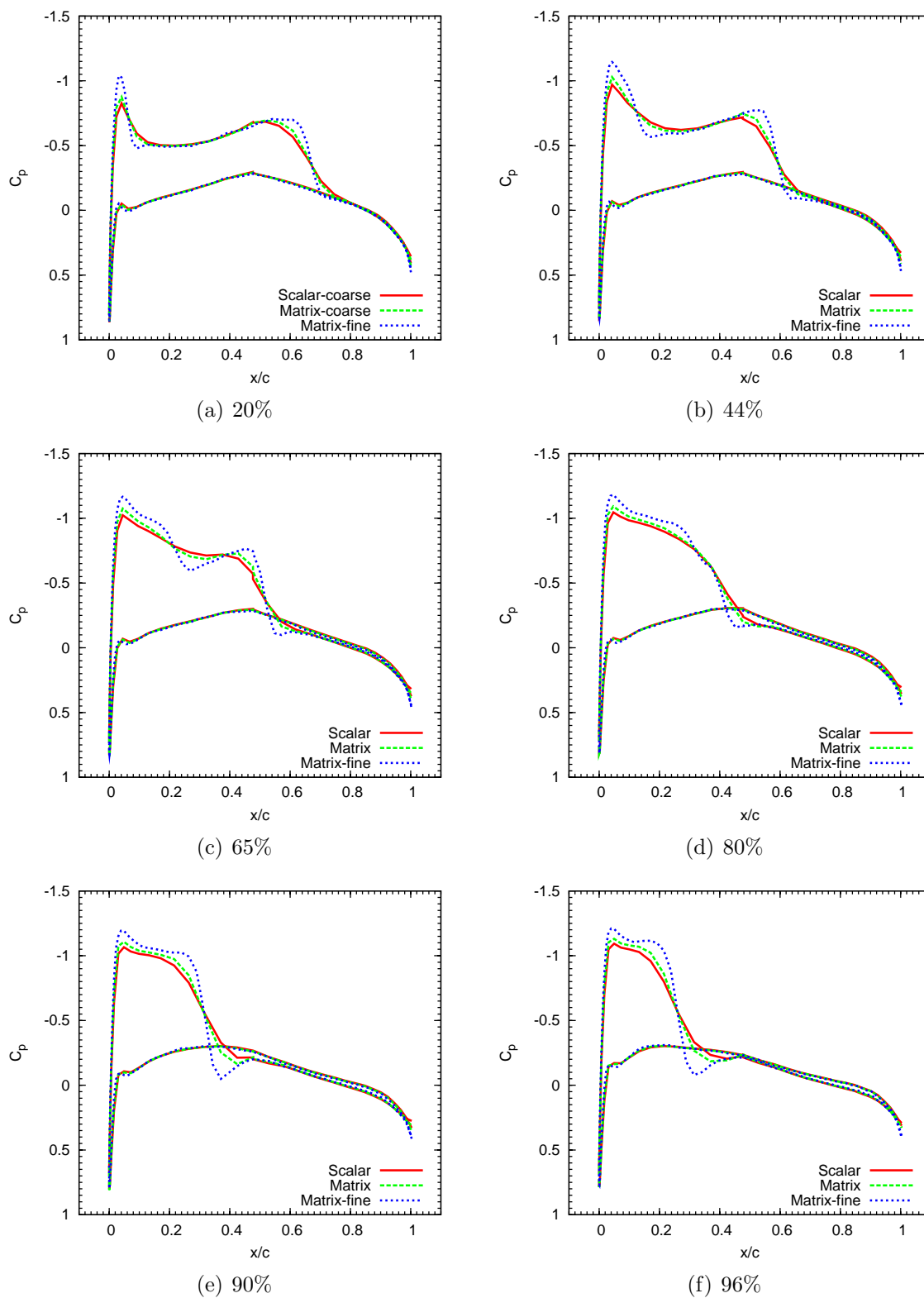


Figure 4.10: Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 3rd order method

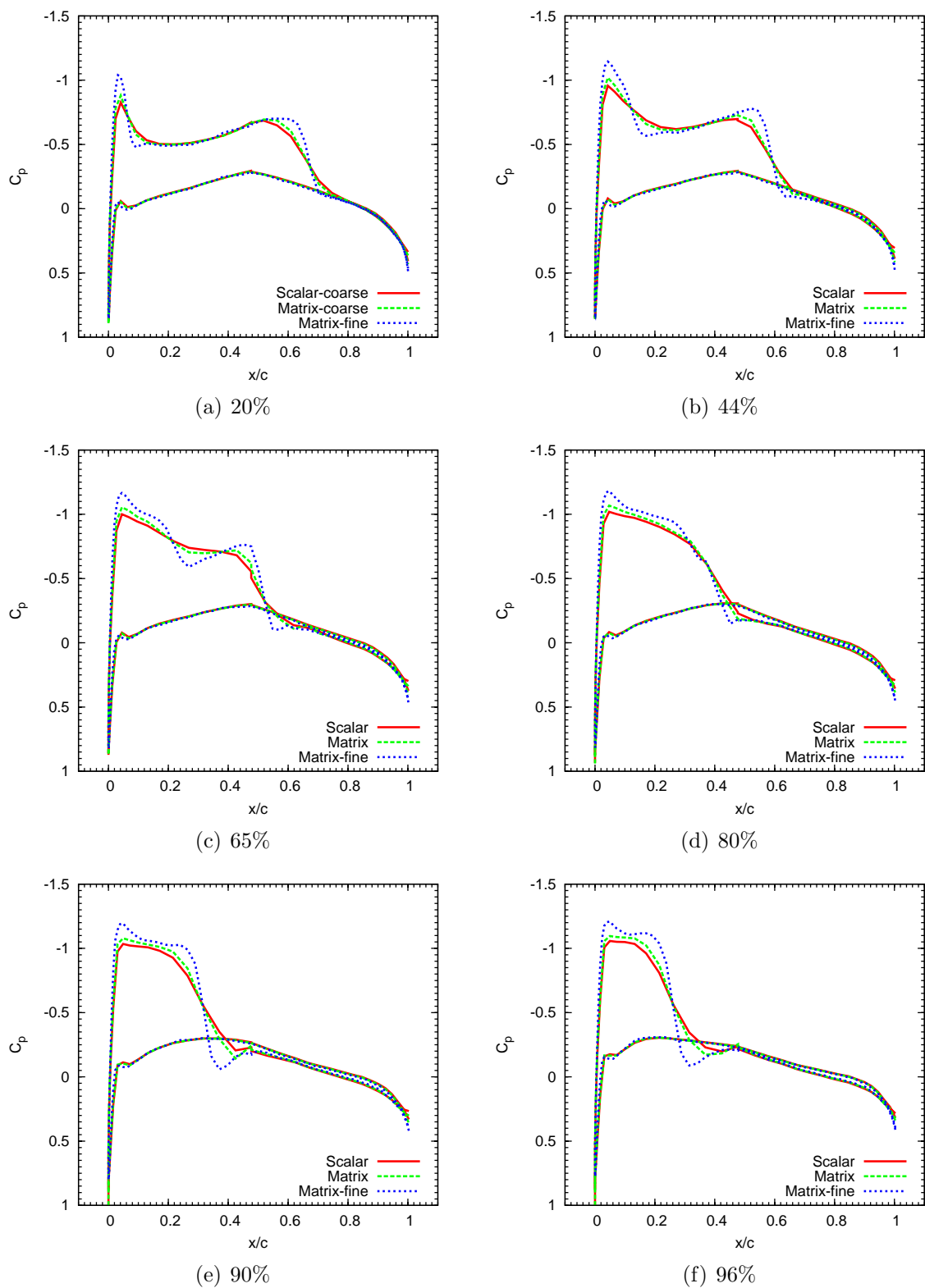


Figure 4.11: Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$, 4th order method

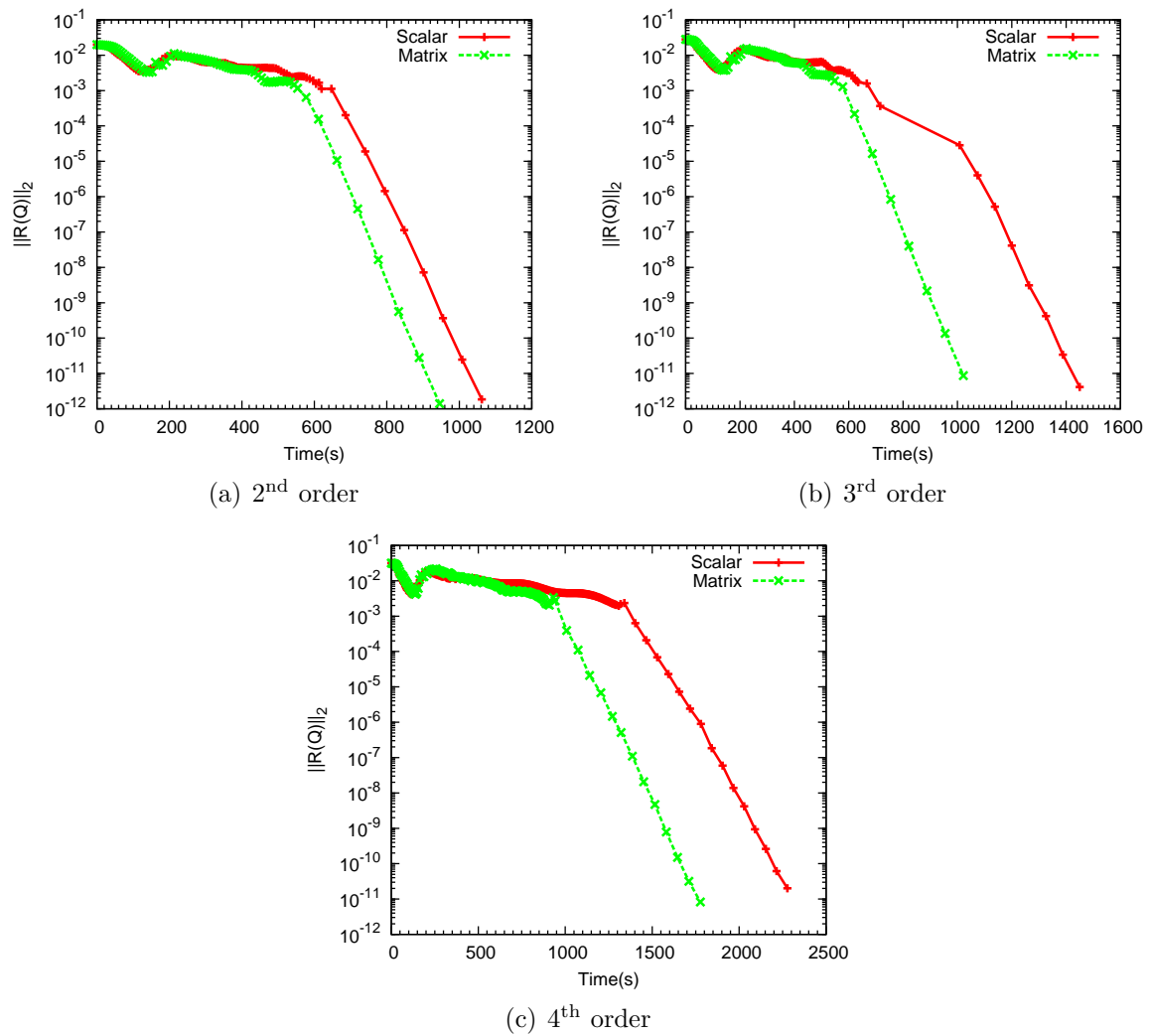
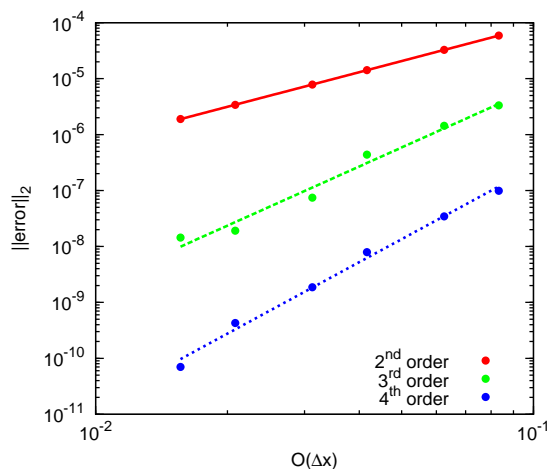


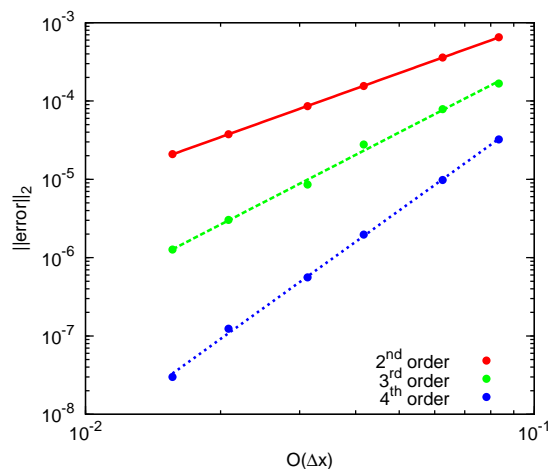
Figure 4.12: Convergence histories - Matrix and scalar dissipation - ONERA M6 wing $M=0.84$, $\alpha=3.06^\circ$ with preconditioner lumping factor $\sigma = 7$

Method	Slope
Quasi-one-dimensional nozzle	
2 nd order	2.054 ± 0.001
3 rd order	3.5 ± 0.3
4 th order	4.2 ± 0.2
MMS	
2 nd order	2.053 ± 0.004
3 rd order	3.0 ± 0.1
4 th order	4.1 ± 0.1
Ringleb flow	
2 nd order	1.874 ± 0.002
3 rd order	2.9 ± 0.1
4 th order	3.81 ± 0.03
Supersonic vortex	
2 nd order	1.907 ± 0.004
3 rd order	2.60 ± 0.05
4 th order	4.0 ± 0.1
Subsonic vortex	
2 nd order	1.937 ± 0.005
3 rd order	2.8 ± 0.1
4 th order	3.72 ± 0.01

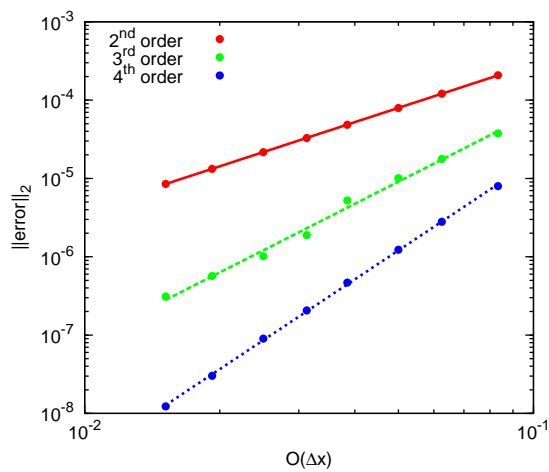
Table 4.4: Order of accuracy for different test cases, slopes of lines of best fit



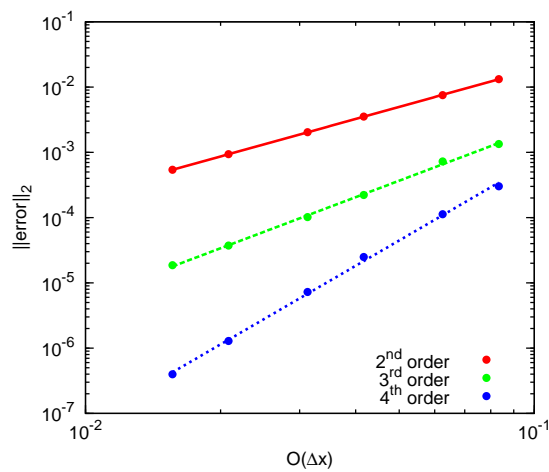
(a) Quasi-one-dimensional nozzle



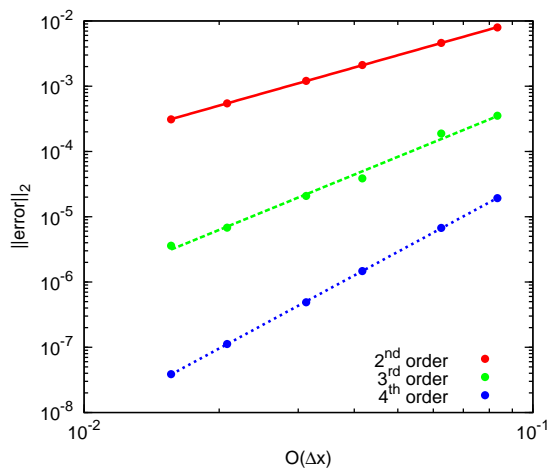
(b) MMS



(c) Ringleb flow



(d) Supersonic vortex



(e) Subsonic vortex

Figure 4.13: Order of accuracy for different test cases

where:

$$C_p = \frac{2}{M_\infty^2} \left(p - \frac{1}{\gamma} \right)$$

$$d\vec{S} = \left(\left(1 - \frac{1}{2}z \right) \cos \theta \hat{x} + \left(1 - \frac{1}{2}z \right) \sin \theta \hat{y} + \frac{1}{2} \left(1 - \frac{1}{2}z \right) \hat{z} \right) d\theta dz$$

$$p = \frac{1}{\gamma} z \cos \theta \sin \theta$$

The free stream Mach number M_∞ is arbitrarily set to 1. Equation 4.16 can now be integrated exactly to give,

$$f = \oint C_p \hat{x} \cdot d\vec{S} = -\frac{23}{18\gamma} \quad (4.17)$$

Now that we have an exact solution we can perform an order of accuracy study. The grid and the manufactured solution imposed on the surface is shown in Figure 4.14. Our integration method can be divided into the metrics and the trapezoidal and Simpson's rules. The results for the trapezoidal rule, Simpson's rule and the different metrics is shown in Figure 4.15 and computed slopes are shown in Table 4.5. The data establishes that we require high order metrics and Simpson's rule to attain a high-order integration method.

