

TOWARDS INDUSTRY-READY HIGH-ORDER FLOW SOLVERS:
INCREASING ROBUSTNESS AND USABILITY

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND
ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Manuel Rodrigo López Morales
May 2016

Abstract

High-order methods are truer to the flow physics and more accurate per degree of freedom than low-order methods. Their high computational intensity relative to their communication requirements makes them prime candidates for implementation in the latest hyper-parallel computing architectures like Graphical Processing Units (GPUs). Why are they not more prevalent in the toolset of design teams?

One of the main barriers to wide adoption of high-order numerical methods in industrial applications is the schemes' low *robustness* relative to low-order methods [83]. Their stability is highly dependent on the quality of the grid, even when the solution is relatively smooth. Aliasing, underresolution, and non-smoothness are the main causes of instabilities.

This dissertation proposes solutions from two fronts: the creation of a set of families of numerical schemes with guaranteed linear stability and tunable dispersion-dissipation properties, and the formulation of filters with spectral effectiveness and element-local stencil.

To ease the adoption, or increase *usability*, of high-order methods in industry and academia, a well-documented, validated, verified, and constantly improved source code is needed. This dissertation walks the reader through the efforts by the Aerospace Computing Laboratory (ACL) to provide such code: High-Fidelity Large Eddy Simulation (LES) Open Source Code (HiFiLES) (hifiles.stanford.edu).

Acknowledgments

Mentorship by Peter Vincent, Patrice Castonguay, and David Williams was essential for my involvement in the intellectually stimulating topic of high order methods. David's tenacity, talent, and example helped me keep the passion for modeling and simulation alight. The work of Guido Lodato served as an inspiration for many of the topics of this thesis. Jonathan Bull's application of high order methods to turbulent flows served as guidance for the selection of capabilities that I would like to see in a Navier-Stokes solver.

Prof. Antony Jameson provided me with an invaluable environment of intellectual freedom framed by his sharing of pioneering ideas and insightful recommendations. His mastery of numerical analysis and crispness of communication are traits I shall strive to emulate.

Feedback and questions by Prof. Juan Alonso and Prof. Charbel Farhat have helped me contextualize the contributions in this dissertation in the larger scopes of Aerospace Engineering and Computational Fluid Dynamics. Comments by Prof. Parviz Moin and Prof. Brian Cantwell have encouraged my desire to apply the methods discussed here to simulations relevant to the understanding of flow physics.

Friendship and comradeship with Abhishek Sheshadri, Kartikey Asthana, Jerry Watkins II, Josh Romero, Jacob Crabill, and David Manosalvas made it enjoyable working in the Aerospace Computing Laboratory. I relied on their knowledge and eagerness to help to move past theoretical and implementation hurdles.

Support for this work was provided by the Aeronautics and Astronautics Departmental Fellowship and the Air Force Office of Scientific Research under grant # FA9550-14-1-0186, monitored by Dr. Fariba Fahroo.

Abbreviations

- Ma** Mach number. xiii, xiv, xvi, xix, 3, 6, 19–23, 27, 31, 34, 38, 41, 44, 45, 86, 104, 108–110, 112–115, 124, 126
- Re** Reynolds number. x, xiii, xv, xvi, xix, 3, 4, 6, 19, 22, 27, 31, 33, 35, 36, 38, 44, 45, 86, 88, 104, 105, 107, 109–113, 115, 124, 126
- Pr** Prandtl number. 10, 11, 25
- St** Strouhal number. xix, xx, 114, 115, 117–124
- HiFiLES** High-Fidelity LES Open Source Code. iv, ix, xii, xiv, 3–12, 16, 18, 23, 24, 26–31, 33, 34, 36, 38, 41, 46, 88, 104, 107, 108, 126, 128, 129
- ACL** Aerospace Computing Laboratory. iv, 3, 5, 7, 9, 128
- AIAA** American Institute of Aeronautics and Astronautics. 9
- AoA** Angle of Attack. xiv, 20, 22
- AUSM+** Advection Upstream Splitting Method Plus. 82
- AV** Artificial Viscosity. xiv, 23, 87, 88
- C1FR** C^1 Flux Reconstruction. xvi, xvii, 66, 70–74, 76–80, 82–85, 128
- CFD** Computational Fluid Dynamics. 12, 38, 46, 86
- CFL** Courant-Friedrichs-Lowy. xvii, 16, 46, 74, 77, 78, 80, 82, 83

- CMFR** C^m Flux Reconstruction. ix, 4–6, 47–49, 54, 57, 65, 66, 84, 85, 128, 129
- DG** Discontinuous-Galerkin. xv–xvii, 14, 16, 38–40, 44, 45, 47–49, 71–76, 78, 79, 81, 82, 84, 85, 87, 125
- DNS** Direct Numerical Simulation. xv, 8, 39, 87, 125
- DoF** Degrees of Freedom. 41, 113–115, 117
- DRP** Dispersion Relation-Preserving. 38
- ELED** Essentially-Local-Extremum-Diminishing. 89
- ESFR** Energy Stable Flux Reconstruction (FR). 4, 8, 16, 46–49, 66
- FR** Flux Reconstruction. vii, ix, xvi, 3–5, 12, 14–18, 24, 38, 41, 44–49, 53, 55, 68, 72, 84, 85, 88, 125
- GPU** Graphical Processing Unit. 5, 8, 24, 46, 88, 104, 105, 107, 108, 126
- HLLE** Harten-Lax-van Leer and Einfeldt. 82
- JST** Jameson-Schmidt-Turkel. 89
- LDG** Local Discontinuous Galerkin. 104
- LDV** Laser Doppler Velocimetry. xv, 41, 43
- LES** Large Eddy Simulation. iv, vi, 2–4, 7, 8, 22, 41, 46, 48, 87, 125
- LFS** Local Fourier Spectral. ix, xi, 4, 6, 86–90, 99, 111, 113, 117, 124–129
- MMS** Method of Manufactured Solutions. 8, 24, 46
- MPI** Message Passing Interface. 46
- NS** Navier-Stokes. 9–12, 24, 27, 31, 46, 87, 89, 104, 125