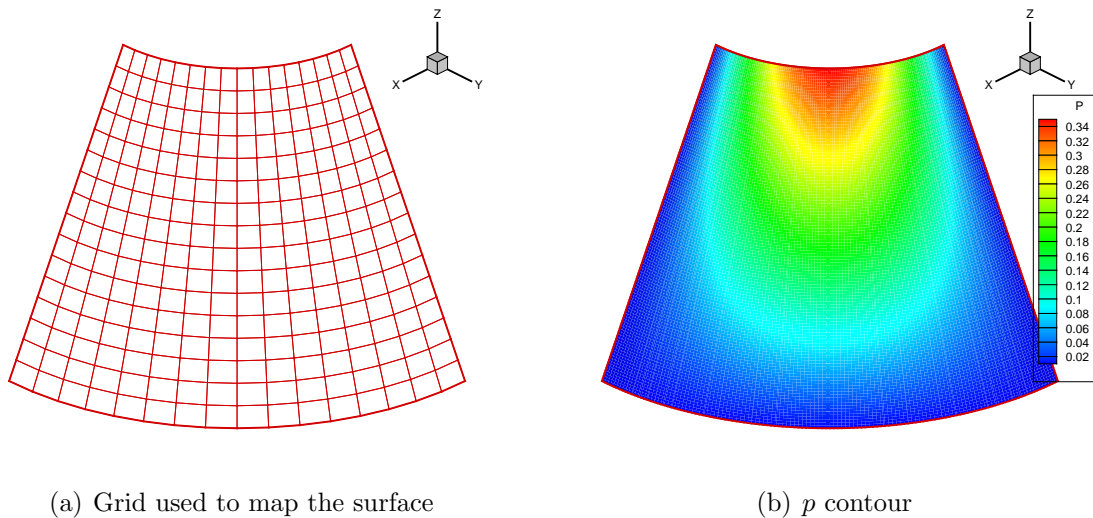


Figure 4.14: Surface used to test the order of accuracy of the integration method

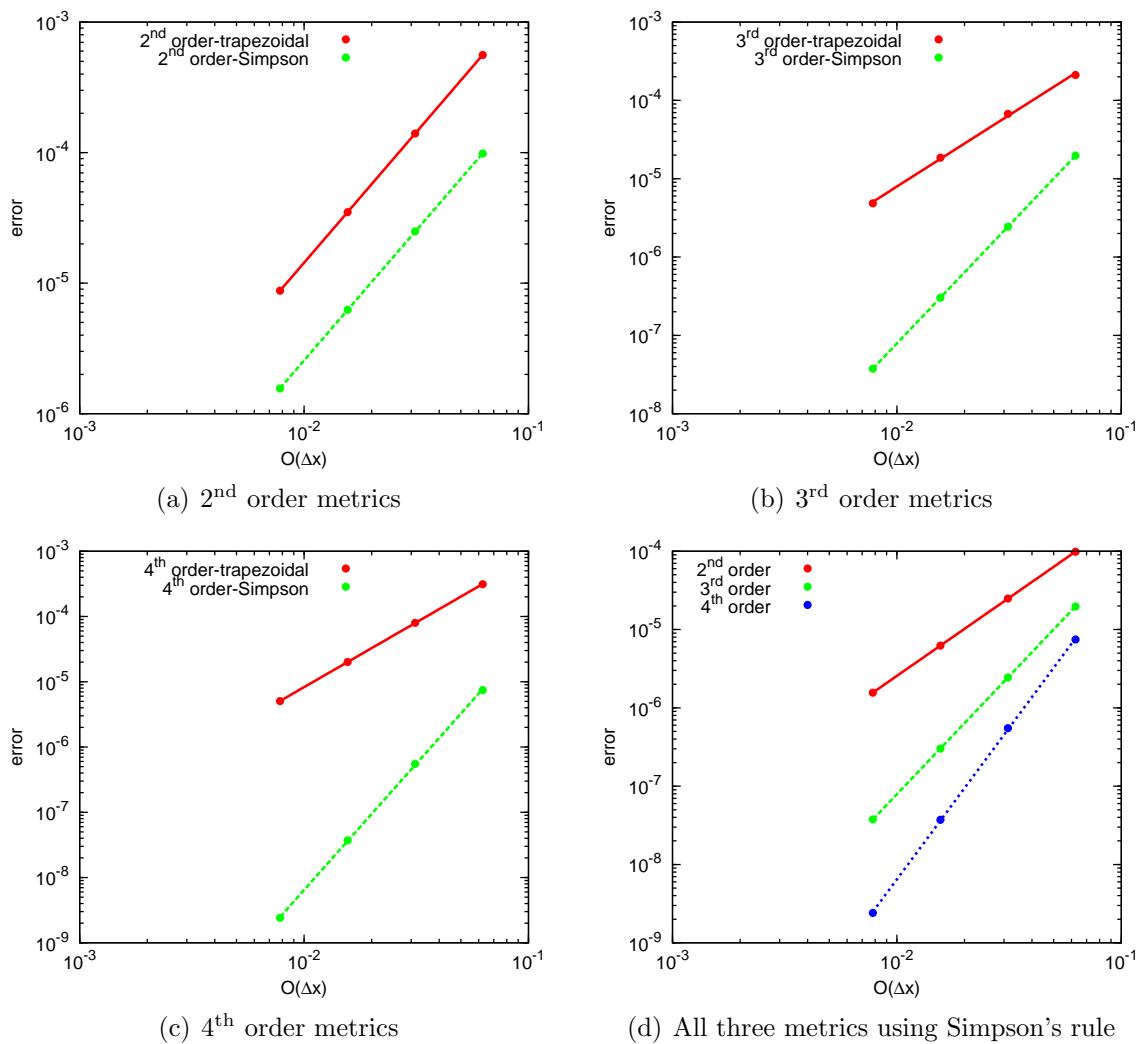


Figure 4.15: Order of accuracy for functional, f

Metrics	Slope	
	Trapezoidal rule	Simpson's rule
2 nd order	$2 \pm 1e - 06$	1.993 ± 0.003
3 rd order	1.82 ± 0.05	3.012 ± 0.001
4 th order	1.985 ± 0.004	3.87 ± 0.03

Table 4.5: Slopes of the lines of best fit for the functional, f

4.5.2 Integration method and flow solver

In this section we use the subsonic vortex shown in Figure 4.5. We define two functionals $C_L S$ and $C_D S$:

$$C_D S = \oint C_p \hat{x} \cdot d\vec{S} \quad (4.18)$$

$$C_L S = \oint C_p \hat{y} \cdot d\vec{S} \quad (4.19)$$

where

$$C_p = \frac{2}{M_\infty^2} \left(p - \frac{1}{\gamma} \right)$$

$$d\vec{S} = r (\sin \theta \hat{x} + \cos \theta \hat{y}) d\theta dz$$

The free stream Mach number M_∞ is arbitrarily set to 1. We integrate along the inner wall of the subsonic vortex and therefore $r = 2$. The pressure p can be calculated using equation (4.13) and the isentropic relationships. The variable z runs from 0 to $\frac{6}{5}$, and θ runs from 0 to $\frac{\pi}{4}$. One can now simplify and integrate to find an exact solution:

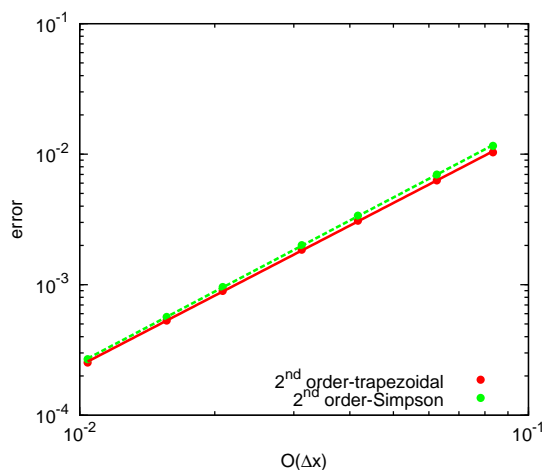
$$C_L S = \int_0^{\frac{6}{5}} \int_0^{\frac{\pi}{4}} \frac{(\gamma p - 1)}{\gamma} 2r \cos(\theta) d\theta dz = \frac{r6\sqrt{2}}{5} \frac{(\gamma p - 1)}{\gamma} \quad (4.20)$$

$$C_D S = \int_0^{\frac{6}{5}} \int_0^{\frac{\pi}{4}} \frac{(\gamma p - 1)}{\gamma} 2r \sin(\theta) d\theta dz = \frac{r6(2 - \sqrt{2})}{5} \frac{(\gamma p - 1)}{\gamma} \quad (4.21)$$

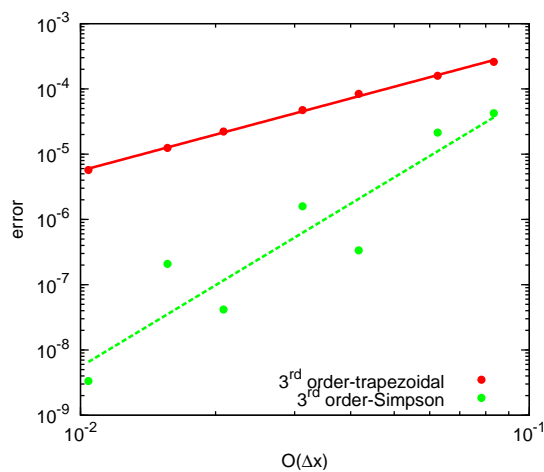
With an exact solution, we can calculate an error and perform an order of accuracy study. A similar study was performed by Nemeč et al. [43]. The studies in this work were performed using both the trapezoidal and Simpson's rules. The data for $C_L S$ and $C_D S$ are shown in Figures 4.16 and 4.17 respectively. The slopes for the lines of best fit for the data are shown in Tables 4.6 and 4.7. The errors shown are errors in the slopes of the lines of best fit. The data reinforce the fact that one requires high-order methods in both the integration and the flow solution to attain high-order accuracy in a functional.

4.6 Efficiency

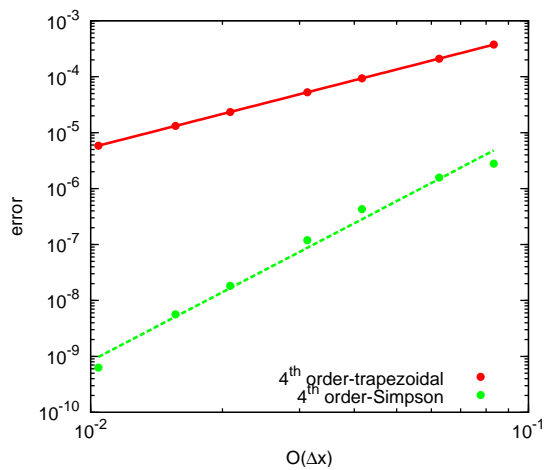
Efficiency may be studied quantitatively by analyzing the Ringleb flow, supersonic and subsonic vortex solutions shown in the previous sections. Normally two factors are taken



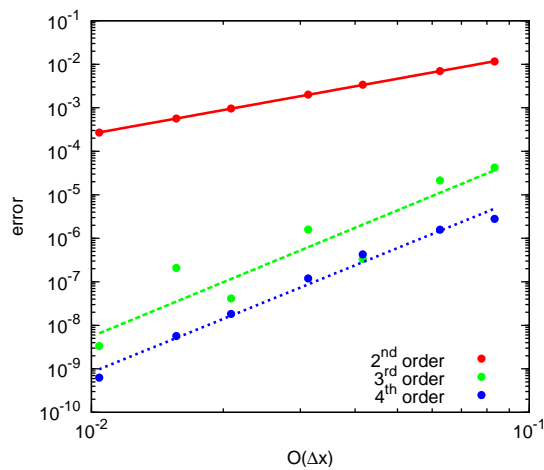
(a) 2nd order flow solve



(b) 3rd order flow solve

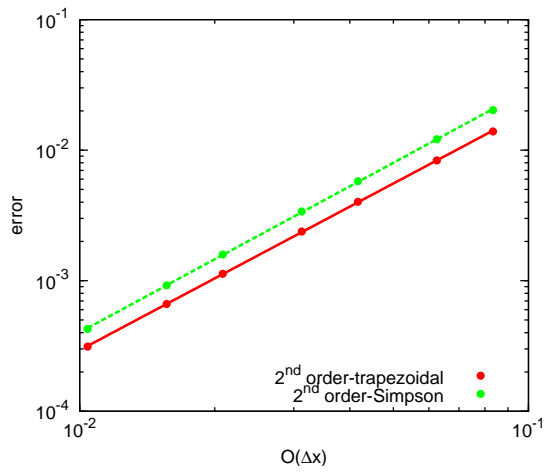


(c) 4th order flow solve

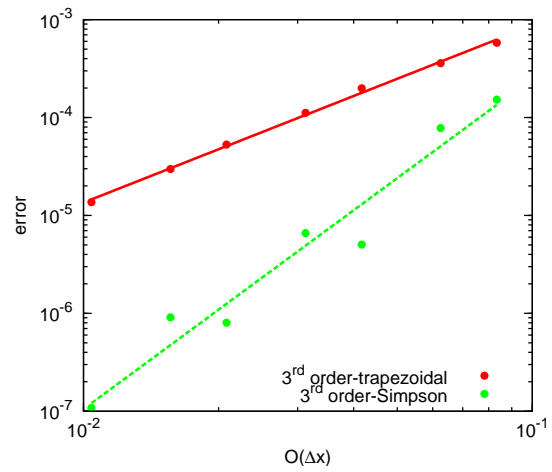


(d) All three flow solve methods using Simpson's rule

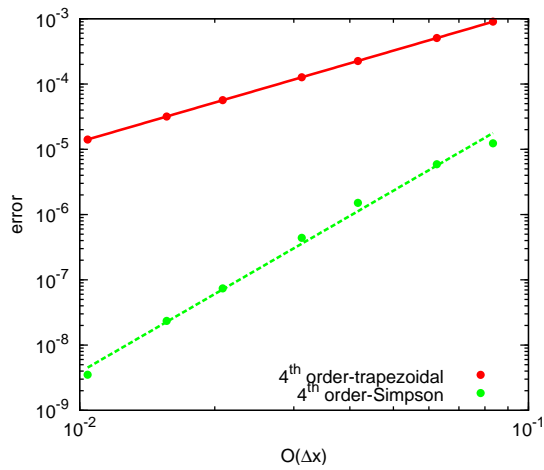
Figure 4.16: Order of accuracy for $C_L S$



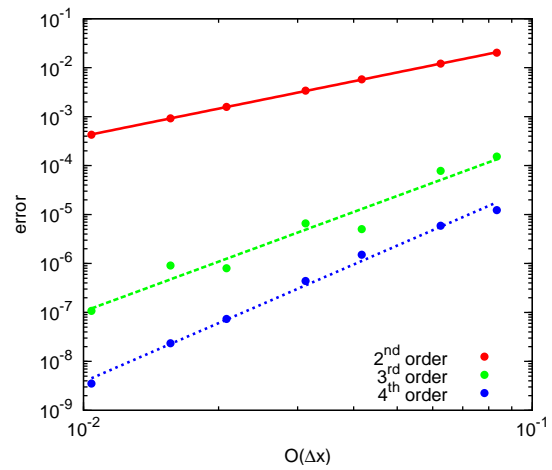
(a) 2nd order flow solve



(b) 3rd order flow solve



(c) 4th order flow solve



(d) All three flow solve methods using Simpson's rule

Figure 4.17: Order of accuracy for $C_D S$

Flow solve	Slope	
	Trapezoidal rule	Simpson's rule
2 nd order	1.784 ± 0.008	1.811 ± 0.006
3 rd order	1.84 ± 0.03	4.1 ± 0.8
4 th order	1.999 ± 0.001	4.1 ± 0.2

Table 4.6: Slopes of the lines of best fit for lift integration ($C_L S$) using both the integration method and the flow solver

Flow solve	Slope	
	Trapezoidal rule	Simpson's rule
2 nd order	1.83 ± 0.01	1.858 ± 0.008
3 rd order	1.81 ± 0.04	3.4 ± 0.3
4 th order	1.998 ± 0.001	4.0 ± 0.1

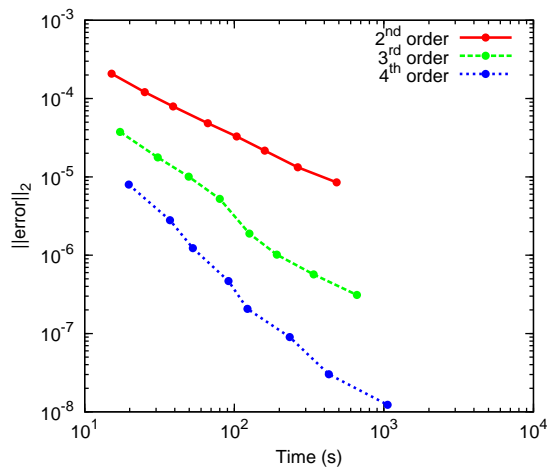
Table 4.7: Slopes of the lines of best fit for drag integration ($C_D S$) using both the integration method and the flow solver

into consideration when one performs a flow solve, the time to achieve convergence and the accuracy of the solution. Ideally one would want the most accurate solution in the shortest possible time. One may then specify the desired accuracy, or the time, if the time is a limiting factor. If the accuracy is specified, the most efficient method would provide a solution of this accuracy in the shortest possible time. If the time is a limiting factor, the most efficient method would provide the most accurate solution in this given time.

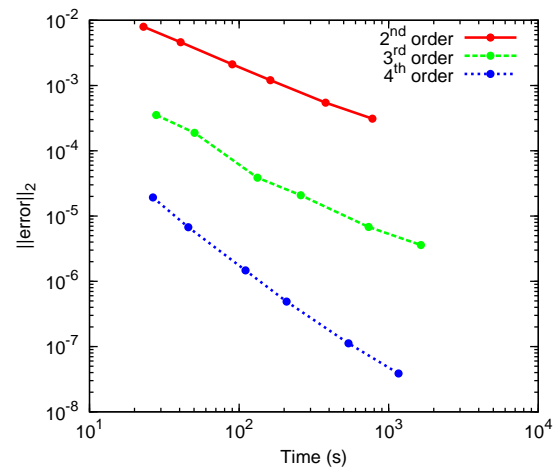
In Figure 4.18, the variation of the convergence time with accuracy of the solution is shown. The errors are based on the L_2 norm of the error in density. It is quite easy to see that with the above definition of efficiency the high-order methods are indeed efficient compared to the second-order method. For example, for the Ringleb flow, to achieve an error below 10^{-5} the third-order method is more than three times faster than the second-order method, and the fourth-order method is roughly eight times faster than the second-order method.

4.7 Mesh smoothness

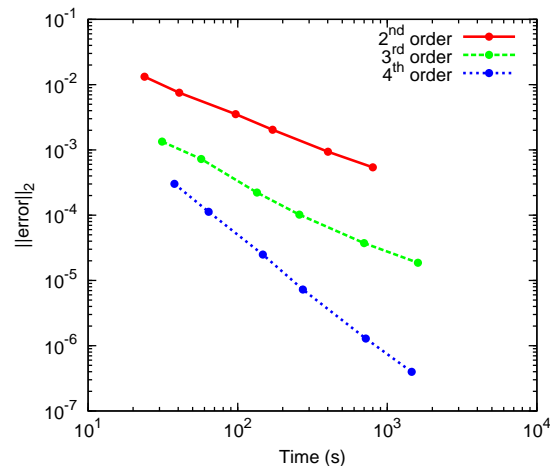
This sections attempts to demonstrate the need for smooth meshes to attain high-order accuracy. We used the method of manufactured solutions described previously to establish high-order accuracy. In this section, we perform a similar study; however, we now do it on a single block with three grids ranging from $17 \times 17 \times 17$ to $65 \times 65 \times 65$ nodes and we use two sets of meshes, one with a constant mesh spacing and another with a jump in mesh spacing



(a) Ringleb flow



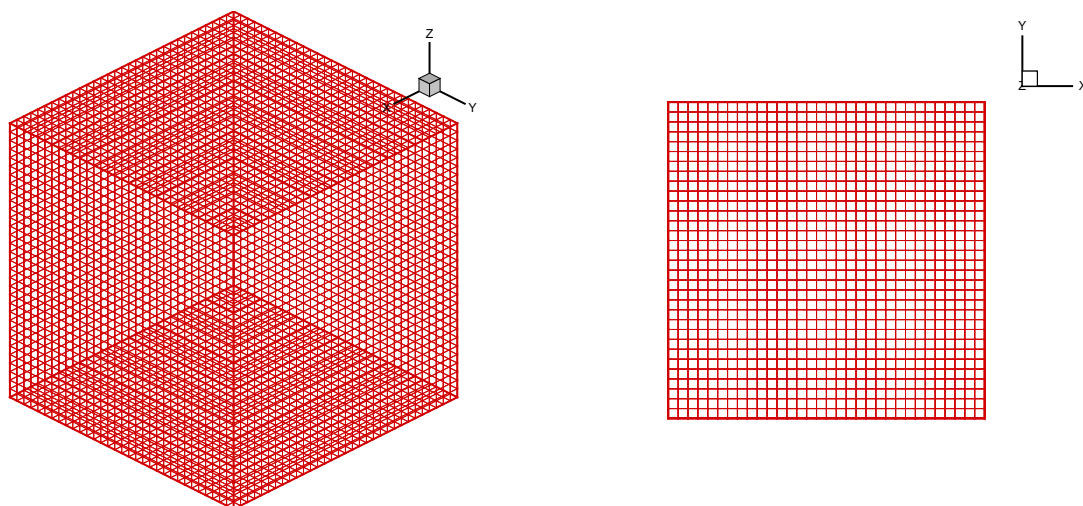
(b) Subsonic vortex



(c) Supersonic vortex

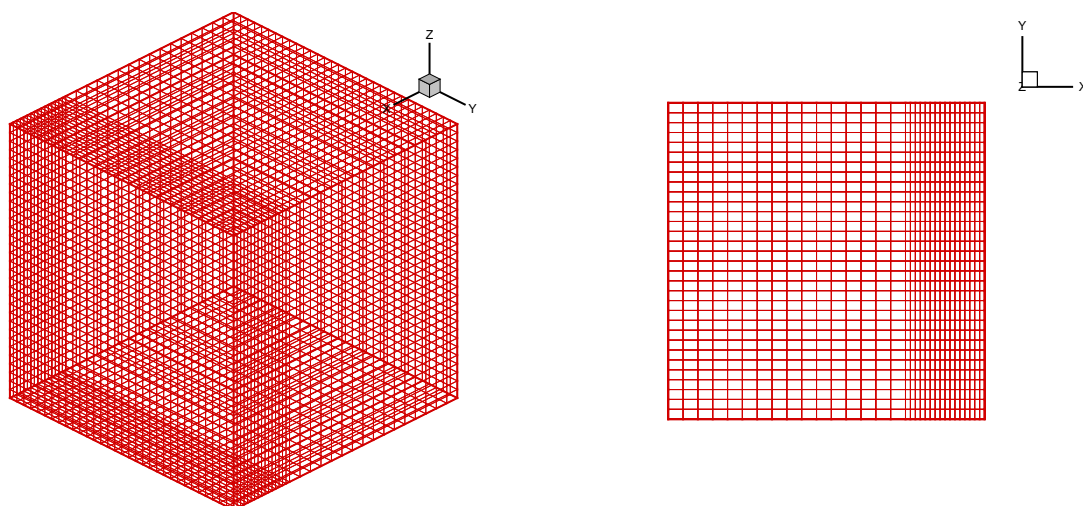
Figure 4.18: Efficiency study

in the x direction at $x = 0.75$. The grids are shown in Figure 4.19. Using the method of manufactured solutions with the second- and fourth-order methods we perform an order of convergence study, and the results for the constant and non-constant mesh spacing grids are shown in Figure 4.20. We see for that the order of accuracy for the non-constant spacing meshes is less than what we have for the constant spacing meshes. The slopes of the lines of best fit are 1.54 ± 0.04 and 2.06 ± 0.01 for the second-order method, and 1.9 ± 0.2 and 4.61 ± 0.04 for the fourth-order method respectively. Although, this brief study on mesh smoothness is not exhaustive it highlights the importance of smooth meshes.



(a) Three-dimensional view of constant spacing mesh

(b) $x - y$ plane of constant spacing mesh



(c) Three-dimensional view of non-constant spacing mesh

(d) $x - y$ plane of non-constant spacing mesh

Figure 4.19: Grids used to test mesh smoothness

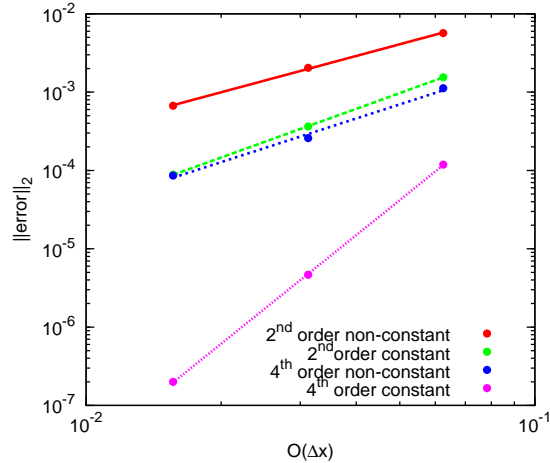


Figure 4.20: Order of accuracy depending on the grid smoothness

4.8 Parameter study

In this section, the effect of different parameters is studied. The aim is to determine optimal parameter values. The ONERA M6 wing at $M=0.699$ & $\alpha=3.06^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$ on a $96 \text{ blocks} \times 17 \times 17 \times 17$ grid is used in this study.

The dissipation coefficients are set to $\kappa_2 = 1$ and $\kappa = 0.04$. V_n and V_l are set to 0.25 for all flows. The transition region in the dissipation model is set to 8%. Far field and wall boundary conditions are used. The first-order preconditioner is used, and coefficients during the start-up phase are $a = 0.01$, $b = 1.2$ and $\omega = 0.05$. During the inexact-Newton phase β is chosen to be 2, and the parameter that decides when we switch to the inexact phase, τ , is set to $1/15$. The number of GMRES iterations is limited to 80 without restarts, and the solution is said to be converged when the residual norm has dropped below $1E - 11$.

We shall investigate optimal parameters for $ILU(k)$ factorization during the start-up and inexact-Newton phases, the dissipation lumping factor σ , relaxation, a lagged Jacobian update and the boundary condition weighting constant.

4.8.1 ILU fill parameter

As discussed earlier, ILU factorization of the preconditioner is required during the start-up and inexact-Newton phases. Here we investigate the optimal fill level k for both stages. For this we utilize the ONERA M6 wing at $M=0.699$ & $\alpha=3.06^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$ on a $96 \text{ blocks} \times 17 \times 17 \times 17$ grid. The dissipation lumping factor is set to $\sigma = 10$, relaxation is set to 0.5, the Jacobian is updated every iteration and the boundary condition weighting

constant is set to $\tau_{bc} = 1$. The data for the start-up and inexact-Newton phases¹ is shown in Figures 4.21 and 4.22, and Figures 4.23 and 4.24, respectively. From the data we see that ILU(0) for the start-up phase and ILU(1) for the inexact-Newton phase provide the best performance for all three methods.

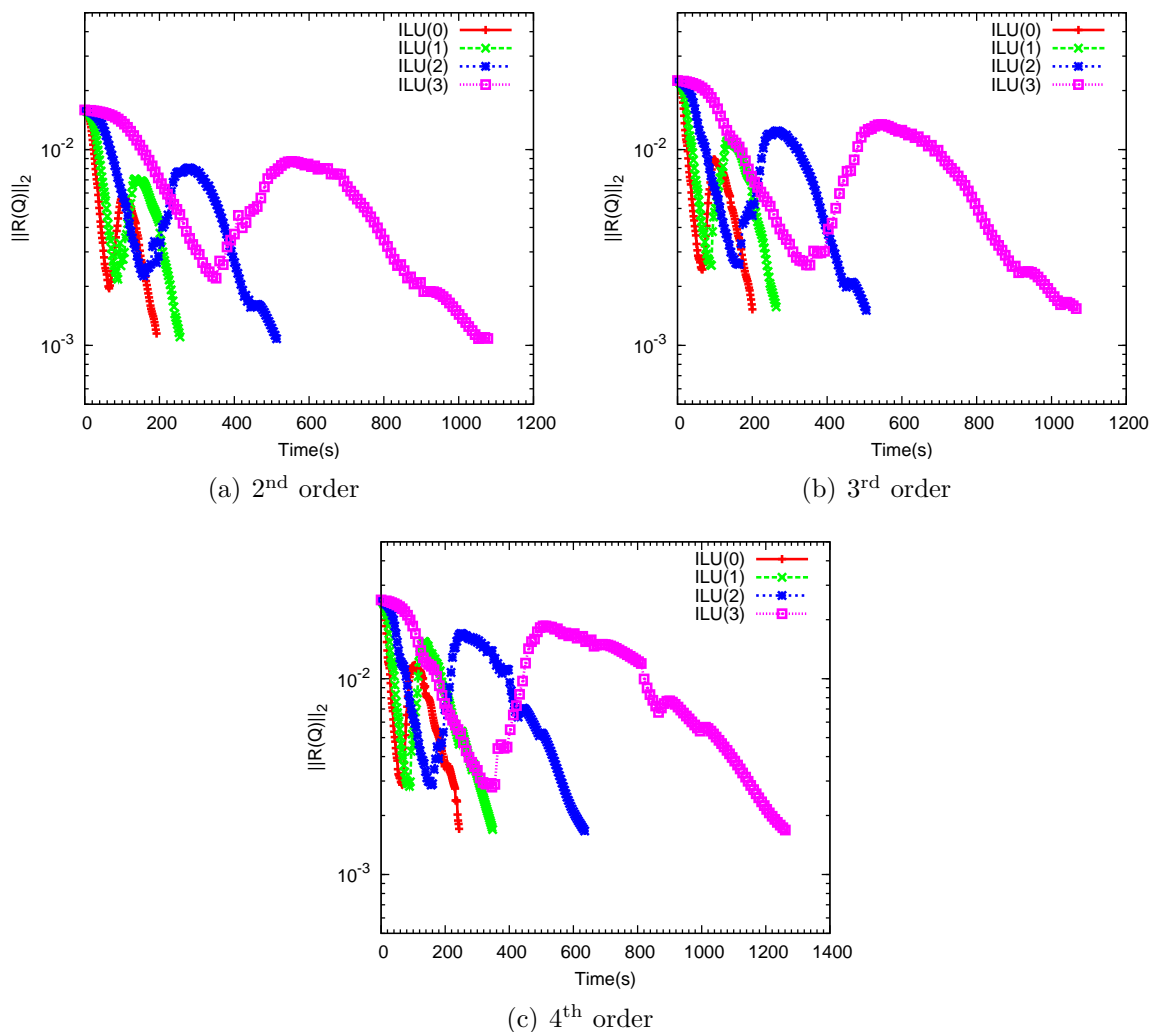


Figure 4.21: Effect of the fill level, k , for the ILU(k) factorization during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$

4.8.2 Dissipation lumping factor

In this section, the effect of the constant σ that is used to lump the first and high-order dissipation coefficients in the approximate Jacobian used to form the preconditioner is studied.

¹ILU(0) is used during the start-up phase for the study of ILU factorization during the inexact-Newton phase

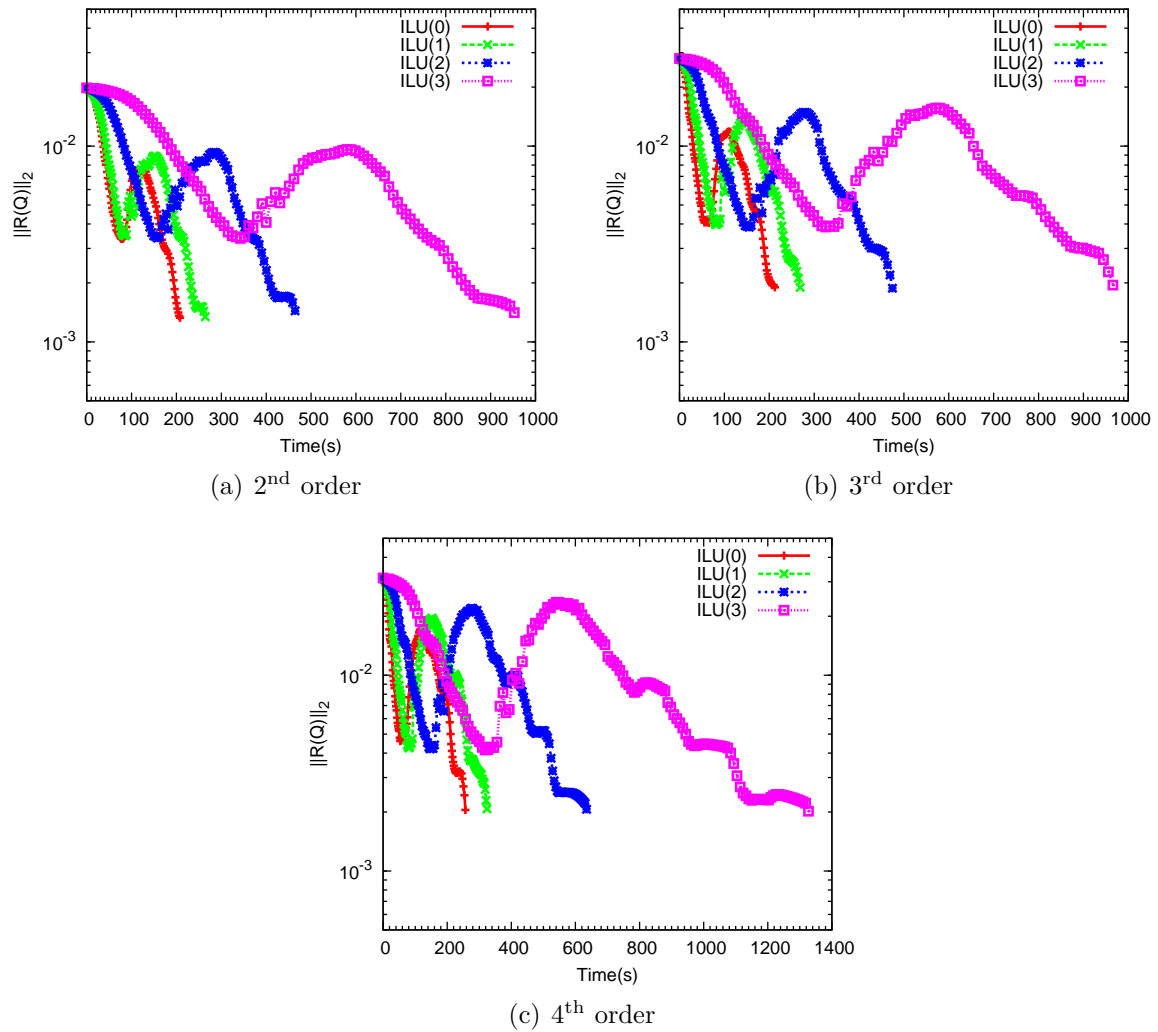


Figure 4.22: Effect of the fill level, k , for the $ILU(k)$ factorization during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$

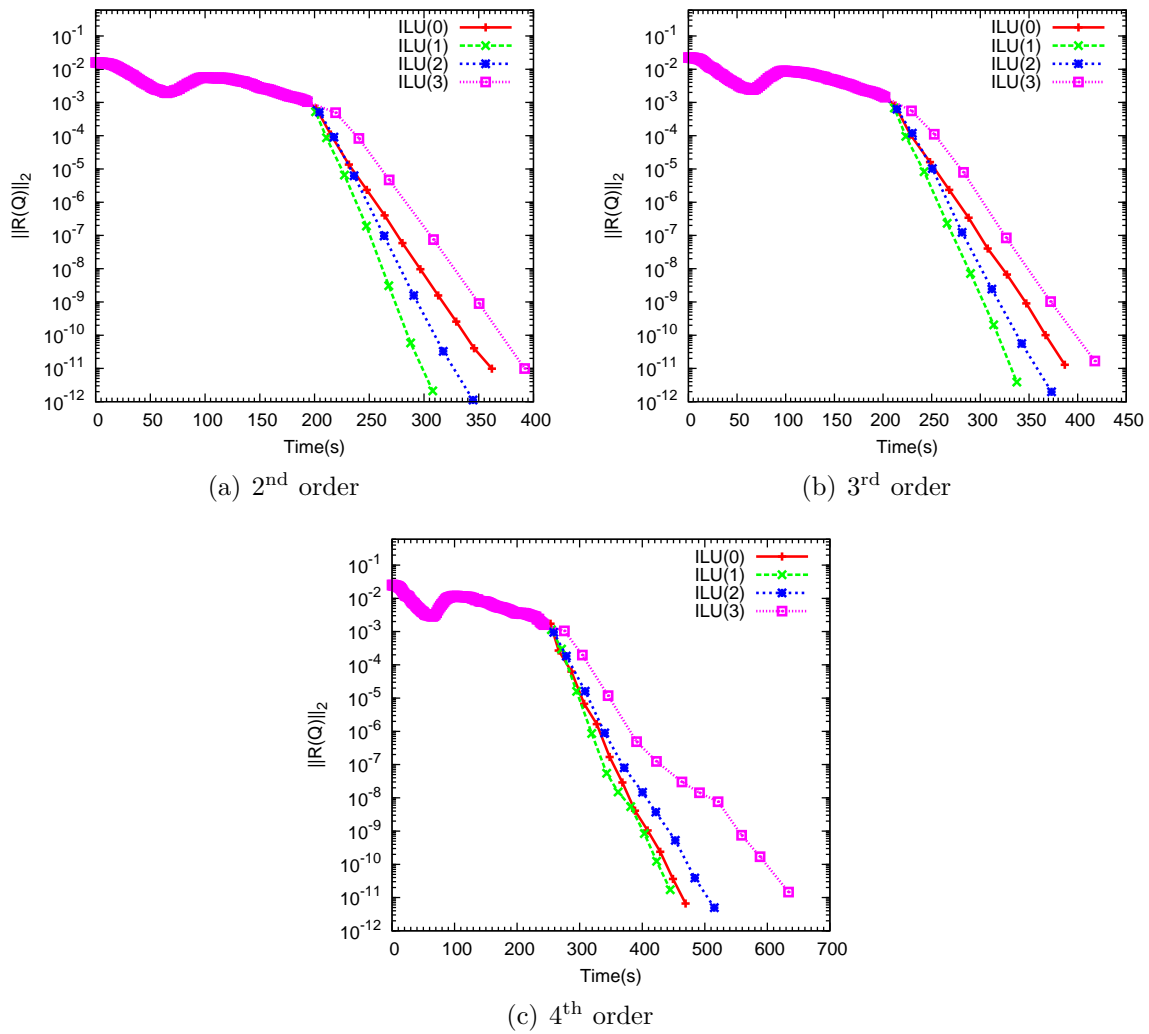


Figure 4.23: Effect of the fill level, k , for the $ILU(k)$ factorization during the inexact-Newton phase for $M=0.699$ & $\alpha = 3.06^\circ$

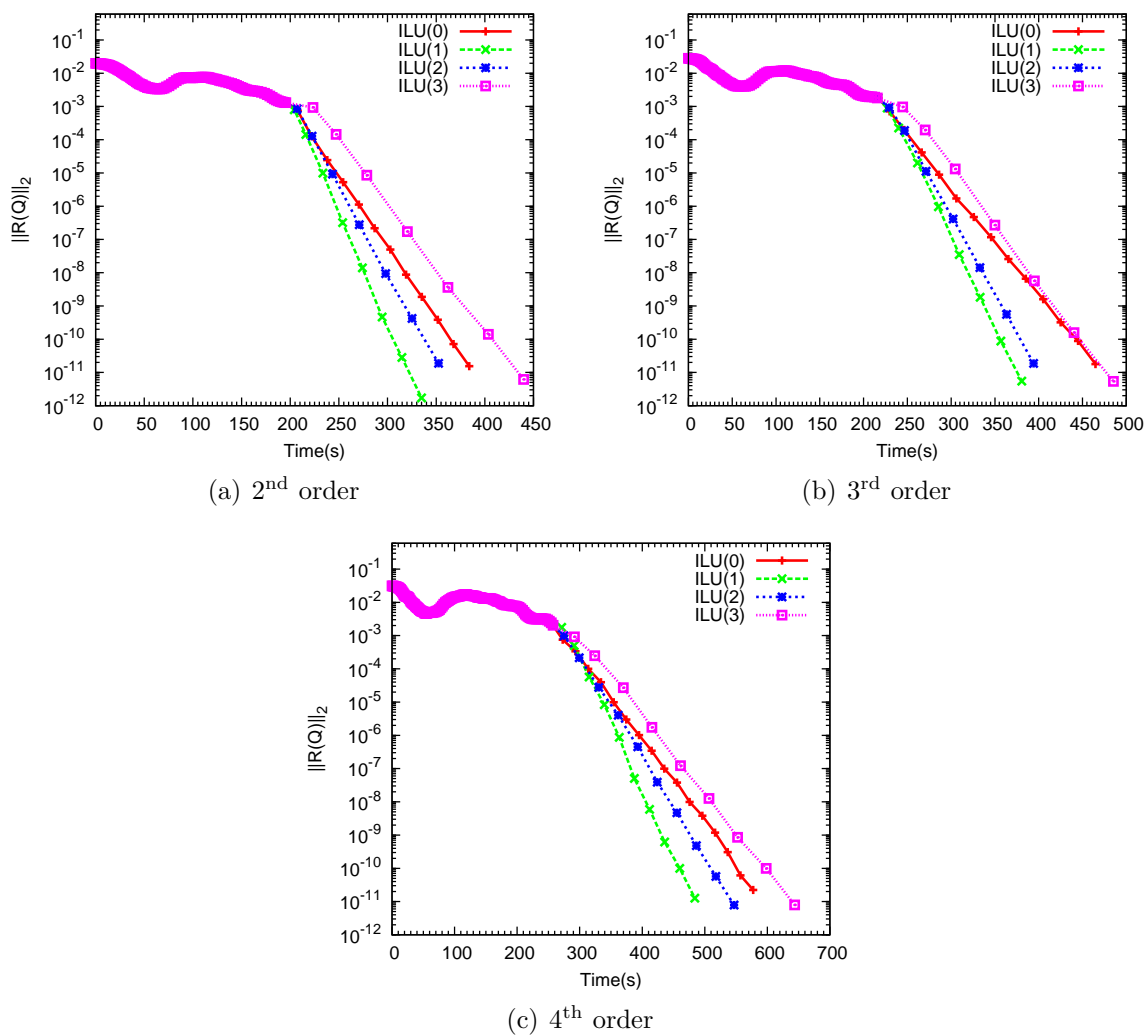


Figure 4.24: Effect of the fill level, k , for the $ILU(k)$ factorization during the inexact-Newton phase for $M=0.84$ & $\alpha = 3.06^\circ$

The data for the the ONERA M6 wing at $M=0.699$ & $\alpha=3.06^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$ on a $96 \text{ blocks} \times 17 \times 17 \times 17$ grid is shown in Figures 4.25 and 4.26. ILU(0) and ILU(1) is used during the start-up and inexact-Newton phases respectively, relaxation is set to 0.5, the Jacobian is updated every iteration and the boundary condition weighting constant is set to $\tau_{bc} = 1$. We see that 10 is optimal for all three methods. Values less than 2 may make the method unstable.