- PDE** Partial Differential Equation. 24, 48, 54, 73
- RANS** Reynolds Averaged Navier-Stokes. viii, 11, 113
- RK** Runge-Kutta. viii, 18, 41, 71, 74, 78
- RK45** Runge-Kutta (RK) 45. 18, 25, 104
- SA** Spallart-Allmaras. x, xvi, 11, 19, 44, 45
- SD** Spectral Difference. xv, 16, 38, 41, 43, 47, 48, 87, 125, 126
- SGS** Sub-Grid Scale. 3, 7, 22, 24, 46, 87, 124–126
- TGV** Taylor-Green Vortex. xv, 38, 40
- URANS** Unsteady Reynolds Averaged Navier-Stokes (RANS). 113
- V&V** Validation and Verification. 5, 7–9
- WALE** Wall-Adapting Local Eddy-Viscosity. viii, xv, 23, 24
- WSM** Wall-Adapting Local Eddy-Viscosity (WALE)-Similarity. xv, 24, 41, 43

Contents

Abstract	iv
Acknowledgments	v
1 Introduction	1
1.1 The Role of High-Order Methods in the Future of Industrial Simulations	1
1.2 Motivation	3
1.3 Contributions	5
1.3.1 Maintenance and Validation and Verification of HiFiLES	5
1.3.2 Tackling Stability via Numerics: Creation of Provably Linearly Stable C^m Flux Reconstruction (CMFR) Schemes	5
1.3.3 Tackling Stability via Filtering: Creation of Generalized Local Fourier Spectral (LFS) Filters	6
2 HiFiLES: an open source GPU-powered, High-Order Large-Eddy Simulation code	7
2.1 Introduction	7
2.2 Governing Equations	9
2.2.1 Navier Stokes equations	9
2.2.2 Reynolds Averaged Navier-Stokes equations	10
2.3 Numerical Methods	11
2.3.1 FR Method	12
2.3.2 Shock Capturing and Stabilization Models	18

2.3.3	Spallart-Allmaras (SA) Turbulence Model and Negative $\tilde{\nu}$ Modification	19
2.3.4	Large Eddy Simulation	22
2.3.5	Computing Architecture and Scalability	24
2.4	Verification and Validation	24
2.4.1	Method of Manufactured Solutions	24
2.4.2	Subsonic laminar flat-plate	27
2.4.3	Circular Cylinder	31
2.4.4	SD7003 airfoil at 4° angle of attack	34
2.4.5	SD7003 wing section at 4° angle of attack	36
2.4.6	Taylor-Green Vortex at Reynolds number (Re) = 1,600	38
2.4.7	LES of Flow Over a Square Cylinder at Re = 21,400	41
2.4.8	NACA 0012 airfoil at 0° angle of attack, Re = 6 million, Ma = 0.15	44
2.5	Conclusion	46
3	Flux Reconstruction Schemes with Corrected Fluxes Continuous in m Derivatives	47
3.1	Introduction	47
3.2	Background	48
3.3	The C^m Flux Reconstruction Approach	50
3.3.1	Preliminaries: the general advection equation	50
3.3.2	CMFR schemes for the general advection-diffusion equation .	53
3.4	Linear stability of C^m continuous flux reconstruction ($\hat{f} = a\hat{u}$)	54
3.4.1	Part 1	58
3.4.2	Part 2	64
3.5	C^1 linearly stable flux reconstruction family of schemes	66
3.5.1	C1FR for the linear advection equation	67
3.6	Numerical test cases	69
3.6.1	Order of Accuracy of C1FR	71
3.6.2	Advection-Diffusion Energy Preservation	72

3.6.3	Solutions to the Euler Equations	76
3.7	Conclusions	84
4	Local Fourier Spectral Filters	86
4.1	Introduction	86
4.2	Local Fourier Spectral Filters	88
4.2.1	Desired Properties	88
4.2.2	Mechanics of LFS Filters	89
4.2.3	Internal Filtering Component	91
4.2.4	Boundary Filtering Component	95
4.3	Visualization of the LFS Filters in Triangular Elements	99
4.3.1	Internal Filtering Components	100
4.3.2	Boundary Filtering Components	100
4.3.3	Filtered Solutions	102
4.4	Results	104
4.4.1	Stabilization Strategy	106
4.4.2	Coarse mesh used in simulations	106
4.4.3	Flow Around a Circular Cylinder, Re = 10⁶, Ma = 0.2	107
4.4.4	High Reynolds Number, Flow Around a Circular Cylinder, Re = 10⁶, Ma = 0.077	108
4.4.5	High Reynolds Number, Flow Around a Circular Cylinder, Re = 10⁶, Ma = 0.87	109
4.4.6	High Reynolds Number, Flow Around a Circular Cylinder, Re = 10⁶, Ma = 0.0077 , less-coarse mesh	111
4.4.7	Effects of Filtering in a Well-resolved Simulation	111
4.5	Conclusion	125
4.6	Future Work	126
5	Conclusion	128
5.0.1	Future Work	129
Bibliography		130

List of Tables

2.1	Accuracy of HiFiLES for NS equations with source term in tetrahedral meshes at $t = 10$. L_2 error is the L_2 -norm of the error in the energy field: ρe	26
2.2	Accuracy of HiFiLES for NS equations with source term in tetrahedral meshes at $t = 10$. L_2 error is the L_2 -norm of the error in the gradient of the energy field: $\frac{\partial}{\partial x_i}(\rho e)$	26
2.3	Accuracy of HiFiLES for NS equations with source term in triangular meshes at $t = 1$. L_2 error is the L_2 -norm of the error in the energy field: ρe	27
2.4	Accuracy of HiFiLES for NS equations with source term in triangular meshes at $t = 1$. L_2 error is the L_2 -norm of the error in the gradient of the energy field: $\frac{\partial}{\partial x_i}(\rho e)$	28
2.5	HiFiLES convergence using different grids and polynomial order. \times / \checkmark indicates not converged/converged resp.	28
2.6	Time-averaged values of the lift and drag coefficients for the SD7003 airfoil flows with $Re = 10,000, 22,000, 60,000$	34
2.7	Time-averaged values of the lift and drag coefficients for the SD7003 wing-section in a flow with $Re = 10,000$	36
4.1	Number of internal points N_s , number of boundary points N_f , and number of equations and unknowns in matrix \mathcal{B} for each type of element, assuming the solution is being discretized with a polynomial of degree p	97