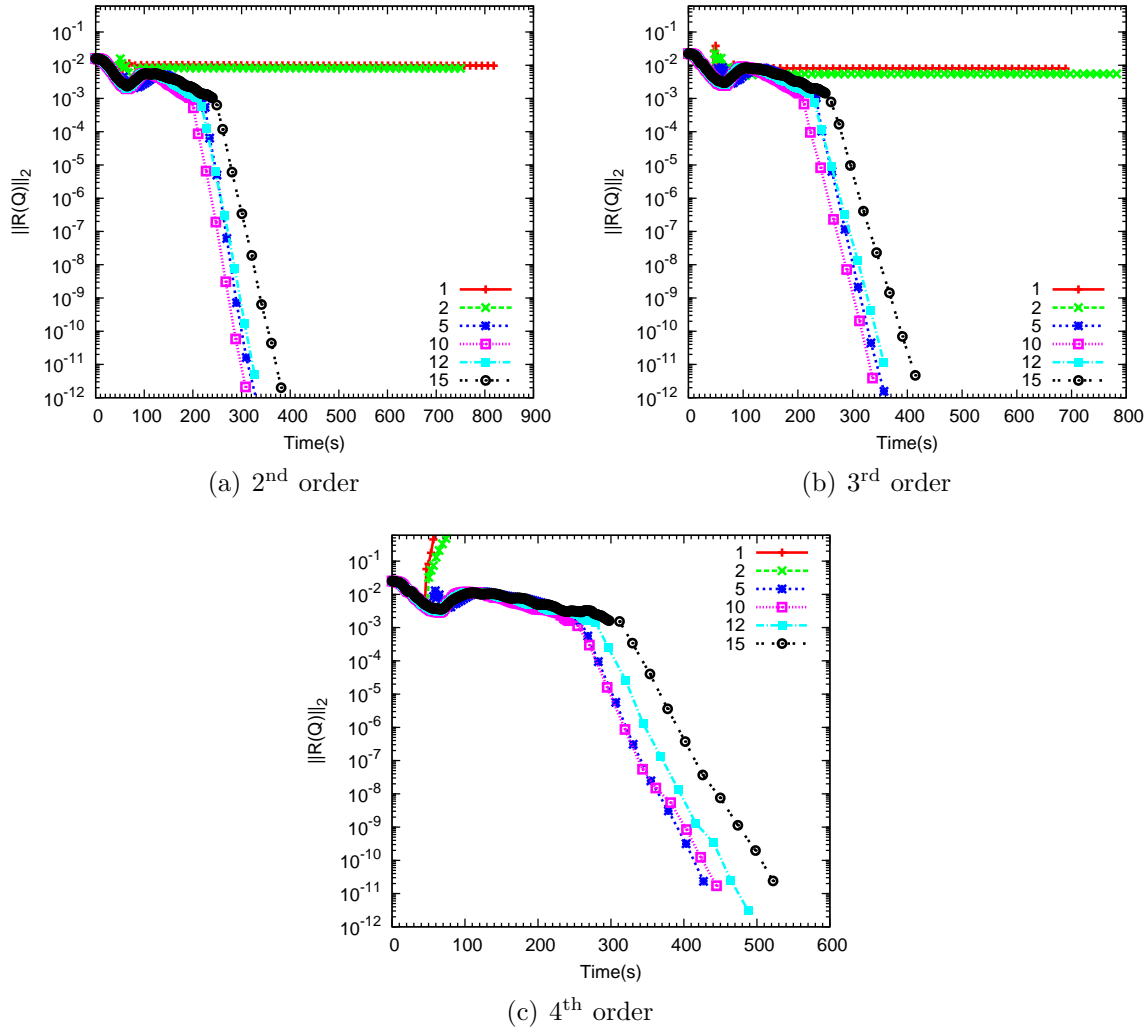


Figure 4.25: Effect of the constant σ used to lump the first and high-order dissipation coefficients for $M=0.699$ & $\alpha = 3.06^\circ$

4.8.3 Relaxation

As discussed earlier, relaxation may be used during the start-up phase. Here we investigate the effects of over- and under-relaxation. The data for the ONERA M6 wing at $M=0.699$ &

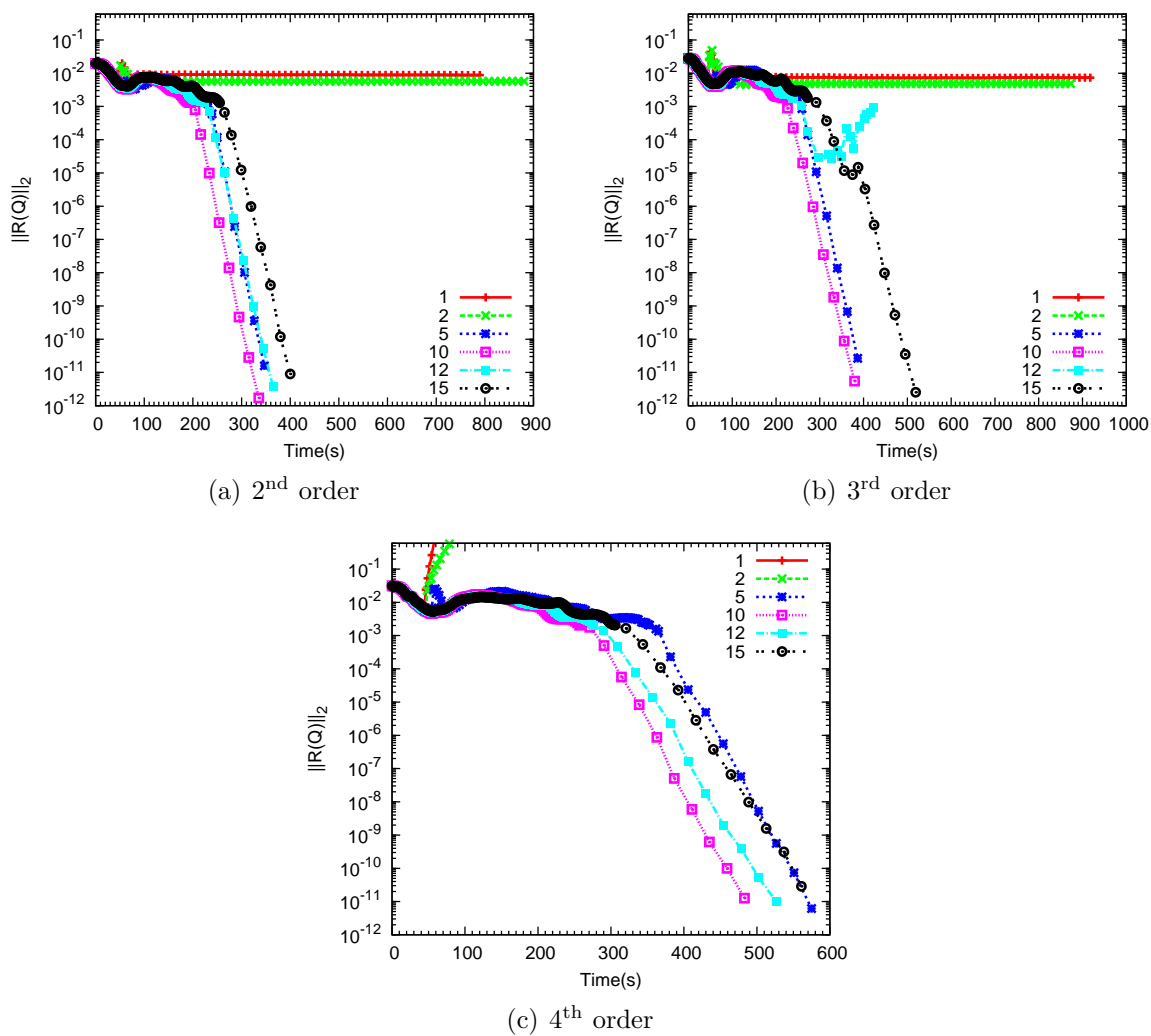


Figure 4.26: Effect of the constant σ used to lump the first and high-order dissipation coefficients for $M=0.84$ & $\alpha = 3.06^\circ$

$\alpha=3.06^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$ on a $96 \text{ blocks} \times 17 \times 17 \times 17$ grid is shown in Figures 4.27 and 4.28. ILU(0) and ILU(1) is used during the start-up and inexact-Newton phases respectively, the dissipation lumping factor is set to $\sigma = 10$, the Jacobian is updated every iteration and the boundary condition weighting constant is set to $\tau_{bc} = 1$. We see that for certain cases under-relaxation may be required to maintain stability. On the other hand, over-relaxation up to a point helps in improving convergence rates. From the data, under-relaxation of 0.6 provides a good balance between stability and speed.

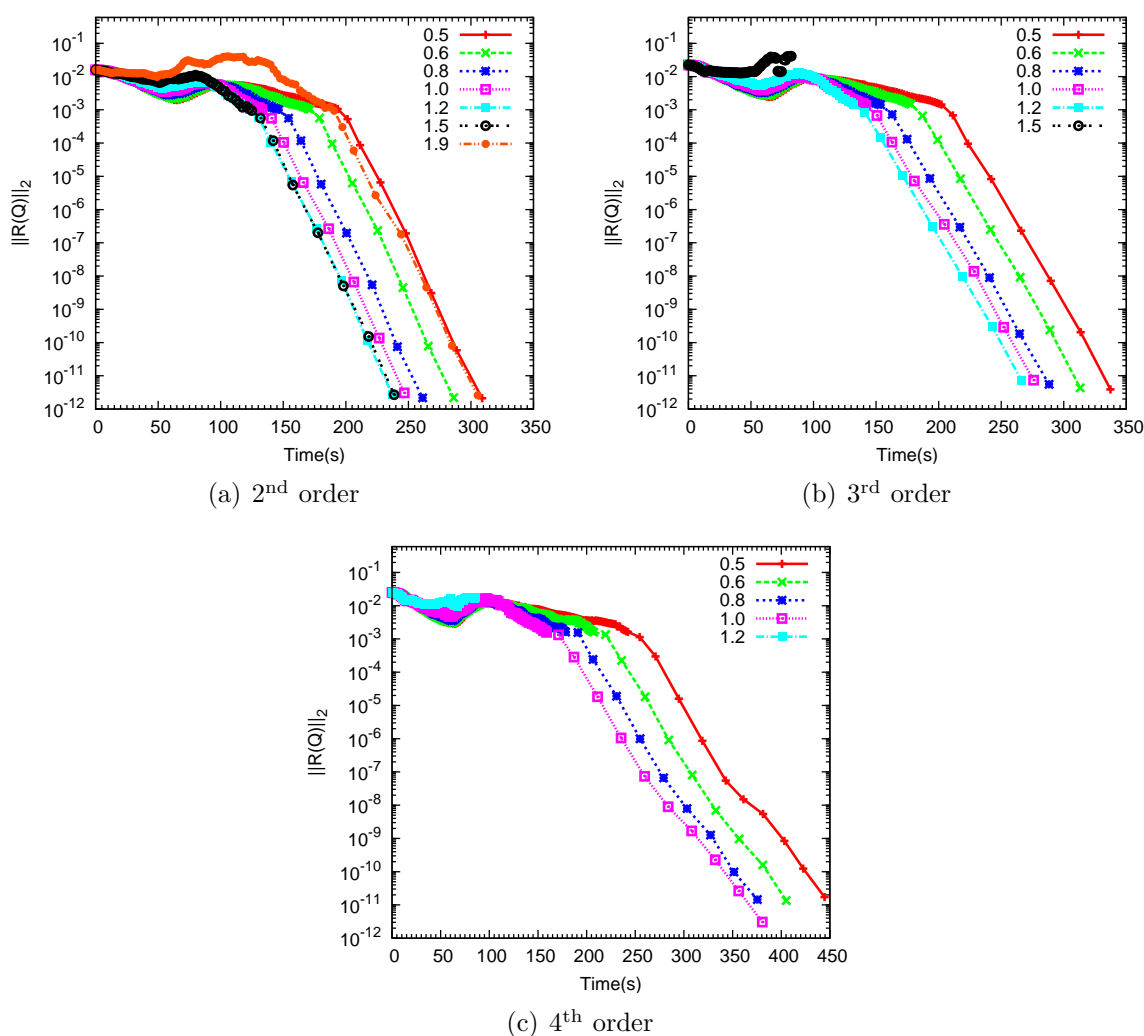


Figure 4.27: Effect of over- and under-relaxation during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$

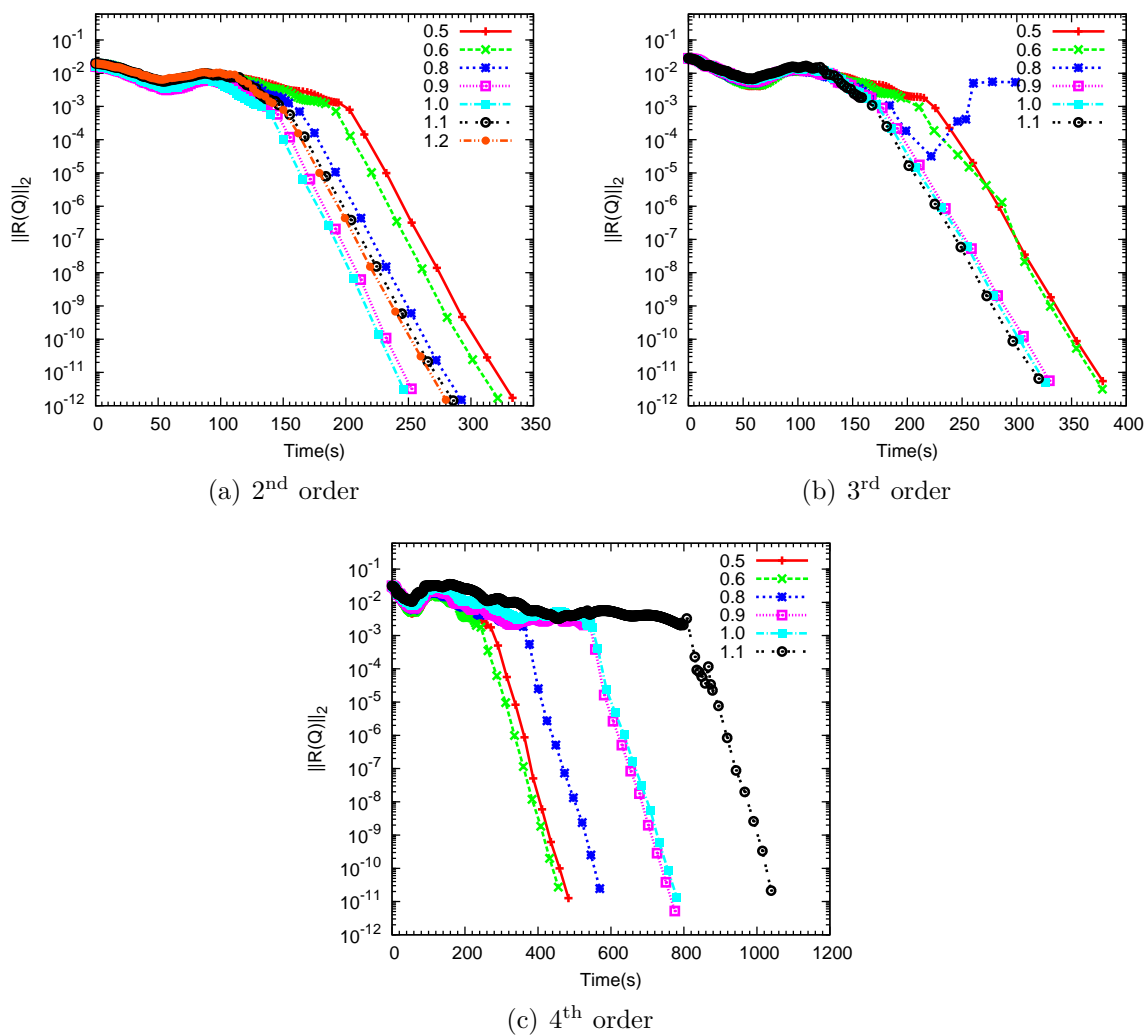


Figure 4.28: Effect of over- and under-relaxation during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$

4.8.4 Lagged Jacobian update

In this section, we study how the frequency with which we update the Jacobian during the start-up phase affects the convergence rate. The data for the the ONERA M6 wing at $M=0.699$ & $\alpha=3.06^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$ on a $96 \text{ blocks} \times 17 \times 17 \times 17$ grid is shown in Figures 4.29 and 4.30. ILU(0) and ILU(1) is used during the start-up and inexact-Newton phases respectively, the dissipation lumping factor is set to $\sigma = 10$, relaxation is set to 0.6 and the boundary condition weighting constant is set to $\tau_{bc} = 1$. From the data we see significant reductions in convergence time if we update the Jacobian every 3 outer iterations. There is only marginal further improvement if we update the Jacobian less frequently.