4.2 Summary of simulation results. All cases were run at $\text{Re} = 1e6$ with polynomial representation of order 4. Cases 1-3 were run using the mesh shown in Figure 4.8. Case 4 was run using the mesh shown in Figure 4.13. Cases with 0* Strouhal number reached an artificial steady-state.	105
4.3 Simulation results that illustrate effect of filtering, changing meshes, and varying the spatial order of accuracy. All cases were run at Mach number (Ma) = 0.1, $\text{Re} = 3.9e3$	115

List of Figures

2.1	density contours for viscous flow at $\text{Ma} = 1.2$ over a NACA 0012 airfoil at 5° Angle of Attack (AoA) with polynomial order 6	20
2.2	Energy contours	20
2.3	Figure shows the elemental shock “sensor” for the $\text{Ma} = 1.2$ viscous case shown in figure 2.1. The shock sensor is just the maximum value of the enhanced kernel in each element	21
2.4	Ma contours for inviscid flow over NACA0012 at $\text{Ma} = 1.6$ and $\text{AoA} = 0^\circ$ on a triangle-mesh using Persson and Peraire’s method and using artificial viscosity	22
2.5	Element-wise Artificial Viscosity (AV) coefficients for the inviscid $\text{Ma}=1.6$ case	23
2.6	AV coefficients with continuity enforcement	23
2.7	Detail of the flat-plate leading edge ($x=0.0$, mesh a2).	29
2.8	Flow solution at the end of the flat-plate ($x=1.0$, mesh a2).	29
2.9	Convergence comparison (3^{rd} order, finest grids).	30
2.10	Comparison of HiFiLES with SU2 using a similar time integration scheme.	30
2.11	The mesh for the circular cylinder simulations along with x-velocity contours for the $Re = 20$ case.	32
2.12	Pressure contours for the steady and unsteady (instantaneous) cylinder cases.	32
2.13	Instantaneous solution contours for the unsteady cylinder case.	33

2.14 Density and vorticity contours for the flow with $\text{Re} = 10,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements	35
2.15 Density and vorticity contours for the flow with $\text{Re} = 22,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements	35
2.16 Density and vorticity contours for the flow with $\text{Re} = 60,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements	36
2.17 Density and vorticity isosurfaces colored by Mach number for the flow with $Re = 10,000$ around the SD7003 wing-section. $p = 3$ on unstructured tetrahedral grid with $N = 711,332$ elements	37
2.18 Taylor-Green vortex results on hexahedral and tetrahedral meshes from Bull and Jameson [11]. (a, b) Evolution of average kinetic energy $\langle k \rangle$; (c, d) dissipation rate $-d\langle k \rangle/dt$. ‘SD- $M \times N$ ’ refers to M^3 mesh, N th-order accurate SD scheme. (---) 4th-order Discontinuous-Galerkin (DG) on 64^3 mesh [7]; (○) Direct Numerical Simulation (DNS) [26].	39
2.19 Taylor-Green Vortex (TGV) solution on the fine mesh using fourth order accurate DG method, showing isosurfaces of q criterion colored by velocity magnitude at time $t = 2.5$ to 10.75 seconds.	40
2.20 Square cylinder geometry and tetrahedral boundary layer mesh showing all degrees of freedom	42
2.21 Isosurface of the q -criterion colored by velocity magnitude showing the wake behind the square cylinder	42
2.22 (a) Mean streamwise and vertical velocity and mean Reynolds stresses along vertical lines in the wake. (—) current results, (---) 4th order Spectral Difference (SD)+WALE-Similarity (WSM) on hexahedral mesh by Lodato and Jameson [48], (○) Laser Doppler Velocimetry (LDV) experiments by Lyn et al. [56, 55].	43

2.23	Turbulent flow past a NACA 0012 airfoil at $\text{Re} = 6$ million, $\text{Ma} = 0.15$, $\alpha = 0^\circ$ using FR to recover 4th order accurate DG method and the SA turbulence model.	44
2.24	Pressure coefficient on the NACA 0012 airfoil at $\text{Re} = 6$ million, $\text{Ma} = 0.15$, $\alpha = 0^\circ$ using FR to recover 4th order accurate DG method and the SA turbulence model.	45
3.1	Left and right correction functions for the C1FR scheme with $P = 3$, in which the zeroth and first derivatives of the corrected flux are continuous	69
3.2	Left and right correction functions for the C2FR scheme with $P = 3$, in which the zeroth, first, and second derivatives of the corrected flux are continuous	70
3.3	L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 1$. Order of accuracy in solution: DG: 2.728, C^1 Flux Reconstruction (C1FR): 3.374. Order of accuracy in first derivative: DG: 2.691, C1FR: 3.359.	72
3.4	L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 2$. Order of accuracy in solution: DG: 4.960, C1FR: 3.917 . Order of accuracy in first derivative: DG: 4.971, C1FR: 4.178.	73
3.5	L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 3$. Order of accuracy in solution: DG: 7.119, C1FR: 7.187. Order of accuracy in first derivative: DG: 7.908, C1FR: 7.882.	74
3.6	Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with low wavenumber: $k = 0.25(P + 1)\pi/h$, $P = 3$, $h = 1$	75