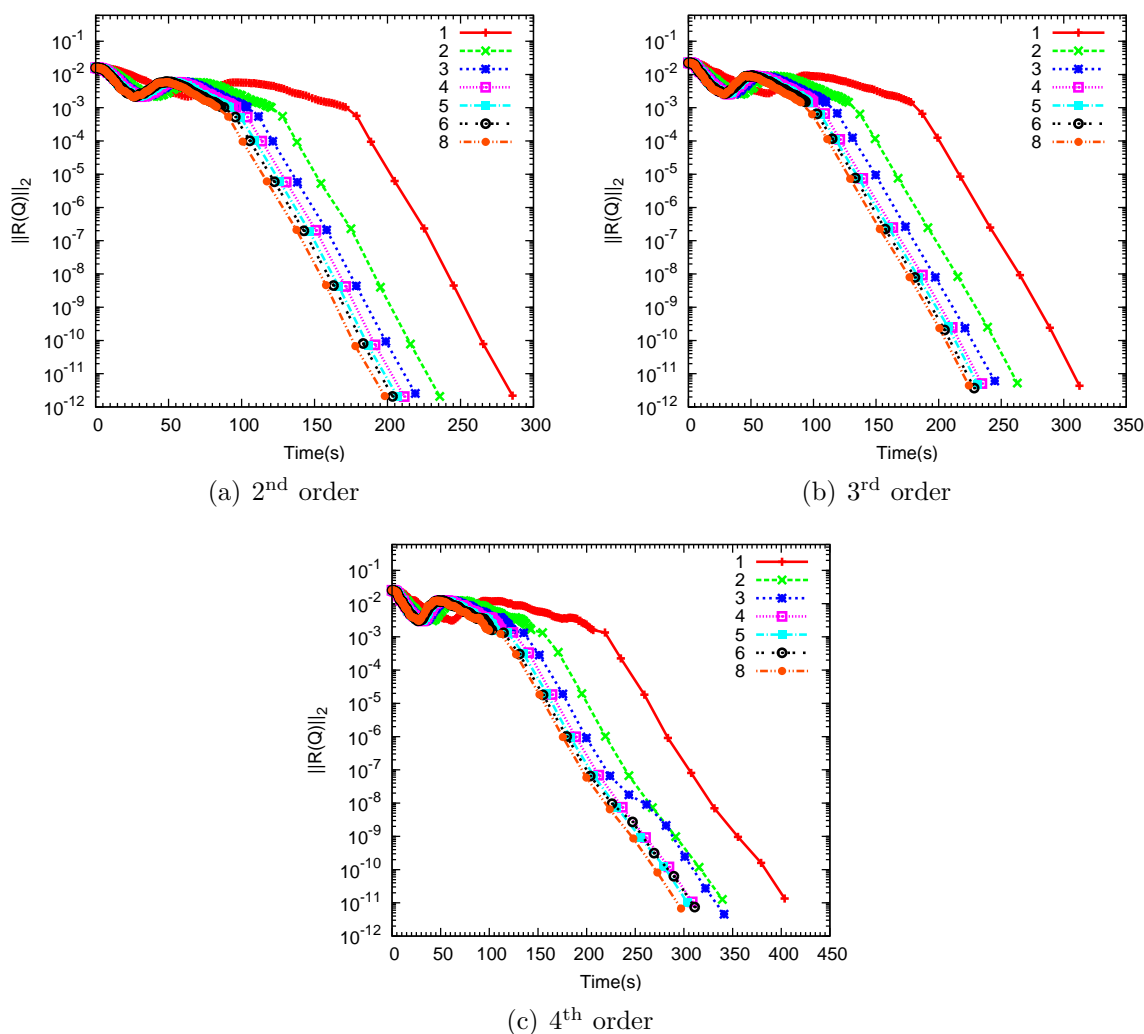


Figure 4.29: Effect of the update frequency of the Jacobian during the start-up phase for $M=0.699$ & $\alpha = 3.06^\circ$

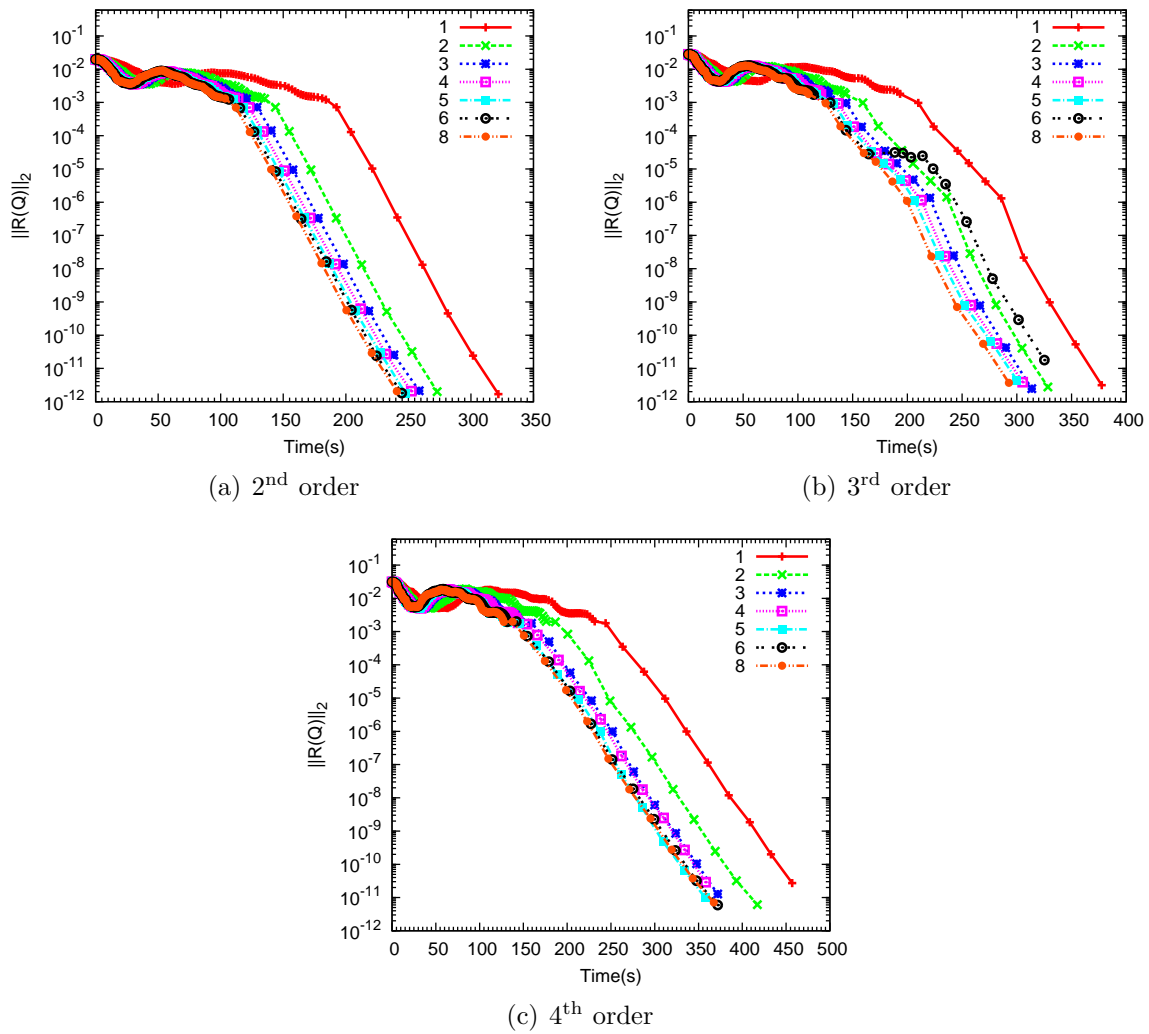


Figure 4.30: Effect of the update frequency of the Jacobian during the start-up phase for $M=0.84$ & $\alpha = 3.06^\circ$

τ	2 nd order	3 rd order	4 th order
0.75	0.0764%	9.77%	6.95%
1.0	-7.87%	10.48%	4.27%
1.25	-13.27%	11.01%	2.26%
1.5	-17.34%	11.42%	0.59%
1.75	-	-	-0.87%

Table 4.8: Effect of the boundary and interface weighting constant τ_{bc} on the drag coefficient

τ	2 nd order	3 rd order	4 th order
0.75	-0.0719%	0.171%	0.122%
1.0	0.0794%	0.148%	0.145%
1.25	0.176%	0.122%	0.179%
1.5	0.238%	0.096%	0.216%
1.75	-	-	0.254%

Table 4.9: Effect of the boundary and interface weighting constant τ_{bc} on the lift coefficient

4.8.5 Boundary condition and interface weighting constant

In this section, the effect of the boundary and interface weight constant, τ_{bc} , is studied. We use the ONERA M6 wing at $M=0.3$ & $\alpha=4.0^\circ$ on a 96 blocks $\times 17\times 17\times 17$. The data for drag and lift are shown in Tables 4.8 and 4.9 respectively. The data are shown as the percentage difference from those values obtained on a 96 blocks $\times 33\times 33\times 33$ grid using the fourth-order method with $\tau_{bc} = 1$. From the data we see that the lift and drag values tend to move in the opposite direction with variation in τ_{bc} . Also values of τ_{bc} less than 0.75 tend to render the method unstable.

4.9 ONERA M6 Wing

In this section, we use the ONERA M6 wing in three test cases, $M=0.699$ & $\alpha=3.06^\circ$, $M=0.3$ & $\alpha=4.0^\circ$ and $M=0.84$ & $\alpha=3.06^\circ$. The data for the last two cases may be found in Appendix A. The second test case is purely subsonic flow whereas the other two are transonic flows. The first test case has a weak leading edge shock, and the last test case has a strong shock on the upper surface of the wing.

We use five grids ranging from 96 blocks $\times 17\times 17\times 17$ nodes (471,648 nodes) to 96 blocks $\times 33\times 33\times 33$ nodes (3,449,952 nodes). The far field is placed 25 chord lengths away from the wing. The 96 blocks $\times 33\times 33\times 33$ nodes mesh used is shown in Figure 4.31.

The dissipation coefficients for the subsonic flow are $\kappa_2 = 0$ and $\kappa = 0.04$. This is because

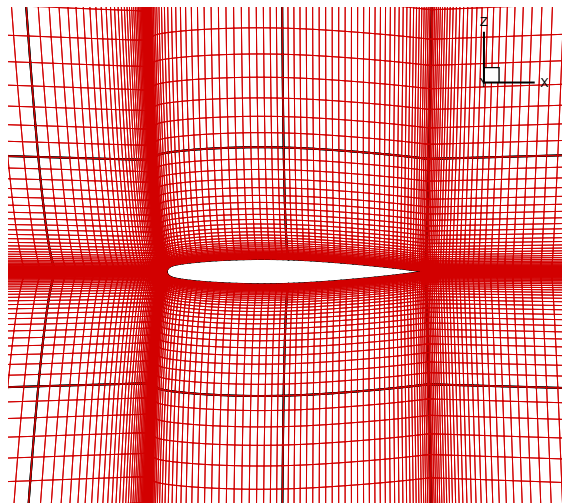


Figure 4.31: Mesh for the ONERA M6 wing

we have no shocks, and thus no first-order smoothing is required. For the transonic flows, $\kappa_2 = 1$ and $\kappa = 0.04$. V_n and V_l are set to 0.25 for all flows. The transition region in the dissipation model is set to 4% and 8% for the subsonic and transonic flows respectively. Far field and wall boundary conditions are used, and the boundary and interface conditions are scaled by unity, i.e. $\tau_{bc} = 1$.

ILU(0) is used during the start-up phase and ILU(1) is used during the inexact-Newton phase. The first-order preconditioner is used, and the dissipation coefficients are lumped with $\sigma = 10$. The coefficients during the start up phase are $a = 0.01$, $b = 1.2$ and $\omega = 0.05$. Also the preconditioner is updated every 3 iterations during the start-up phase. Relaxation is used only during the start-up phase and is set at $\theta = 0.6$. During the inexact-Newton phase β is chosen to be 2, and the parameter that decides when we switch to the inexact-Newton phase, τ , is set to $1/15$. The number of GMRES iterations is limited to 80 without restarts, and the solution is said to be converged when the residual norm has dropped below $1E - 11$.

The convergence summary for the ONERA M6 wing at $M=0.699$ and $\alpha=3.06^\circ$ is shown in Table 4.10, and the convergence history is shown in Figure 4.32. From the data we see that the times to convergence for the second and third-order methods are comparable. Also the number of inner iterations required to achieve convergence is similar for the second and third-order methods. The time for each residual evaluation is slightly higher for the high-order methods as expected since the high-order methods have a larger stencil. Lastly, we see that the CPU time to convergence scales nearly linearly with the number of nodes, i.e. for

[47, 56]

$$t = \kappa N^\beta \tag{4.22}$$

we find β to be close to unity. The values of β for this solver as well as others are shown in Table 4.11. The values for the other flow solvers were obtained for a different test case, and the error shown is the error in the slope for the lines of best fit. From the data shown in Figure 4.33 and Table 4.11, we see that our flow solver scales close to linearly for all three methods and scales better than the other flow solvers shown.

The drag and lift coefficients are shown in Table 4.12. The drag and lift coefficients agree within 0.2% and 3% of each other, respectively. The C_p plots at different cross sections are shown in Figure 4.34. The coarse grid is 96 blocks \times 25 \times 25 \times 25 nodes and the fine grid is 96 blocks \times 33 \times 33 \times 33 nodes. The experimental C_p data is included as well, and even though we are not using the Navier-Stokes equations and a turbulence model the agreement is quite good.

A zoomed in C_p plot for the 20% cross section is shown in Figure 4.35. We see that near the leading edge the high-order methods perform better. Also at the interface on the wing, the discontinuity in C_p is reduced noticeably. However, near the trailing edge, the coarse second-order method is closer to the finer mesh. Similar results are obtained for the other two test cases. Additionally, for the subsonic case shown in Appendix A, the high-order methods on the coarse meshes are more accurate than the second-order method on the finest mesh.

Test case	Total time(s)	Residual evaluation time(s)	Inner iterations	Equivalent residual evaluations
96 blocks×17×17×17 nodes				
2 nd order	220.3	0.06009	876	3666
3 rd order	244.3	0.10480	871	2331
4 th order	342.0	0.10879	1136	3143
96 blocks×21×21×21 nodes				
2 nd order	441.2	0.11335	1006	3892
3 rd order	506.6	0.19353	1021	2617
4 th order	643.9	0.20467	1242	3146
96 blocks×25×25×25 nodes				
2 nd order	738.6	0.19383	1064	3810
3 rd order	930.1	0.33273	1179	2795
4 th order	1081	0.34858	1325	3101
96 blocks×29×29×29 nodes				
2 nd order	1301	0.30694	1212	4238
3 rd order	1497	0.56579	1234	2645
4 th order	1962	0.55285	1520	3548
96 blocks×33×33×33 nodes				
2 nd order	2279	0.46370	1285	4914
3 rd order	2777	0.79640	1379	3486
4 th order	3604	0.83765	1704	4302

Table 4.10: Convergence summary for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$

	β		
	$M = 0.699 \ \& \ \alpha = 3.06^\circ$	$M = 0.84 \ \& \ \alpha = 3.06^\circ$	
2 nd order	-	1.15±0.04	1.23±0.08
3 rd order	-	1.19 ±0.04	1.23±0.07
4 th order	-	1.16±0.07	1.26±0.05
ARC2D	1.73	-	-
PROBE	1.325	-	-
TYPHOON	1.288	-	-

Table 4.11: Scalability of the flow solver

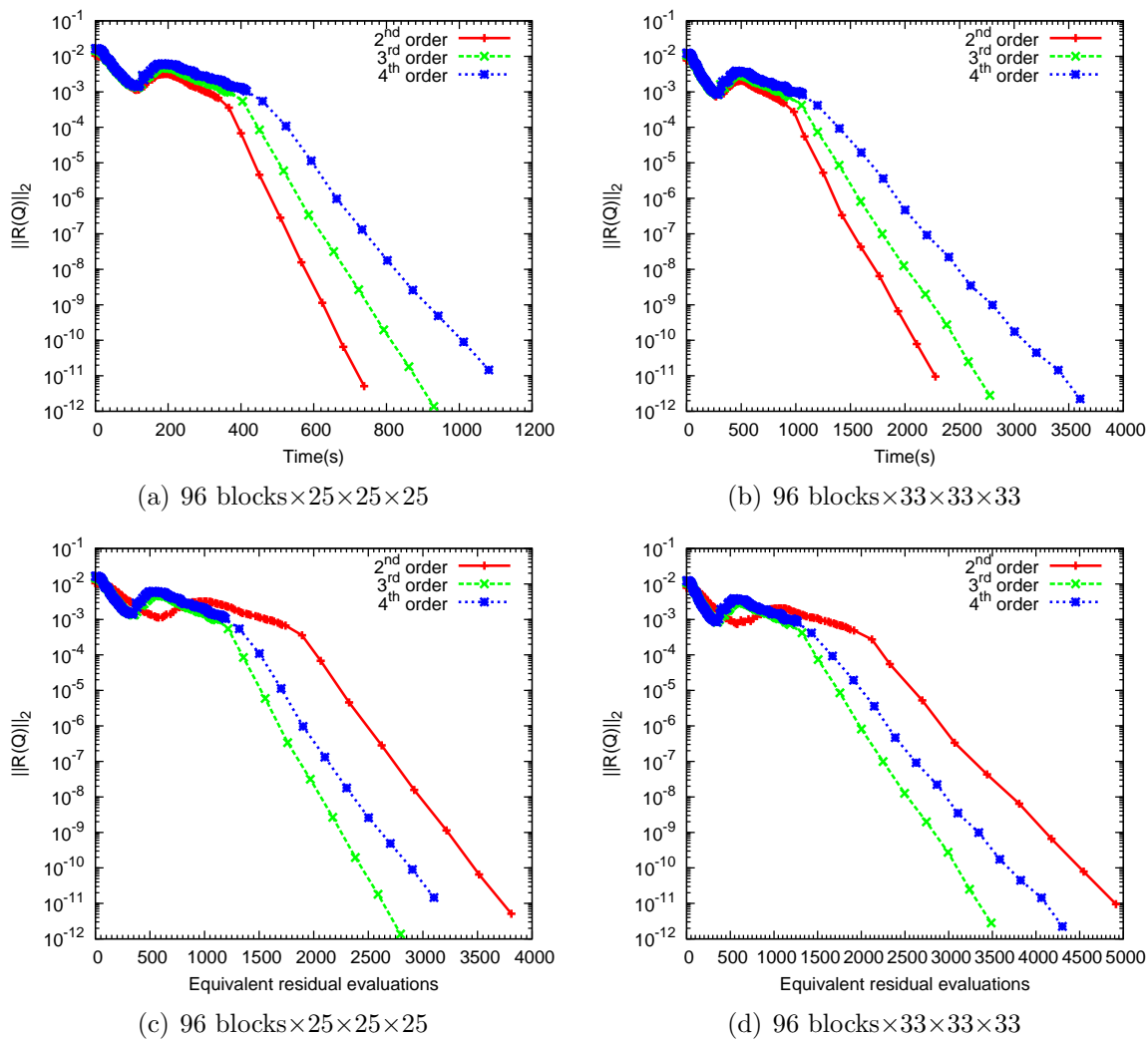


Figure 4.32: Convergence histories for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$

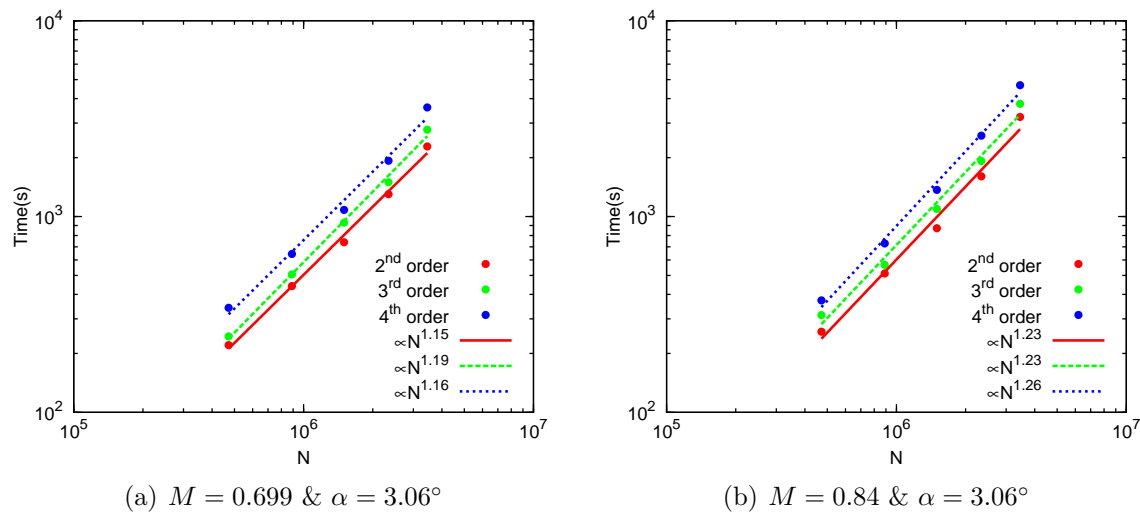


Figure 4.33: Scalability of the flow solver

Test case	Lift coefficient	Drag coefficient	% lift difference (finest mesh)	% drag difference (finest mesh)
96 blocks \times 17 \times 17 \times 17 nodes				
2 nd order	0.27763455	0.0065825086	0.83	7.2
3 rd order	0.27567148	0.0075857959	1.33	18.2
4 th order	0.27322270	0.0074356907	2.15	16.4
96 blocks \times 21 \times 21 \times 21 nodes				
2 nd order	0.27848939	0.0063827045	0.52	4.0
3 rd order	0.27709814	0.0070536488	0.82	9.9
4 th order	0.27642214	0.0068295743	1.00	6.9
96 blocks \times 25 \times 25 \times 25 nodes				
2 nd order	0.27921397	0.0062418883	0.27	1.7
3 rd order	0.27821599	0.0067114483	0.42	4.5
4 th order	0.27779249	0.0066318836	0.51	3.8
96 blocks \times 29 \times 29 \times 29 nodes				
2 nd order	0.27967114	0.0061734778	0.10	0.6
3 rd order	0.27892535	0.0065268452	0.17	1.7
4 th order	0.27872654	0.0064820850	0.18	1.4
96 blocks \times 33 \times 33 \times 33 nodes				
2 nd order	0.27997232	0.0061391219	-	-
3 rd order	0.27940949	0.0064173196	-	-
4 th order	0.27924200	0.0063872851	-	-

Table 4.12: Lift and drag coefficients for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$

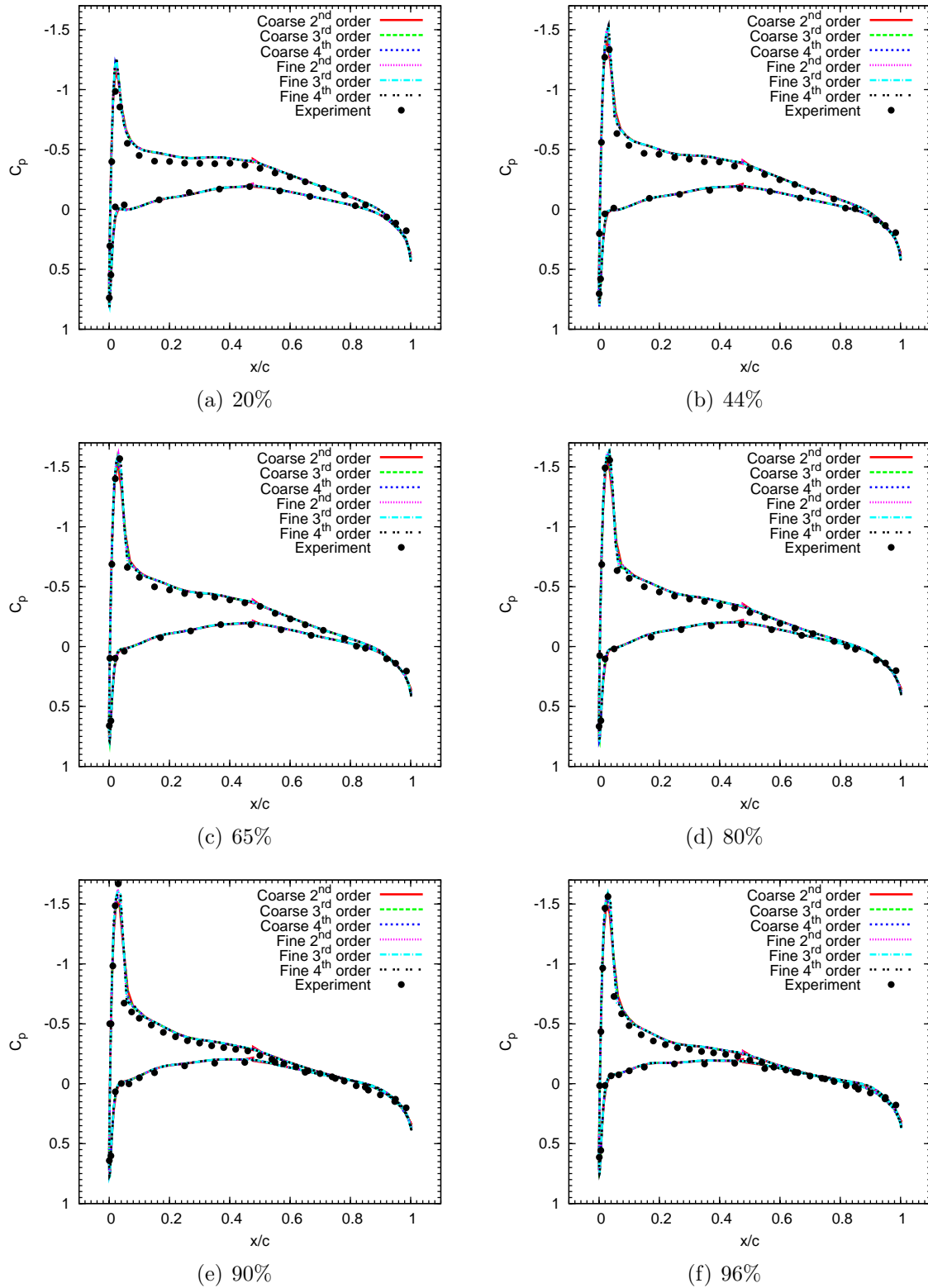


Figure 4.34: Pressure coefficient at different cross sections for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

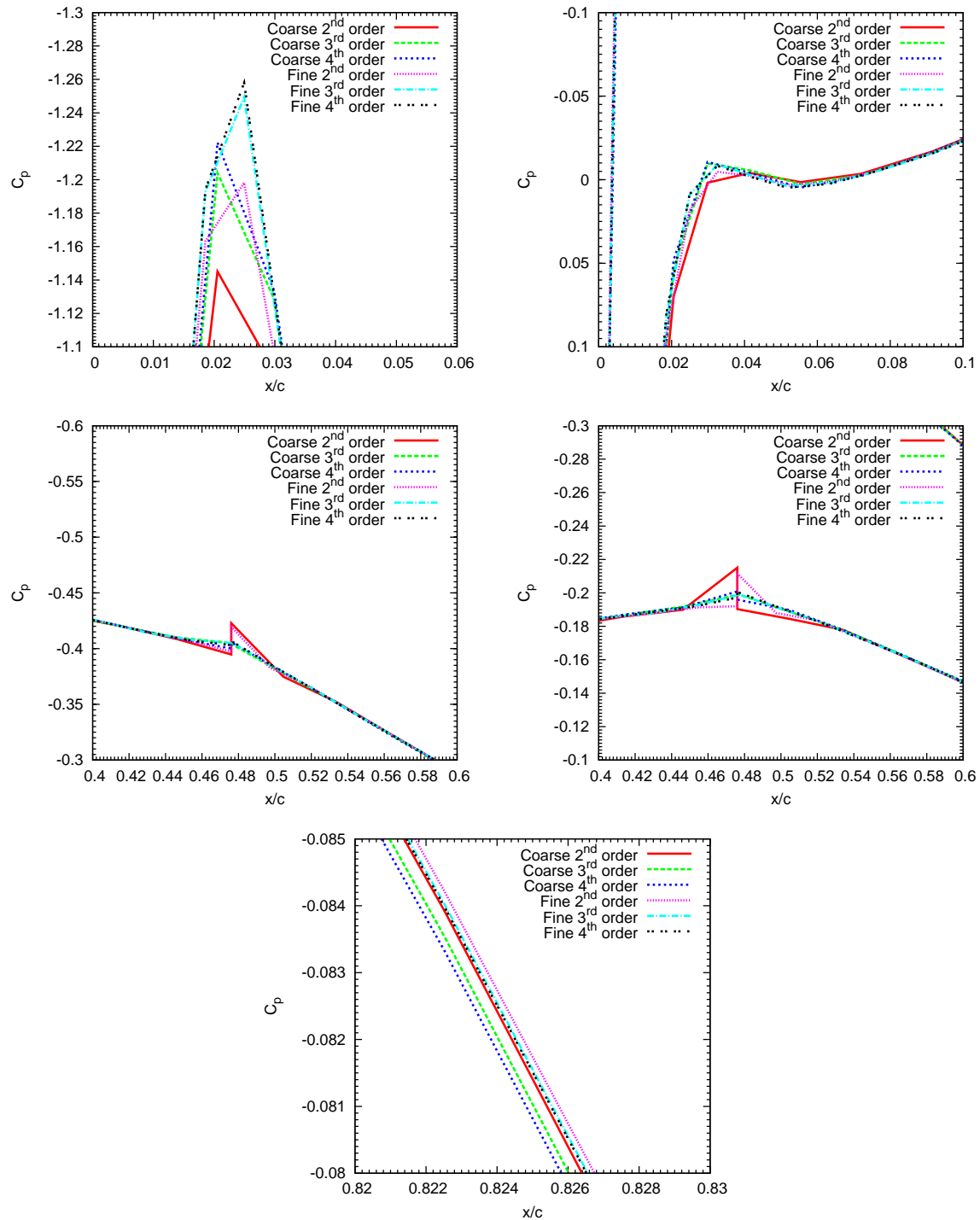


Figure 4.35: Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.699$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

A high-order finite-difference flow solver for the Euler equations was developed. Spatial discretization is carried out using summation-by-parts (SBP) operators, and the boundary and interface conditions are implemented using simultaneous approximation terms (SAT'S). Dissipation is provided using matrix and scalar dissipation. Shock capturing is enabled via first-order dissipation in the vicinity of the shock. The solution is marched to steady state using a two-stage Newton-Krylov approach. The first stage uses the first-order flow Jacobian and performs defect correction. Once a suitable iterate is found, we use the high-order Jacobian (matrix-free vector products) in the inexact-Newton phase. Force integration is performed using the trapezoidal and Simpson's rules.

The code is verified using different exact solutions, such as Ringleb flow, and supersonic and subsonic vortices. The method of manufactured solutions is also used to verify the code. It is found that the observed order of accuracy agrees with the prescribed order of accuracy. Functionals, lift and drag, were also verified using a manufactured solution and the subsonic vortex and again the observed order of accuracy matched the prescribed order of accuracy.

Efficiency studies for the Ringleb flow and the supersonic and subsonic vortices demonstrate that the high-order methods are indeed efficient in that they produce a low error per computational cost.

The code was also validated for the ONERA M6 wing. For subsonic flows, the solution using high-order methods on a coarser grid are closer to the mesh independent solution and thus high-order methods perform well. For transonic flows, the high-order methods are neither significantly better nor significantly worse.

5.2 Recommendations

This work has opened up an array of potential future work. CFD is a relatively new science and there is a lot of interest in optimization and fluid-structure interaction. A list of some of the several different paths one may embark upon are listed below,

- The next obvious extension is the implementation of the Navier-Stokes equations [72, 50].
- Implementation of a turbulence model will enable us to look at turbulence effects. Typical turbulence models that are based on the Reynolds averaged Navier-Stokes equations (RANS) are, the Baldwin-Lomax model [3], the Spalart-Allamars model [69], $k - \omega$ model [79, 80, 37] and the $k - \epsilon$ model.
- The method of manufactured solutions has been used to verify our inviscid code. An analogue can easily be developed for any set of governing equations.
- Grid adaptation is a common way of improving the quality of the solution. Several approaches are used; however, the adjoint approach appears to be promising [54, 87].
- This work demonstrated the need for smooth meshes to see high-order benefits. The grid adaptation technique mentioned above may help with mesh smoothness problems, however, an exhaustive mesh study is definitely needed.
- Mesh generation can take up a lot of user time. An automatic grid generator [31] coupled with mesh adaptation would give the flow solver a high level of automation and would allow a novice user to use CFD intelligently.
- For transonic flows the present shock capturing method is not optimal for high-order methods since there are oscillations, and it would therefore be beneficial to investigate other shock capturing methods such as WENO schemes [68].
- All the infrastructure exists to study unsteady flows. To investigate this further, a suitable time marching method will have to be used [35, 22].
- High-order methods will result in efficiency gains for optimization and mesh movement algorithms because high-order methods enable the use of coarser grids. This will help tackle even more complex problems.

- Fluid-structure interaction is an area of active research. To study fluid-structure interaction, in addition to the fluid stresses we would need to model the stress-strain behaviour on the object of interest. Fluid-structure interaction would help us look at phenomena such as wing flutter and may also give us more insight into optimizing aerodynamic structures [28].

Appendix A

ONERA M6 wing data

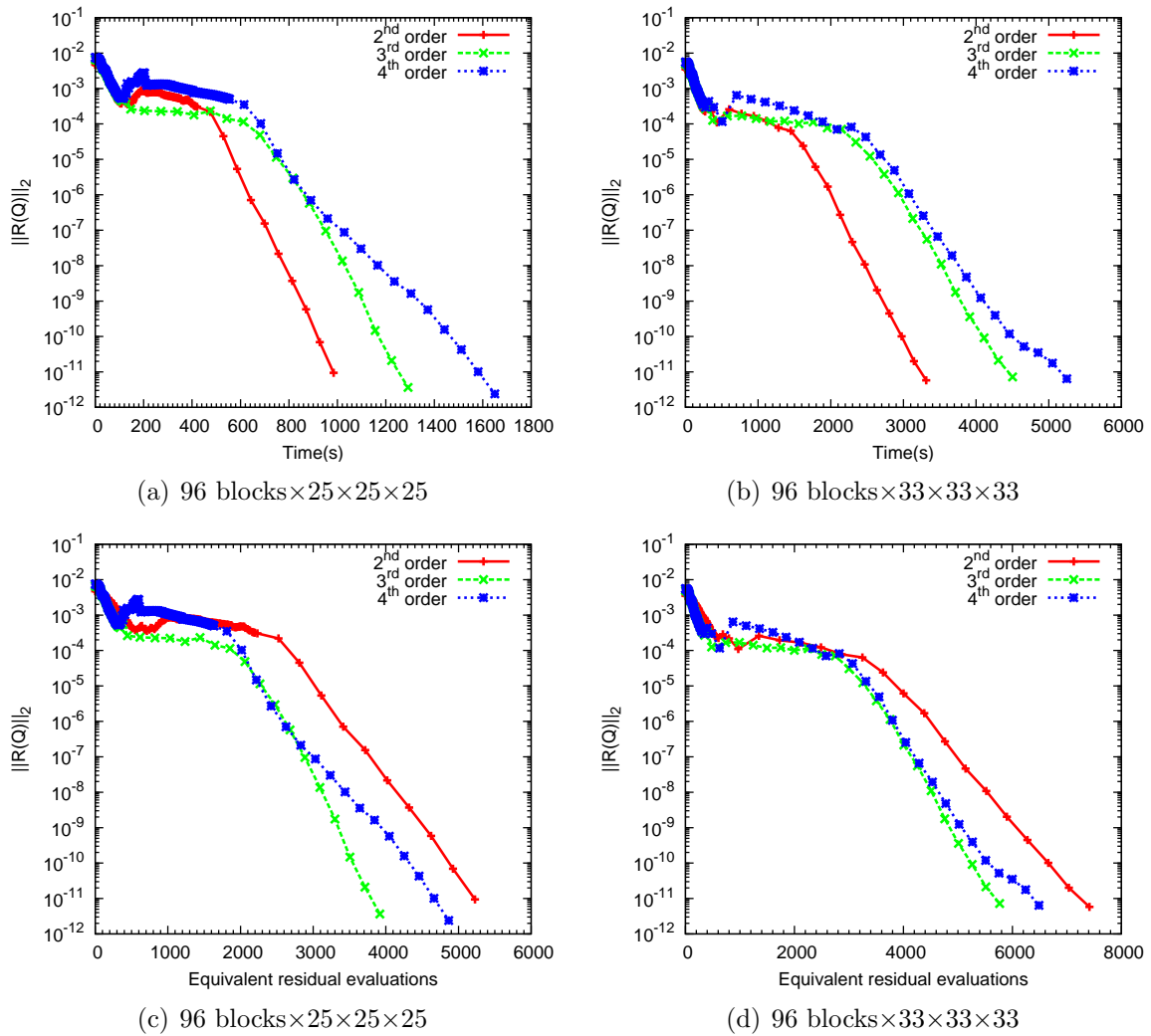


Figure A.1: Convergence histories for the ONERA M6 wing - $M=0.3$, $\alpha=4^\circ$

Test case	Lift coefficient	Drag coefficient	% lift difference (finest mesh)	% drag difference (finest mesh)
96 blocks \times 17 \times 17 \times 17 nodes				
2 nd order	0.31392843	0.0070063028	0.065	1.35
3 rd order	0.31451765	0.0084693023	0.084	19.2
4 th order	0.31447273	0.0078915101	0.077	11.1
96 blocks \times 21 \times 21 \times 21 nodes				
2 nd order	0.31403080	0.0069865033	0.033	1.62
3 rd order	0.31435295	0.0079310876	0.032	11.6
4 th order	0.31430692	0.0075018287	0.024	5.62
96 blocks \times 25 \times 25 \times 25 nodes				
2 nd order	0.31409301	0.0070269572	0.013	1.05
3 rd order	0.31429966	0.0076911958	0.015	8.29
4 th order	0.31425535	0.0074264652	0.008	4.56
96 blocks \times 29 \times 29 \times 29 nodes				
2 nd order	0.31412204	0.0070651952	0.003	0.52
3 rd order	0.31426633	0.0075564362	0.004	6.39
4 th order	0.31423398	0.0073745482	0.001	3.83
96 blocks \times 33 \times 33 \times 33 nodes				
2 nd order	0.31413459	0.0071021972	-	-
3 rd order	0.31425078	0.0074821970	-	-
4 th order	0.31422879	0.0073482131	-	-

Table A.1: Lift and drag coefficients for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$

Test case	Total time(s)	Residual evaluation time(s)	Inner iterations	Equivalent residual evaluations
96 blocks×17×17×17 nodes				
2 nd order	302.3	0.05859	1192	5159
3 rd order	341.6	0.09968	1182	3426
4 th order	456.0	0.10575	1519	4312
96 blocks×21×21×21 nodes				
2 nd order	589.2	0.11058	1318	5328
3 rd order	674.0	0.19063	1321	3535
4 th order	877.0	0.19820	1666	4424
96 blocks×25×25×25 nodes				
2 nd order	984.2	0.18831	1389	5226
3 rd order	1292	0.32974	1473	3918
4 th order	1650	0.33916	1965	4864
96 blocks×29×29×29 nodes				
2 nd order	1781	0.29841	1585	5968
3 rd order	2348	0.51895	1669	4524
4 th order	2669	0.53413	1886	4996
96 blocks×33×33×33 nodes				
2 nd order	3313	0.44689	1553	7413
3 rd order	4502	0.78108	1816	5763
4 th order	5249	0.80924	2113	6486