3.7	Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with medium wavenumber: $k = 0.5(P + 1)\pi/h$, $P = 3$, $h = 1$	75
3.8	Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with high wavenumber: $k = 0.75(P + 1)\pi/h$, $P = 3$, $h = 1$	76
3.9	Sod's Shock Tube Problem with "thickened" discontinuity at $t = 0$. . .	79
3.10	Sod's Shock Tube Problem at $t = 0.25$ solved with C1FR. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the C1FR scheme and "thickened" discontinuity in the initial conditions shown in Figure 3.9. $N = 71$, $P = 3$, $c_1 = 1e - 2$, $\alpha_0 = 1$, $\alpha_1 = 0$, Courant – Friedrichs – Lewy(CFL) = $2.5e - 2$	80
3.11	Sod's Shock Tube Problem at $t = 0.25$ solved with regular DG. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the DG scheme and "thickened" discontinuity in the initial conditions shown in Figure 3.9. $N = 70$, $P = 3$, $\alpha_0 = 0$	81
3.12	123 Problem with "thickened" discontinuity at $t = 0$ solved with C1FR. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the C1FR scheme and "thickened" discontinuity in the initial conditions shown in Figure 3.12. $N = 71$, $P = 3$, $c_1 = 1e - 2$, $\alpha_0 = 1$, $\alpha_1 = 0$, CFL = $2.5e - 2$	83
3.13	123 problem at $t = 0.1$. Solid black line is the exact solution. Superimposed solid red line is the solution obtained with the C1FR scheme.	84

4.1	Using the same filter width in two different domains introduces different bias. The small, solid circles represent the location of solution points. The large circles represent the filter width acting on two specific internal points in two different domains. Integration is being performed in the area encompassed by the solid, straight lines.	94
4.2	Sketch of ellipsis $\ \xi - \xi_0\ _{rt} = 1$ defined in the right triangle domain maps to a circle in the equilateral triangle domain	95
4.3	Internal and boundary points in the reference triangular domain when solution is represented with polynomial of degree 5 ($p = 5$). Black circles represent the internal points. Red squares represent the boundary points.	99
4.4	Solution values after filtering with internal component exclusively, $\vec{v} = \mathcal{T}\vec{v}$, where $v_i = \sin(k\xi_{i1}) + \cos(k\xi_{i2})$, where $k = 500$. Hollow black circles show the unfiltered solution at the interior points, transparent colored surface is the polynomial representation of the unfiltered solution, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the filtered solution.	100
4.5	Solution values after filtering with boundary component exclusively, $\vec{v} = \mathcal{B}\vec{v}^*$, where $\vec{v}^* = a\vec{\xi}_1^* + b\vec{\xi}_2^* + c$ for some constants a, b, c . Red squares show the values of the solution at the boundaries, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the solution.	101
4.6	Solution values after filtering with boundary component exclusively, $\vec{v} = \mathcal{B}\vec{v}^*$, where $v_i^* = \sin(k\xi_{i1}^*) + \cos(k\xi_{i2}^*)$, $k = 500$. Red squares show the values of the solution at the boundaries, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the solution.	102

4.7	Solution values after filtering with both internal and boundary components, $\vec{v} = \alpha\mathcal{T}\vec{v} + (1-\alpha)\mathcal{B}\vec{v}^*$, where $\alpha = 0.8$, $v_i = \sin(k\xi_{i1}) + \cos(k\xi_{i2})$, $k = 500$, $\vec{v}^* = \beta(-\xi_1^* + \xi_2^*)$. Hollow black circles show the unfiltered solution at the interior points, transparent colored surface is the polynomial representation of the unfiltered solution, filled black circles show the filtered values of the solution at the interior points, hollow red squares show the value of the solution at the boundary points, and the meshed surface shows the polynomial representation of the filtered solution.	103
4.8	Unstructured, coarse mesh of a circular cylinder with 714 triangular elements. Elements adjacent to the cylinder have quadratic edges.	107
4.9	Flow past a cylinder. $Re = 1e6$, $Ma = 0.2$, $p = 4$	108
4.10	Flow past a cylinder. $Re = 1e6$, $Ma = 0.077$, $p = 4$	109
4.11	Flow past a cylinder. $Re = 1e6$, $Ma = 0.87$, $p = 4$	110
4.12	History of C_D and energy residual of simulation in Figure 4.11	111
4.13	Unstructured, coarse mesh of a circular cylinder with 5,616 triangular elements. Elements adjacent to the cylinder have quadratic edges.	112
4.14	Flow past a cylinder. $Re = 1e6$, $Ma = 0.0077$, $p = 4$	112
4.15	Mesh used in case A in Table 4.3. Contains 131,635 triangular elements with second order edges.	115
4.16	Mesh used in cases B, D, and G in Table 4.3. Contains 23,313 triangular elements with second order edges.	116
4.17	Mesh used in case C in Table 4.3. Contains 11,721 triangular elements with second order edges.	116
4.18	Mesh used in cases E and F in Table 4.3. Contains 16,219 triangular elements with second order edges.	117
4.19	Case A in Table 4.3 lift coefficient, Strouhal number (St) power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.15 with $P = 1$	118
4.20	Case B in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.16 with $P = 4$	119

4.21 Case C in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.17 with $P = 4$.	120
4.22 Case D in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.16 with $P = 4$.	121
4.23 Case F in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.18 with $P = 5$.	122
4.24 Case G in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.16 with $P = 5$.	123

Chapter 1

Introduction

1.1 The Role of High-Order Methods in the Future of Industrial Simulations

Over the last 20 years, much fundamental work has been done in developing high-order numerical methods for Computational Fluid Dynamics. Moreover, the need to improve and simplify these methods has attracted the interest of the applied mathematics and the engineering communities. Now, these methods are beginning to prove themselves sufficiently robust, accurate, and efficient for use in real-world applications.

However, low-order numerical methods are still the standard in the aeronautical industry. There has been a sustained scientific and economical investment to develop this successful and robust technology for a long time. Currently, an industry-standard, second-order finite volume computational tool performs adequately well in a broad range of aeronautical engineering applications. For that reason, the introduction of new, high-order numerical schemes in the aeronautical industry is challenging, particularly in areas where the low-order numerical methods already provide the required robustness and accuracy (keeping in mind the limitations of current turbulence model technology).

Thanks to new and emerging aircraft roles (very small or large concepts, very

high or low altitude, quiet vehicles, low fuel consumption vehicles, etc.), revolutionary aircraft design concepts will appear in the near future, and the need for high-fidelity simulation techniques to predict their performance is growing rapidly. Undoubtedly, high-order numerical methods are starting to find their place in the aeronautical industry.