Table A.2: Convergence summary for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$

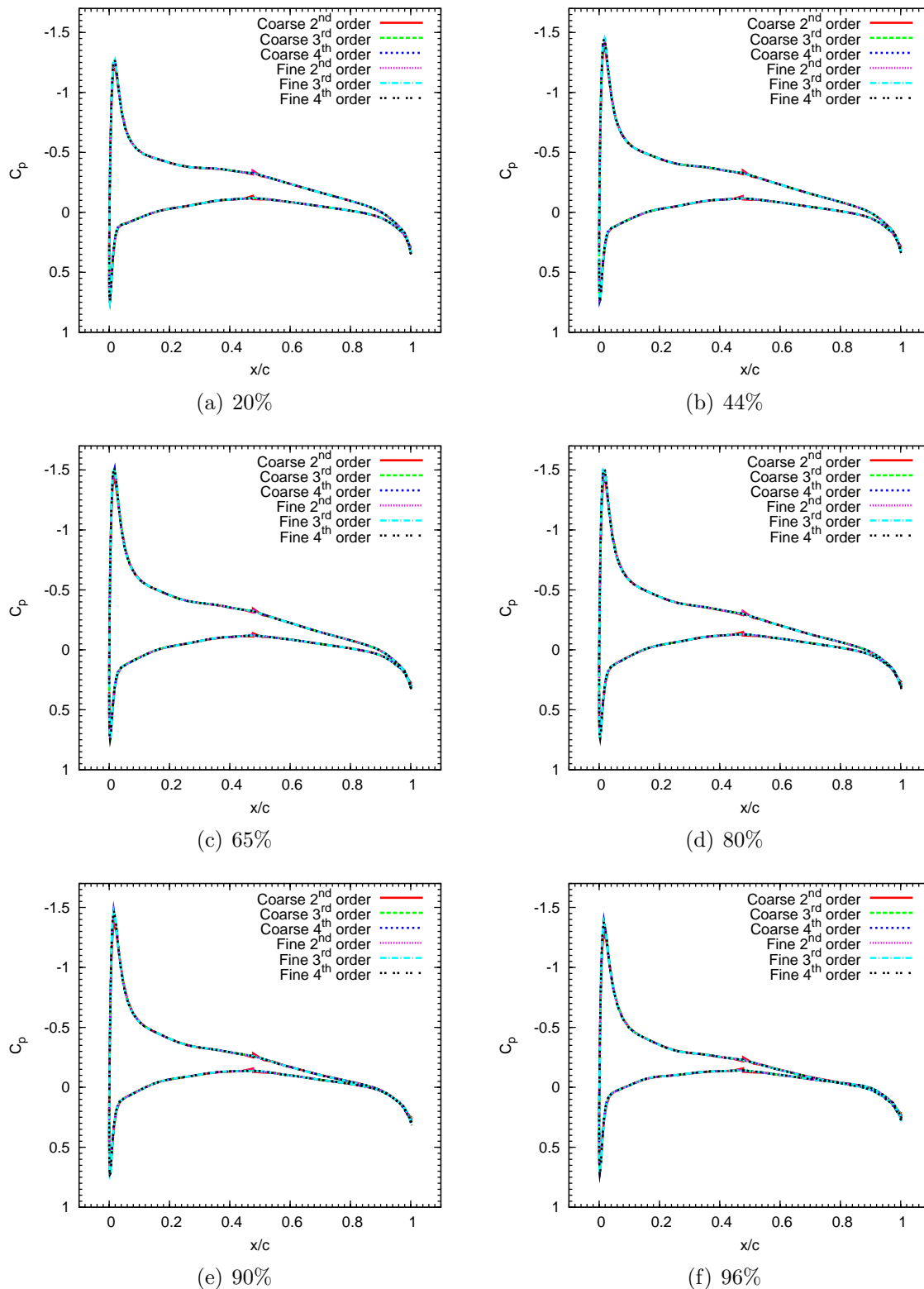


Figure A.2: Pressure coefficient at different for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

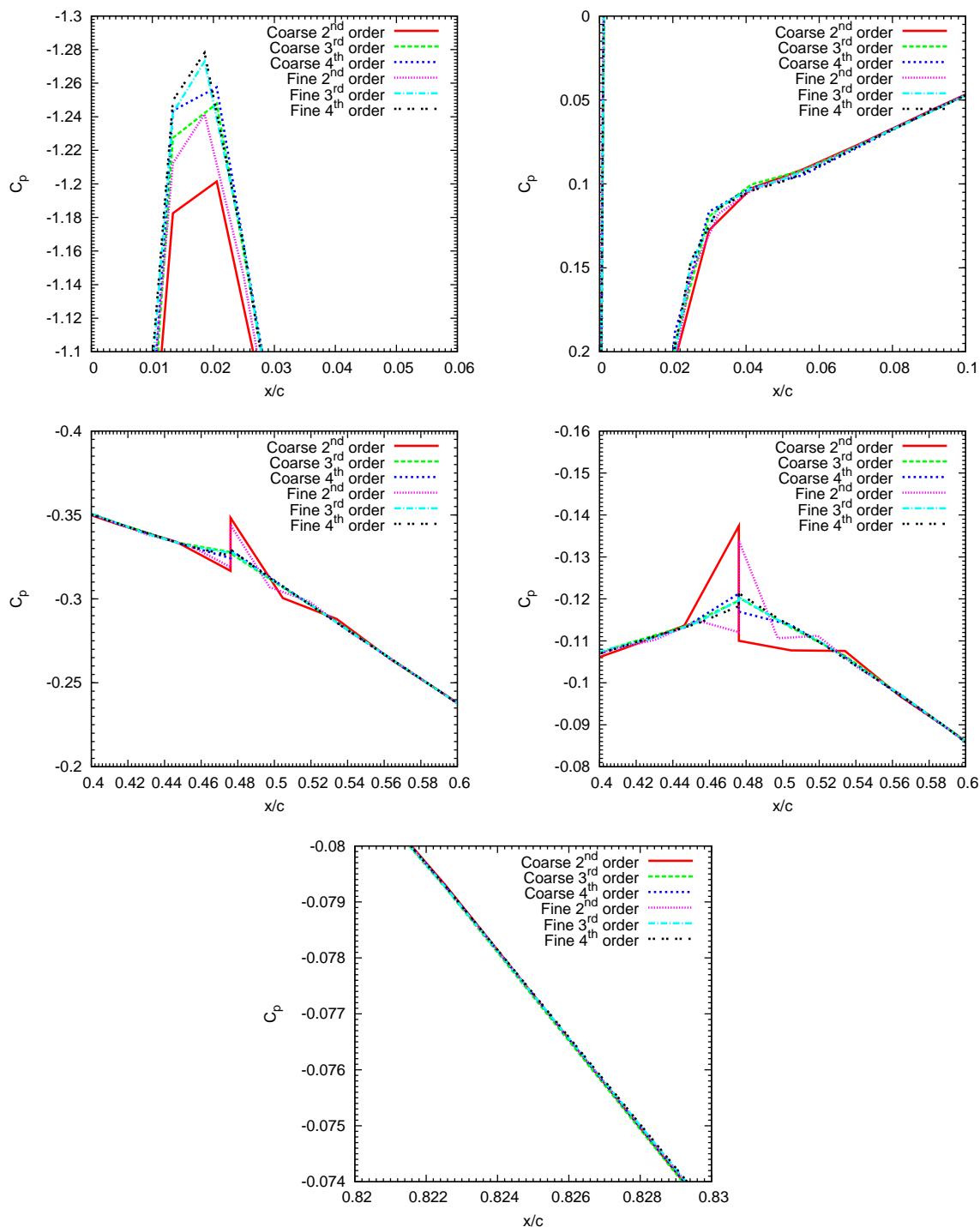


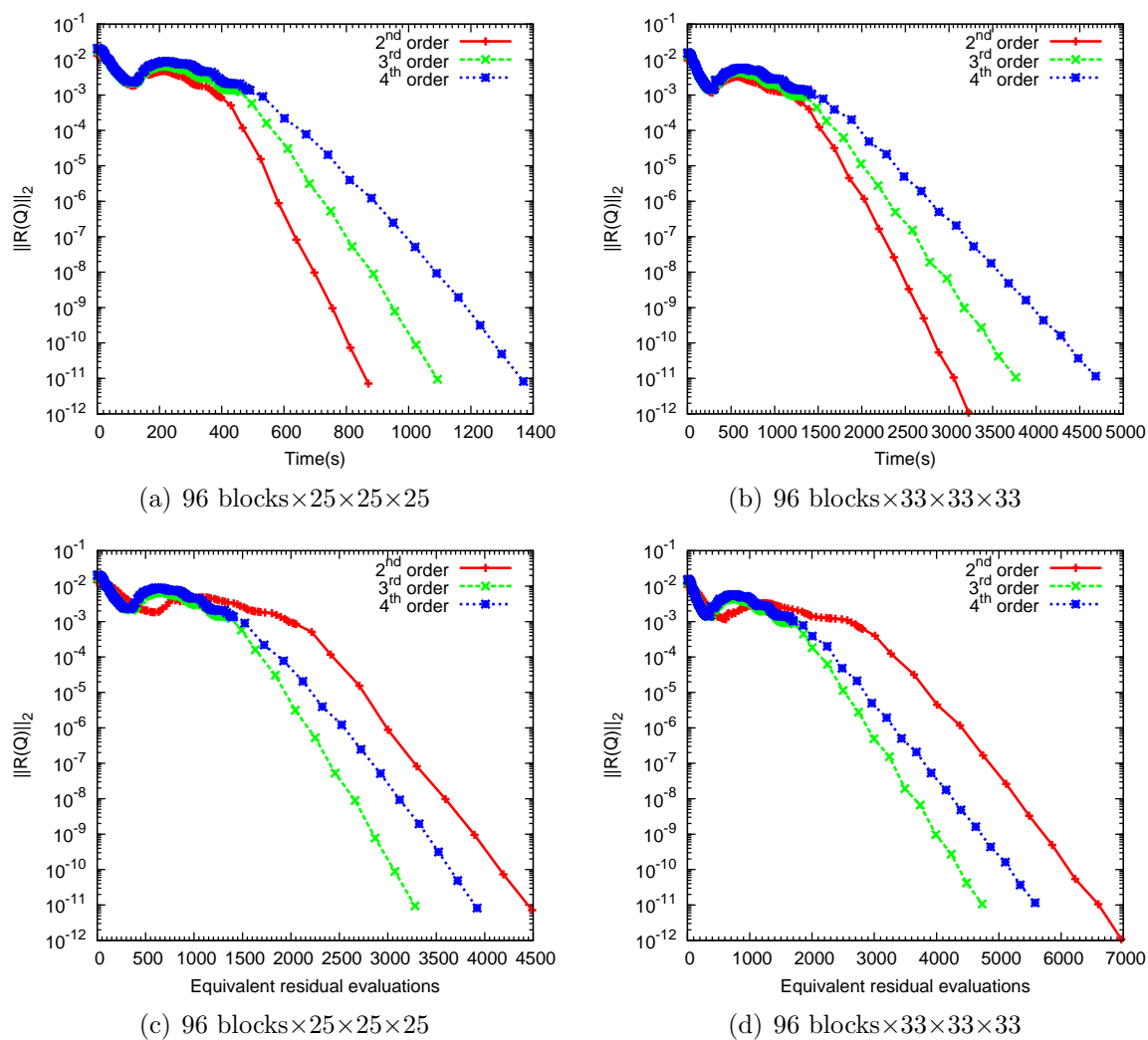
Figure A.3: Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.3$, $\alpha=4.0^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

Test case	Lift coefficient	Drag coefficient	% lift difference (finest mesh)	% drag difference (finest mesh)
96 blocks \times 17 \times 17 \times 17 nodes				
2 nd order	0.33299097	0.013007464	1.55	0.95
3 rd order	0.33200610	0.014094903	1.94	6.75
4 th order	0.32721793	0.013980653	3.37	6.07
96 blocks \times 21 \times 21 \times 21 nodes				
2 nd order	0.33513676	0.012907962	0.92	0.18
3 rd order	0.33482316	0.013651433	1.11	3.39
4 th order	0.33410331	0.013451543	1.33	2.05
96 blocks \times 25 \times 25 \times 25 nodes				
2 nd order	0.33666015	0.012870826	0.47	0.11
3 rd order	0.33674413	0.013394675	0.54	1.45
4 th order	0.33656043	0.013319976	0.61	1.06
96 blocks \times 29 \times 29 \times 29 nodes				
2 nd order	0.33763273	0.012872037	0.18	0.10
3 rd order	0.33785865	0.013271598	0.21	0.51
4 th order	0.33792925	0.013238582	0.20	0.44
96 blocks \times 33 \times 33 \times 33 nodes				
2 nd order	0.33823852	0.012885009	-	-
3 rd order	0.33857850	0.013203850	-	-
4 th order	0.33861949	0.013180798	-	-

Table A.3: Lift and drag coefficients for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$

Test case	Total time(s)	Residual evaluation time(s)	Inner iterations	Equivalent residual evaluations
96 blocks \times 17 \times 17 \times 17 nodes				
2 nd order	258.1	0.05981	1152	4315
3 rd order	313.7	0.10166	1225	3085
4 th order	373.1	0.10888	1400	3426
96 blocks \times 21 \times 21 \times 21 nodes				
2 nd order	512.0	0.11314	1295	4525
3 rd order	565.8	0.19342	1295	2925
4 th order	782.3	0.20497	1569	3816
96 blocks \times 25 \times 25 \times 25 nodes				
2 nd order	870.9	0.19397	1381	4489
3 rd order	1093	0.33318	1552	3280
4 th order	1369	0.34906	1850	3921
96 blocks \times 29 \times 29 \times 29 nodes				
2 nd order	1605	0.30629	1694	5240
3 rd order	1920	0.52717	1751	3642
4 th order	2588	0.55197	2213	4688
96 blocks \times 33 \times 33 \times 33 nodes				
2 nd order	3224	0.46277	1968	6966
3 rd order	3766	0.79590	2043	4731
4 th order	4682	0.83874	2392	5582

Table A.4: Convergence summary for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$

Figure A.4: Convergence histories for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$

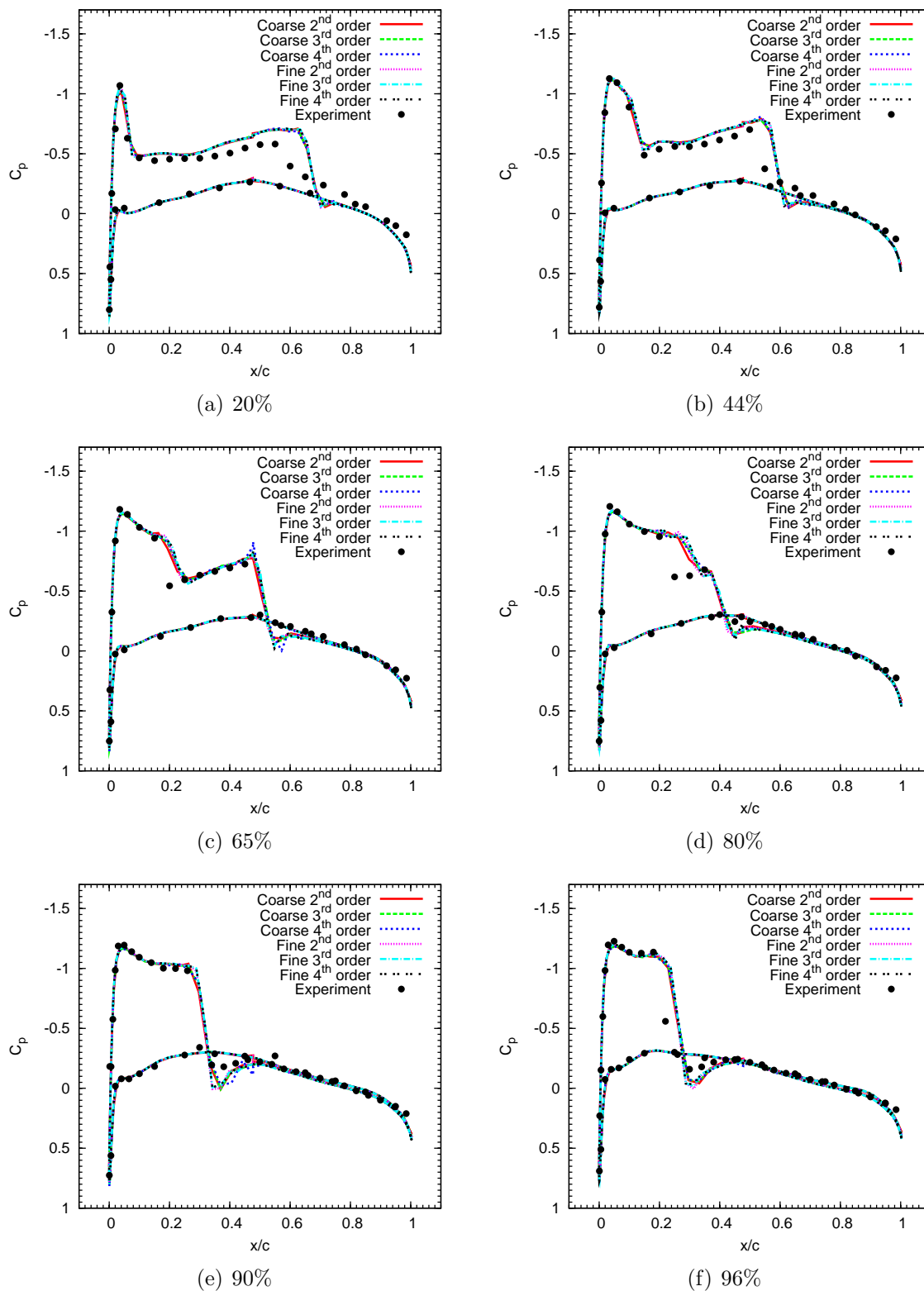


Figure A.5: Pressure coefficient at different cross sections for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

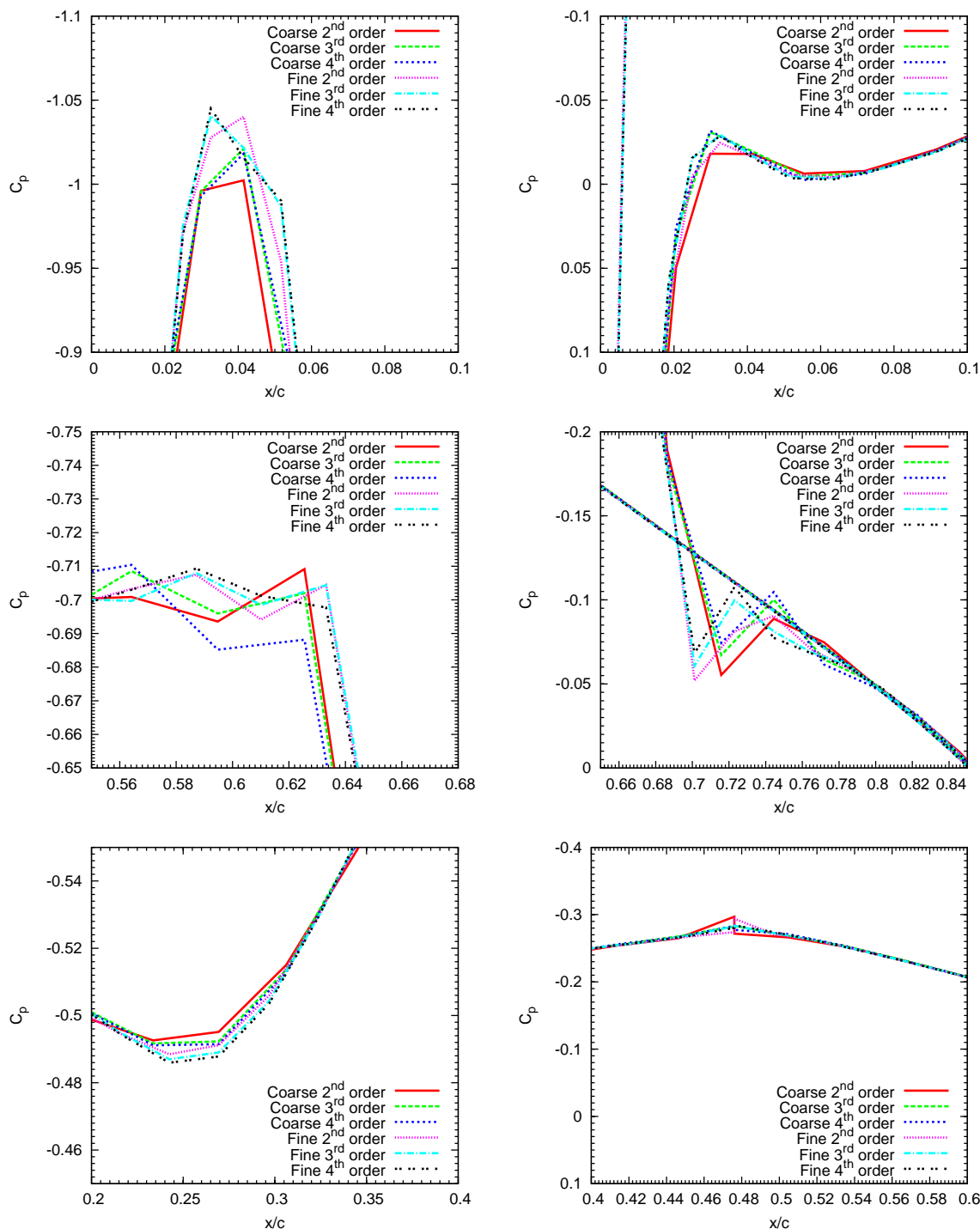


Figure A.6: Zoomed in pressure coefficient plots at the 20% cross section for the ONERA M6 wing - $M=0.84$, $\alpha=3.06^\circ$. The coarse grid has $96 \text{ blocks} \times 25 \times 25 \times 25$ nodes and the fine grid has $96 \text{ blocks} \times 33 \times 33 \times 33$ nodes.