Unsteady simulations, flapping wings, wake capturing, noise prediction, and LES are just a few examples of computations that could benefit from high-order numerical methods. In particular, high-order methods have a significant edge in applications that require accurate resolution of the smallest scales of the flow. Such situations include the generation and propagation of acoustic noise from an airframe, or at the limits of the flight envelope where unsteady, vortex-dominated flows have a significant effect on aircraft performance. Utilizing a high-order representation enables smaller scales to be resolved with a greater degree of accuracy than standard second-order methods. Furthermore, high-order methods are inherently less dissipative, resulting in less unwanted interference with the correct development of the turbulent energy cascade. This factor makes the combination of high-order numerics with LES modeling very powerful, with the potential to significantly improve upon the accuracy and computational cost of the standard approach of LES with second-order methods. The amount of computing effort to achieve a small error tolerance can also be much smaller with high-order than second-order methods. Even real time simulations (one second of computational time, one second of real flight), could benefit from high-order algorithms that feature more intensive computation within each mesh element (ideal for vector machines and new computational platforms like GPUs, FPGAs, coprocessors, etc).

However, before claiming the future success of high-order numerical methods in industry, two main difficulties should be overcome: a) high-order numerical schemes must be as robust as state-of-the-art low-order numerical methods, b) the existing level of verification and validation in high-order CFD codes should be similar to the typical level of their low-order counterparts.

1.2 Motivation

The SD++ code originally designed and developed by Peter Vincent, Patrice Castonguay, and David Williams [16] marked the culmination of multiple years of research at the ACL. It solved the Navier-Stokes and Euler equations in general unstructured grids in 2D and 3D. The original creators demonstrated its capabilities and excellent scalability in GPUs. Its stability in linear problems was guaranteed for triangular [96] and tetrahedral [97] elements with constant Jacobians, and its non-linear stability properties were understood to a practical extent [40]: as long as the exact flux of a hyperbolic equation being solved with the FR scheme is not projected *exactly* onto the polynomial space of the flux, aliasing instabilities will *invariably* arise.

At the beginning of my graduate studies, I was extremely impressed by the capabilities of SD++, its clean code base, the high performance it achieved, and its general applicability beyond Aerospace applications. The members of the ACL agreed that a code with such capabilities could not be allowed to wither into oblivion. We set ourselves the goal, inspired by the success of fellow graduate students in the SU2 [65] team and guided by their lead developer – Francisco Palacios–, to bring the capabilities of SD++ to a level where industrial applications (complex geometries, imperfect grids, high Re, high Ma) are feasible.

We concluded that releasing the code open source would allow us to more easily identify what capabilities are most useful and speed up development with the help of outside researchers. After adding additional LES Sub-Grid Scale (SGS) models, local time-stepping, and artificial dissipation for shock-capturing, the ACL released the developer’s edition of the code HiFiLES on GitHub at github.com/HiFiLES/HiFiLES-solver with an official presence in the Stanford University servers at hifiles.stanford.edu.

Prof. Jameson’s conclusion regarding aliasing in the solution of non-linear equations with FR limited HiFiLES’s applicability in the following ways:

- High gradients in the solution (shocks, fast moving fluid over a boundary) would lead to instabilities

- Only medium to low Re flows could be simulated
- Even in smooth problems, coarse meshing of areas far away from the regions of interest could lead to an unpredictable halt in the computations
- Sharp corners in the geometry could stop the simulation
- The time-step in mildly non-linear problems needed to be lower than expected to deal with the artificial stiffness introduced by aliased solutions

The thrill of getting a code as powerful and modern as HiFiLES to produce an answer in all possible scenarios, very much how ANSYS Fluent and CFX can, was a strong drive to tackle the issues of stability.

The first idea to tackle de-stabilizing aliasing came while perusing the derivation of the Energy Stable FR (ESFR) schemes [85] and the search for their optimal dispersion and dissipation properties [4]. In Vincent's derivation, the correction functions are parameterized with a constant and are created so the corrected fluxes are globally continuous. The proof would work similarly if the correction functions were globally continuous in arbitrarily many derivatives, and thus had arbitrarily many parameterizing constants. This observation lead to the development of the CMFR schemes [52]. Suddenly, a linearly provably-stable numerical scheme could have its dispersion and dissipation properties tuned as freely as necessary.

Approaching the stability issues from the numerics had the potential to yield good results, but a more immediate approach was needed to stabilize HiFiLES in a general way. It was not clear CMFR schemes could be readily applied to triangles and tetrahedral elements. Explicit filtering, as performed by Visbal et al. [87], seemed like an attractive prospect. White and Visbal use high-order Padé differentiation and low-pass spatial filtering procedures in implicit LES to solve extremely complex flow scenarios like turbulence-shock interactions [93]. The creation of the LFS filters [3] and their extension to general, multi-dimensional elements [53] are the first attempt to bring the capabilities developed by Visbal to high-order codes for unstructured grids.

1.3 Contributions

The vision of bringing a powerful simulation tool like HiFiLES to the engineer’s tool-box prompted me to focus on the areas that are essential in any such tool: validation and verification efforts, support to the open source community of researchers who want to use HiFiLES, re-assessment of and contribution to the stability properties of FR –HiFiLES’s underlying numerical method–, and enabling HiFiLES to provide a result even in the most extreme of simulation cases.

1.3.1 Maintenance and Validation and Verification of Hi-FiLES

After exposure to SD++, HiFiLES’s starting point, by creating its sparse matrix-vector multiplication routine [16] in Graphical Processing Unit (GPU)s, it became clear that anyone who has just started working on the source code needs guidance and assurances about its correctness. The Validation and Verification (V&V) efforts undertaken by the ACL [51] provided the community assurances about its correctness. The crucial task of documenting HiFiLES, creating tutorials, providing direct support to researchers, and eliminating bugs that are found by collaborators is an ongoing task, as can be seen in HiFiLES’s repository: github.com/HiFiLES/HiFiLES-solver.

1.3.2 Tackling Stability via Numerics: Creation of Provably Linearly Stable CMFR Schemes

When confronting the problem of the high propensity of HiFiLES to Nan when a single element was not grided properly, or the flow non-linearities were even slightly too high for the grid, going back to the basics yielded interesting results. The CMFR schemes [52] are provably stable for linear problems, very much the same way FR is, but provide a range of selections of numerical schemes with varying dissipation and dispersion properties.

The academic endeavor of discovering these schemes provided valuable insights

into the behavior and potential extensions of high-order methods. However, developing CMFR schemes further would have detracted focus from the end goal of making HiFiLES a robust tool.

1.3.3 Tackling Stability via Filtering: Creation of Generalized LFS Filters

It was encouraging to see the results obtained by Asthana et. al [3] regarding stabilization of extreme high order computations in 1D and 2D tensor product elements using LFS. The idea of filtering via truncated, element-wise convolutions of the solution with a kernel seemed to be too simple to work. Absolute accuracy in non-linear problems did increase with higher order approximations.

The extension to simplex 2D and 3D elements seemed straightforward. The extreme stabilizing properties of the filters were shown in [53]. The main difference between Asthana's formulation of the LFS filters and the LFS filters presented in this dissertation is that I allow the basis functions outside of the element being filtered to have any form (in fact, their form need not be defined), Asthana requires the basis functions to extend to infinity when computing the convolution.

My work regarding this form of stabilization has brought HiFiLES closer to performing reliably in realistic industrial scenarios: complex, unstructured geometries with under-resolved grids being used to solve highly non-linear (high-Re, high-Ma) flows.

Chapter 2

HiFiLES: an open source GPU-powered, High-Order Large-Eddy Simulation code

2.1 Introduction

During the last decade, the ACL of the Department of Aeronautics and Astronautics at Stanford University has developed a series of high-order numerical schemes and computational tools that have demonstrated the viability of this technique. In this chapter, the code named HiFiLES, developed in the ACL and built on top of SD++ (Castonguay et al.[16]), is described in detail with a particular emphasis on robustness in a range of applications and V&V. HiFiLES takes advantage of the synergies between applied mathematics, aerospace engineering, and computer science in order to achieve the ultimate goal of developing an advanced high-fidelity simulation environment.

In addition to the original characteristics of the SD++ code, HiFiLES includes some important physical models and computational methods such as: LES using explicit filters and advanced SGS models, high-order stabilization techniques, shock detection and capturing for compressible flow calculations, and local time stepping.

During the development of this software, several key decisions have been taken to guarantee a flexible and lasting infrastructure for industrial Computational Fluid

Dynamics simulations:

- The selection of the ESFR scheme on unstructured grids[85]. The flexibility of this method has been critical to guarantee a correct solution independently of the particular physical characteristics of the problem.
- High performance, materialized in a multi-GPU implementation that takes advantage of the ease of parallelization afforded by discontinuous solution representation. Furthermore, HiFiLES aims to guarantee compatibility with future vector machines and revolutionary hardware technologies.
- Code portability by using ANSI C++ and relying on widely-available, and well-supported mathematical libraries like Blas, LAPACK, CuBLAS, and ParMetis.
- Object oriented structure to boost the re-usability and encapsulation of the code. This abstraction enables modifications without incorrectly affecting other portions of the code. Although some level of performance is traded for re-usability and encapsulation, the loss in performance is minor.

As the mathematical basis and computational implementation of HiFiLES have been described in previous work [16], the goal of this chapter is to illustrate the level of robustness of HiFiLES for interesting problems. This will be accomplished via a V&V study, which is fundamental for increasing the credibility of this technology in a competitive industrial framework.

In particular, to ensure that the implementation of the aforementioned features in HiFiLES is correct, the following verification tests are shown: checks of spatial and temporal order of accuracy using the Method of Manufactured Solutions (MMS) in 2D and 3D for viscous and inviscid flows and characterization of stable time-step limits. After the Verification, a detailed Validation of the code is presented to illustrate that the solutions provided by HiFiLES are an accurate representation of the real world. Simulations of complex flows are validated against experimental or DNS results for the following cases: laminar flat-plane, flow around a circular cylinder, SD7003 wing-section and airfoil at 4° angle of attack, the Taylor-Green Vortex, and LES of a square cylinder.

The organization of this chapter is as follows. Section 2.2 provides a description of the governing equations. Section 2.3 describes the mathematical and numerical algorithms implemented in the code. Section 2.4 focuses on the V&V of HiFiLES, and the conclusions are summarized in Section 2.5. It is important to highlight that the contents of this chapter mirror the American Institute of Aeronautics and Astronautics (AIAA) paper[51] created jointly with members of the ACL and Francisco Palacios.

2.2 Governing Equations

2.2.1 Navier Stokes equations

The Navier-Stokes (NS) [45] equations provide a complete (dynamical) description of a viscous fluid and expresses the conservation of mass, momentum and energy. The complete system of equations (without source terms and assuming adiabatic boundary conditions at the solid wall) can be written in the following conservative form:

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0 \quad (2.1)$$

where $\mathbf{F} = (F, G, H) = (F_I, G_I, H_I) - (F_V, G_V, H_V)$ and

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix} \quad (2.2)$$

$$F_I = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho u e + p u \end{pmatrix} \quad G_I = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v w \\ \rho v e + p v \end{pmatrix} \quad H_I = \begin{pmatrix} \rho w \\ \rho w u \\ \rho w v \\ \rho w^2 + p \\ \rho w e + p w \end{pmatrix} \quad (2.3)$$

$$F_V = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \\ u_i \sigma_{ix} - q_x \end{pmatrix} \quad G_V = \begin{pmatrix} 0 \\ \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \\ u_i \sigma_{iy} - q_y \end{pmatrix} \quad H_V = \begin{pmatrix} 0 \\ \sigma_{zx} \\ \sigma_{zy} \\ \sigma_{zz} \\ u_i \sigma_{iz} - q_z \end{pmatrix} \quad (2.4)$$

As usual, ρ is density, u , v , w are the velocity components in the x , y , z directions, respectively, and e is total energy per unit mass. In HiFiLES, the pressure is determined from the ideal gas equation of state

$$p = (\gamma - 1)\rho \left(e - \frac{1}{2} (u^2 + v^2 + w^2) \right) \quad (2.5)$$

the viscous stresses are those of a Newtonian fluid

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (2.6)$$

and the heat fluxes are defined as

$$q_i = -k \frac{\partial T}{\partial x_i} \quad (2.7)$$

where

$$k = \frac{C_p \mu}{\text{Pr}}, \quad T = \frac{p}{R\rho} \quad (2.8)$$

C_p is the specific heat at constant pressure and R is the specific gas constant. In the case of air, $\gamma = 1.4$ and Prandtl number (Pr) = 0.72. The dynamic viscosity μ in HiFiLES can be a constant or a function of temperature using Sutherland's law.