Appendix B

High-order finite-difference operators

B.1 Third-order method

$$H_{upper} = \begin{pmatrix} \frac{17}{48} & & & & & \\ & \frac{59}{48} & & & & \\ & & \frac{43}{48} & & & \\ & & & \frac{49}{48} & & \\ & & & & 1 & \\ & & & & & \dots \end{pmatrix}$$

$$H_{lower} = \begin{pmatrix} \dots & & & & & \\ & 1 & & & & \\ & & \frac{49}{48} & & & \\ & & & \frac{43}{48} & & \\ & & & & \frac{59}{48} & \\ & & & & & \frac{17}{48} \end{pmatrix}$$

$$D_{upper} = \frac{1}{\Delta x} \begin{pmatrix} -\frac{24}{17} & \frac{59}{34} & -\frac{4}{17} & -\frac{3}{34} & & & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & & & & & \\ \frac{4}{43} & -\frac{59}{86} & 0 & \frac{59}{86} & -\frac{4}{43} & & & & \\ \frac{3}{98} & 0 & -\frac{59}{98} & 0 & \frac{32}{49} & -\frac{4}{49} & & & \\ 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \end{pmatrix}$$

$$D_{lower} = \frac{1}{\Delta x} \begin{pmatrix} \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & 0 & 0 & \\ & & \frac{4}{49} & -\frac{32}{49} & 0 & \frac{59}{98} & 0 & -\frac{3}{98} & \\ & & & \frac{4}{43} & -\frac{59}{86} & 0 & \frac{59}{86} & -\frac{4}{43} & \\ & & & & & -\frac{1}{2} & 0 & \frac{1}{2} & \\ & & & & \frac{3}{34} & \frac{4}{17} & -\frac{59}{34} & \frac{24}{17} & \end{pmatrix}$$

B.2 Fourth-order method

$$H_{upper} = \begin{pmatrix} \frac{13649}{43200} & & & & & & & & \\ & \frac{12013}{8640} & & & & & & & \\ & & \frac{2711}{4320} & & & & & & \\ & & & \frac{5359}{4320} & & & & & \\ & & & & \frac{7877}{8640} & & & & \\ & & & & & \frac{43801}{43200} & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \end{pmatrix}$$

References

- [1] M. AFTOSMIS, D. GAITONDE, AND T. S. TAVARES, *Behavior of linear reconstruction techniques on unstructured meshes*, AIAA Journal, 33 (1995), pp. 2038–2049.
- [2] M. J. AFTOSMIS AND M. J. BERGER, *Multilevel error estimation and adaptive h-refinement for Cartesian meshes with embedded boundaries*, AIAA paper 2002-0863, (2002).
- [3] B. S. BALDWIN AND H. LOMAX, *Thin layer approximation and algebraic model for separated turbulent flows*, AIAA paper 78-257, (1978).
- [4] T. J. BARTH AND S. W. LINTON, *An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation*, AIAA paper 95-0221, (1995).
- [5] G. BRASSEUR, R. COX, D. HAUGLUSTAINE, I. ISAKSEN, J. LELIEVELD, D. LISTER, R. SAUSEN, U. SCHUMANN, A. WAHNER, AND P. WIESEN, *European scientific assessment of the atmospheric effects of aircraft emissions*, Atmospheric Environment, 32 (1998), pp. 2329–2418.
- [6] J. C. BREGAINS, I. C. COLEMAN, F. ARES, AND E. MORENO, *Calculating directivities with the two-dimensional Simpson’s rule*, IEEE Antennas and Propagation Magazine, 46 (2004), pp. 106–112.
- [7] M. H. CARPENTER, D. GOTTLIEB, AND S. ABARBANEL, *Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes*, Journal of Computational Physics, 111 (1994), pp. 220–236.
- [8] M. H. CARPENTER, J. NORDSTRÖM, AND D. GOTTLIEB, *A stable and conservative interface treatment of arbitrary spatial accuracy*, Journal of Computational Physics, 148 (1999), pp. 341–365.

- [9] G. CHIOCCHIA, *Exact solutions to transonic and supersonic flows*, AGARD, AGARD-Dograph No. 211 (1985).
- [10] W. J. COIRIER AND K. G. POWELL, *An accuracy assessment of Cartesian-mesh approaches for the Euler equations*, *Journal of Computational Physics*, 117 (1995), pp. 121–131.
- [11] E. CUTHILL AND J. MCKEE, *Reducing the bandwidth of sparse symmetric matrices*, *Proceedings of the 1969 24th national conference*, (1969), pp. 157–172.
- [12] S. DE RANGO AND D. W. ZINGG, *A high-order spatial discretization for turbulent aerodynamic computations*, *AIAA Journal*, 39 (2001), pp. 1296–1304.
- [13] P. DIENER, E. N. DORBAND, E. SCHNETTER, AND M. TIGLIO, *Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions*, *Journal of Scientific Computing*, 32 (2007), pp. 109–145.
- [14] K. J. FIDKOWSKI, *A high-order discontinuous Galerkin multigrid solver for aerodynamic applications*, Master’s thesis, MIT, 2004.
- [15] ———, *A Simplex Cut-Cell Adaptive Method for High-Order Discretizations of the Compressible Navier-Stokes Equations*, PhD thesis, MIT, 2007.
- [16] K. J. FIDKOWSKI AND D. L. DARMOFAL, *A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations*, *Journal of Computational Physics*, 225 (2007), pp. 1653–1672.
- [17] 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 (2005), pp. 92–113.
- [18] J. E. HICKEN AND D. W. ZINGG, *An investigation of induced drag minimization using a Newton-Krylov algorithm*, AIAA paper 2008-5807, (2008).
- [19] ———, *A parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms*, *AIAA Journal*, 46 (2008), pp. 2773–2786.
- [20] C. HIRSCH, *Numerical computation of internal and external flows*, Wiley and Sons Ltd., London, 1991.

- [21] R. HIXON, *Numerically consistent strong conservation grid motion for finite difference schemes*, AIAA Journal, 38 (2000), pp. 1586–1593.
- [22] S. ISONO AND D. W. ZINGG, *A Runge-Kutta-Newton-Krylov algorithm for fourth-order implicit time marching applied to unsteady flows*, AIAA paper 2004-0433, (2004).
- [23] L. IVAN AND C. GROTH, *High-order central ENO finite-volume scheme with adaptive mesh refinement*, AIAA paper 2007-4323, (2007).
- [24] A. JAMESON, N. A. PIERCE, AND L. MARTINELLI, *Optimum aerodynamic design using the Navier-Stokes equations*, Theoretical and Computational Fluid Dynamics, (1998), pp. 213–237.
- [25] A. JAMESON, W. SCHMIDT, AND E. TURKEL, *Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes*, AIAA paper 81-1259, (1981).
- [26] F. T. JOHNSON, E. N. TINOCO, AND N. J. YU, *Thirty years of development and application of CFD at Boeing commercial airplanes*, Seattle, Computers & Fluids, 34 (2005), pp. 1115–1151.
- [27] S. L. KEELING, *A critical evaluation of methods for imposing solid-wall boundary conditions in inviscid flow*, tech. rep., Arnold Engineering Development Center, Arnold AFS, TN, 1994.
- [28] R. M. KIRBY, Z. YOSIBASH, AND G. E. KARNIADAKIS, *Towards stable coupling methods for high-order discretization of fluid-structure interaction: Algorithms and observations*, Journal of Computational Physics, 223 (2007), pp. 489–518.
- [29] L. D. LANDAU AND E. M. LIFSHITZ, *Fluid Mechanics (second edition) : Volume 6 of course of theoretical physics*, Elsevier, 2006.
- [30] L. LEHNER, O. REULA, AND M. TIGLIO, *Multi-block simulations in general relativity: high-order discretizations, numerical stability and applications*, Classical and Quantum Gravity, 22 (2005), pp. 5283–5321.
- [31] R. LÖHNER, *Automatic unstructured grid generators*, Finite Elements in Analysis and Design, 25 (1997), pp. 111–134.

- [32] H. LOMAX, T. H. PULLIAM, AND D. W. ZINGG, *Fundamentals of computational fluid dynamics*, Springer, 2001.
- [33] K. MATTSSON, *Boundary procedures for summation-by-parts operators*, Journal of Scientific Computing, 18 (2003), pp. 133–153.
- [34] K. MATTSSON AND J. NORDSTRÖM, *Summation by parts operators for finite difference approximations of second derivatives*, Journal of Computational Physics, 199 (2004), pp. 503–540.
- [35] K. MATTSSON, M. SVÄRD, M. CARPENTER, AND J. NORDSTRÖM, *High-order accurate computations for unsteady aerodynamics*, Computers & Fluids, 36 (2007), pp. 636–649.
- [36] K. MATTSSON, M. SVÄRD, AND J. NORDSTRÖM, *Stable and accurate artificial dissipation*, Journal of Scientific Computing, 21 (2004), pp. 57–79.
- [37] F. R. MENTER, *Two-equation eddy-viscosity turbulence models for engineering applications*, AIAA Journal, 32 (1994), pp. 269–289.
- [38] A. NEJAT AND C. OLLIVIER-GOOCH, *A high-order accurate unstructured Newton-Krylov solver for inviscid compressible flows*, AIAA paper 2006-3711, (2006).
- [39] ———, *On preconditioning of Newton-GMRES algorithm for a high-order accurate unstructured solver*, in Proceedings of the Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada, 2006.
- [40] ———, *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 (2008), pp. 2366–2386.
- [41] ———, *A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows*, Journal of Computational Physics, 227 (2008), pp. 2582–2609.
- [42] A. NEJAT, C. OLLIVIER-GOOCH, AND K. MICHALAK, *Accuracy assessment methodology for a higher-order unstructured finite volume solver*, AIAA paper 2007-4194, (2007).
- [43] M. NEMEC AND M. J. AFTOSMIS, *Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes*, AIAA paper 2007-4187, (2007).

- [44] M. NEMEC AND D. W. ZINGG, *Towards efficient aerodynamic shape optimization based on the Navier-Stokes equations*, AIAA paper 2001-2532, (2001).
- [45] ———, *A Newton-Krylov algorithm for aerodynamic design using the Navier-Stokes equations*, AIAA Journal, 40 (2002), pp. 1146–1154.
- [46] M. NEMEC, D. W. ZINGG, AND T. H. PULLIAM, *Multipoint and multi-objective aerodynamic shape optimization*, AIAA Journal, 42 (2004), pp. 1057–1065.
- [47] J. C. NICHOLS, *A three-multidimensional multi-block Newton-Krylov flow solver for the Euler equations*, Master’s thesis, University of Toronto, 2004.
- [48] E. J. NIELSEN AND W. K. ANDERSON, *Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations*, AIAA paper 98-4809, (1998).
- [49] J. NORDSTRÖM AND M. H. CARPENTER, *Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier-Stokes equations*, Journal of Computational Physics, 148 (1999), pp. 621–645.
- [50] J. NORDSTRÖM AND M. SVÄRD, *Well-posed boundary conditions for the Navier–Stokes equations*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 1231–1255.
- [51] P. OLSSON, *Summation by parts, projections, and stability I*, Mathematics of Computation, 64 (1995), pp. 1035–1065.
- [52] ———, *Summation by parts, projections, and stability. II*, Mathematics of Computation, 64 (1995), pp. 1473–1493.
- [53] P. OLSSON AND S. L. JOHANSSON, *A study of dissipation operators for the Euler equations and a three-dimensional channel flow*, in Supercomputing ’89: Proceedings of the 1989 ACM/IEEE conference on Supercomputing, New York, NY, USA, 1989, ACM, pp. 141–151.
- [54] M. A. PARK, *Adjoint-based, three-dimensional error prediction and grid adaptation*, AIAA paper 2002–3286, (2002).
- [55] J. E. PENNER, *Aviation and the global atmosphere*, Cambridge University Press, 1999.
- [56] A. PUEYO, *An efficient Newton-Krylov method for the Euler and Navier-Stokes equations*, PhD thesis, University of Toronto, 1998.

- [57] A. PUEYO AND D. W. ZINGG, *Efficient NewtonKrylov solver for aerodynamic computations*, AIAA Journal, 36 (1998).
- [58] T. H. PULLIAM, *Artificial dissipation models for the Euler equations*, AIAA Journal, 24 (1986), pp. 1931–1940.
- [59] —, *Efficient solution methods for the Navier-Stokes Equations-Lecture notes for the Von Karman institute for fluid dynamics lecture series: Numerical techniques for viscous flow computation in turbomachinery bladings*, 1986.
- [60] T. H. PULLIAM AND J. L. STEGER, *Implicit finite-difference simulatons and three-dimensional compressible flow*, AIAA Journal, 18 (1980), pp. 159–167.
- [61] P. J. ROACHE, *Verification and validation in computational science and engineering*, Hermosa, 1998.
- [62] C. J. ROY, C. C. NELSON, T. SMITH, AND C. C. OBER, *Verification of Euler/Navier-Stokes codes using the method of manufactured solutions*, International Journal for Numerical Methods in Fluids, 44 (2004), pp. 599–620.
- [63] M. P. RUMPFKEIL AND D. W. ZINGG, *Unsteady optimization using a discrete adjoining approach applied to aeroacoustis shape design*, AIAA paper 2008-18, (2008).
- [64] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, Journal of Scientific Computing, 14 (1993), pp. 461–469.
- [65] —, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.
- [66] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, Journal of Scientific Computing, 7 (1986), pp. 856–869.
- [67] M. D. SALAS, *Some observations on grid convergence*, Computers & Fluids, 35 (2006), pp. 688–692.
- [68] C. SHU, *High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD*, Journal of Computational Fluid Dynamics, 17 (2001), pp. 107–118.

- [69] P. R. SPALART AND S. R. ALLMARAS, *A one-equation turbulence model for aerodynamic flows*, AIAA paper 92-0439, (Jan. 1992).
- [70] B. STRAND, *Summation by parts for finite difference approximations for d/dx* , Journal of Computational Physics, 110 (1994), pp. 47–67.
- [71] M. SVÄRD, *On coordinate transformations for summation-by-parts operators*, Journal of Scientific Computing, 20 (2004), pp. 29–42.
- [72] M. SVÄRD, M. H. CARPENTER, AND J. NORDSTRÖM, *A stable high-order finite difference scheme for the compressible Navier-Stokes equations, far-field boundary conditions*, Journal of Computational Physics, 225 (2007), pp. 1020–1038.
- [73] R. C. SWANSON AND E. TURKEL, *On central-difference and upwind schemes*, Journal of Computational Physics, 101 (1992), pp. 292–306.
- [74] P. D. THOMAS AND C. K. LOMBARD, *Geometric conservation law and its application to flow computations on moving grids*, AIAA Journal., 17 (1979), pp. 1030–1037.
- [75] J. M. VAASSEN, D. VIGNERON, AND J. A. ESSERS, *An implicit high order finite volume scheme for the solution of 3D Navier-Stokes equations with new discretization of diffusive terms*, Journal of Computational and Applied Mathematics, 215 (2008), pp. 595–601.
- [76] D. A. VENDITTI AND D. L. DARMOFAL, *Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow*, Journal of Computational Physics, 164 (2000), pp. 204–227.
- [77] M. R. VISBAL AND D. V. GAITONDE, *On the use of higher-order finite-difference schemes on curvilinear and deforming meshes*, Journal of Computational Physics, 181 (2002), pp. 155–185.
- [78] F. M. WHITE, *Fluid Mechanics*, McGraw-Hill Science Engineering, 2003.
- [79] D. C. WILCOX, *Re-assessment of the scale-determining equation for advanced turbulence models*, AIAA Journal, 26 (1988), pp. 1414–1421.
- [80] ———, *Turbulence modelling*, DCW Industries, 2006.

- [81] D. W. ZINGG AND L. BILLING, *Toward practical aerodynamic design through numerical optimization*, AIAA paper 2007-3950, (2007).
- [82] D. W. ZINGG, L. DIOSADY, AND L. BILLING, *Adaptive airfoils for drag reduction at transonic speeds*, AIAA paper 2006-3656, (2006).
- [83] D. W. ZINGG AND S. ELIAS, *On aerodynamic optimization under a range of operating conditions*, AIAA Journal, 44 (2006), pp. 2787–2792.
- [84] D. W. ZINGG, T. LEUNG, L. DIOSADY, A. H. TRUONG, S. ELIAS, AND M. NEMEC, *Improvements to a Newton-Krylov algorithm for aerodynamic optimization*, AIAA paper 2005-4857, (2005).
- [85] D. W. ZINGG AND T. H. PULLIAM, *Foundations of Computational Fluid Dynamics*, Unpublished, 2007.
- [86] D. W. ZINGG, S. D. RANGO, M. NEMEC, AND T. H. PULLIAM, *Comparison of several spatial discretizations for the Navier-Stokes equations*, Journal of Computational Physics, 160 (2000), pp. 683–704.
- [87] D. W. ZINGG AND P. WALSH, *Solution adaptation of unstructured grids for two-dimensional aerodynamic computations*, AIAA Journal, 39 (2001), pp. 831–837.