2.2.2 Reynolds Averaged Navier-Stokes equations

The compressible NS equations can be used to solve a variety of different flow physics problems but for turbulent flows, direct numerical simulation using these equations

can become excessively expensive. For engineering applications, it is customary to perform a Favre averaging procedure to the NS equations to solve a turbulent mean quantity. This leads to a variety of terms which must be modeled in order to provide closure to the resulting RANS equations [94, 63]. For example, using the one equation SA turbulence model, the conservative form of the RANS equations is very similar to the NS equations with the following extra terms included in Eqn. 2.1:

$$U_{\tilde{\nu}} = \rho \tilde{\nu}, \quad F_{I,\tilde{\nu}} = \rho u \tilde{\nu}, \quad G_{I,\tilde{\nu}} = \rho v \tilde{\nu}, \quad H_{I,\tilde{\nu}} = \rho w \tilde{\nu}, \quad (2.9)$$

$$F_{V,\tilde{\nu}} = \frac{1}{\sigma}(\mu + \mu\psi) \frac{\partial \tilde{\nu}}{\partial x}, \quad G_{V,\tilde{\nu}} = \frac{1}{\sigma}(\mu + \mu\psi) \frac{\partial \tilde{\nu}}{\partial y}, \quad H_{V,\tilde{\nu}} = \frac{1}{\sigma}(\mu + \mu\psi) \frac{\partial \tilde{\nu}}{\partial w}, \quad (2.10)$$

$$S_{\tilde{\nu}} = c_{b_1} \tilde{S} \rho \nu \psi + \frac{1}{\sigma} [c_{b_2} \rho \nabla \tilde{\nu} \cdot \nabla \tilde{\nu}] - c_{w_1} \rho f_w \left(\frac{\nu \psi}{d} \right)^2. \quad (2.11)$$

Note that the flow variables have been redefined as Favre-averaged quantities. Also, the viscous stresses (Eqn. 2.6) now include the Boussinesq approximated Reynolds stress terms,

$$\sigma_{ij} = (\mu + \mu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}(\mu + \mu_t) \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad (2.12)$$

and the heat fluxes are redefined as

$$q_i = -C_p \left(\frac{\mu}{\text{Pr}} + \frac{\mu_t}{\text{Pr}_t} \right) \frac{\partial T}{\partial x_i} \quad (2.13)$$

where μ_t is the dynamic eddy viscosity and Pr_t is the turbulent Prandtl number. The various terms added by the one equation SA turbulence model are defined in a later section.

2.3 Numerical Methods

In this section the main numerical techniques implemented in HiFiLES will be described. We will emphasize the critical role of the selected numerical discretization

(FR Method), and its capability to solve Computational Fluid Dynamics (CFD) problems using unstructured meshes.

2.3.1 FR Method

What follows is an overview of the FR framework. We start the discussion with the solution of the advection equation in one dimension using the FR approach to illustrate the method and how it can be cast as a differential operator. Then we show how it would be possible to use FR to discretize spatial derivatives of arbitrary order. We then proceed to describe which common schemes can be recovered via FR and under which norm they can be proven to be stable. Then we explain how conservation equations can be solved in multiple dimensions. The NS equations are a set of coupled conservation equations in multiple dimensions, so the extension of the FR methodology to them uses the concepts explained here. The detailed description of the algorithm used in HiFiLES is given by Castonguay et al. [16].

2.3.1.1 Solution of the General Advection Equation in One Dimension using the FR Approach

The FR approach is a discretization of the 1st spatial derivative operator. The general advection equation is a good starting point to describe the mechanics of the scheme.

Consider the one-dimensional conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (2.14)$$

in domain Ω , where x is the spatial coordinate, t is time, u –the *solution*– is a scalar function of x and t , and f –the *flux*– is a scalar function of u . Note that by letting $f = f(u, \frac{\partial u}{\partial x})$, Equation 2.14 becomes a model of the NS equations.

Let us partition the domain $\Omega = [x_1, x_{N+1}]$ into N non-overlapping elements with interfaces at $x_1 < x_2 < \dots < x_{N+1}$. Then,

$$\Omega = \bigcup_{n=1}^N \Omega_n \quad (2.15)$$

and $\Omega_n = [x_n, x_{n+1})$ for $n = 1, \dots, N$. To simplify the implementation, let us map each of the physical elements Ω_n to a standard element $\Omega_s = [-1, 1)$ with the function $\Theta_n(\xi)$, where

$$x = \Theta_n(\xi) = \left(\frac{1 - \xi}{2} \right) x_n + \left(\frac{1 + \xi}{2} \right) x_{n+1} \quad (2.16)$$

With this mapping, the evolution of u within each Ω_n can be determined with the following transformed conservation equation

$$\frac{\partial \hat{u}}{\partial t} + \frac{1}{J_n} \frac{\partial \hat{f}}{\partial \xi} = 0 \quad (2.17)$$

where

$$\hat{u} = u(\Theta_n(\xi), t) \text{ in } \Omega_n \quad (2.18)$$

$$\hat{f} = f(\Theta_n(\xi), t) \text{ in } \Omega_n \quad (2.19)$$

$$J_n = \left. \frac{\partial x}{\partial \xi} \right|_{\Omega_n} \quad (2.20)$$

Now, we introduce polynomials of degree p , \hat{u}^δ and \hat{f}^δ , to approximate the exact values \hat{u} , \hat{f} , respectively. We can write these polynomials as

$$\hat{u}^\delta = \sum_{i=1}^{N_s} \hat{u}_i^\delta l_i(\xi) \quad (2.21)$$

$$\hat{f}^\delta = \sum_{i=1}^{N_s} \hat{f}_i^\delta l_i(\xi) \quad (2.22)$$

where N_s is the number of solution points, \hat{u}_i^δ is the current value of the solution approximation function at the i^{th} *solution point* in the reference element, \hat{f}_i^δ is the current value of the flux approximation function at the i^{th} *flux point* in the reference element, l_i is the Lagrange polynomial equal to 1 at the i^{th} solution point and 0 at the others, and δ denotes that the function is an approximation.

Note that the piecewise polynomials might not be continuous (or C^0) across the

interfaces. In the FR approach, the flux used in the time advancement of the solution is made C^0 by introducing flux correction functions.

This can be achieved by finding interface solution values at each element boundary and then correcting the solution. Let $\hat{f}_L^{\delta I}$ and $\hat{f}_R^{\delta I}$ be the interface solution values at left and right boundaries of some element, respectively. $\hat{f}_L^{\delta I}$ and $\hat{f}_R^{\delta I}$ can be found with a Riemann solver for DG methods[34]. Then, select solution correction functions g_L and g_R such that

$$g_L(-1) = 1, \quad g_L(1) = 0 \quad (2.23)$$

$$g_R(-1) = 0, \quad g_R(1) = 1 \quad (2.24)$$

and let

$$\hat{f}^C = \hat{f}^\delta + (\hat{f}_L^{\delta I} - \hat{f}_L^\delta)g_L + (\hat{f}_R^{\delta I} - \hat{f}_R^\delta)g_R \quad (2.25)$$

where superscript C denotes the function is corrected, and $\hat{f}_L^\delta, \hat{f}_R^\delta$ represent the solution approximation evaluated at the left and right boundaries.

As a result, the FR spatial differential operator in element n can be written as

$$\begin{aligned} \frac{\partial f(x)}{\partial x} \Big|_{\Omega_n} &\stackrel{FR}{=} \frac{1}{J_n} \frac{\partial \hat{f}^C(\xi)}{\partial \xi} \\ &= \frac{1}{J_n} \left(\sum_{i=1}^{N_s} \hat{f}_i^\delta \frac{\partial l_i(\xi)}{\partial \xi} + (\hat{f}_L^{\delta I} - \hat{f}_L^\delta) \frac{\partial g_L(\xi)}{\partial \xi} + (\hat{f}_R^{\delta I} - \hat{f}_R^\delta) \frac{\partial g_R(\xi)}{\partial \xi} \right) \end{aligned} \quad (2.26)$$

The actual values of the corrected quantity \hat{f}^C are never used, only its first spatial derivative.

The solution can then be advanced at each solution point i . In semi-discrete form, this is

$$\frac{d\hat{u}_i^\delta}{dt} = -\frac{1}{J_n} \frac{\partial \hat{f}^C}{\partial \xi}(\xi_i) \quad (2.27)$$

2.3.1.2 Extension to Higher Order Spatial Derivatives

FR can be used to discretize spatial differential operators of any order via composition. For example, the second derivative spatial differential in element n can be discretized as

$$\begin{aligned} \frac{\partial^2 *}{\partial x^2} \Big|_{\Omega_n} &= \frac{\partial}{\partial x} \left(\frac{\partial *}{\partial x} \right) \Big|_{\Omega_n} \\ &\stackrel{FR}{=} \frac{1}{J_n} \frac{\partial}{\partial \xi} \left(\frac{1}{J_n} \frac{\partial *^C}{\partial \xi} \right)^C \end{aligned} \quad (2.28)$$

Each differential operator requires the correction of the operand. Then, in the case of the second derivative operator, $*$ is corrected once using its values at each element boundary point, the common $*$ values at each boundary point, and its values at each internal point. This provides the values of $\frac{\partial *^C}{\partial \xi}$ at each internal point.

Let $q = \frac{1}{J_n} \frac{\partial *^C}{\partial \xi}$. We can find the values of $\frac{\partial q^C}{\partial \xi}$ using the same procedure we used to find $\frac{\partial *^C}{\partial \xi}$.

As a result, the m^{th} FR spatial derivative operator will require m corrections.

2.3.1.3 Energy Stability of FR in the Linear Advection-Diffusion Equation

The FR scheme can be made provably stable for the linear advection-diffusion equation by selecting special types of correction functions [15]. In general, these correction functions are polynomials of degree $p + 1$ so both sides in Equation (2.27) are quantities related to polynomials of order p –for consistency [36].

Vincent et al. [85] have shown that in the case of the 1-dimensional, linear advection equation, the FR approach can be proven to be stable for a specific family of correction functions parameterized by a scalar called c . This parameter arises from the desire to ensure the following Sobolev-type norm is bounded above by zero

$$\|u\|_2^2 = \sum_{n=1}^N \int_{x_n}^{x_{n+1}} \left\{ u^2 + \frac{c}{2} \left(\frac{\partial^p u}{\partial x^p} \right)^2 \right\} dx \quad (2.29)$$

In the case of pure linear advection, they showed that by selecting specific values

of c it is possible to recover a particular nodal DG and SD methods plus a FR scheme that was previously found to be stable by Huynh [36]. The family of schemes that are provably stable in the linear advection case are named ESFR. The coefficients that give rise to the correction functions that recover DG and SD schemes in the linear advection case are labeled c_{DG} and c_{SD} , respectively. In addition, there is an ESFR scheme that maximizes the CFL condition. This scheme's correction functions arise from selecting $c = c_+$.

Similarly, the families of schemes that are stable in the linear advection-diffusion equation have an additional parameter, which Castonguay et al. [15] labeled κ . The schemes arising from κ_{DG} and κ_{SD} recover the behavior of DG and SD, respectively, in the linear advection-diffusion equation. The scheme arising from κ_+ provides the largest CFL condition.

It is important to keep in mind that the FR schemes that recover other schemes in the linear equations can be used in non-linear equations, but the FR schemes' non-linear properties will likely be different from those of the schemes they recover in the linear case.

2.3.1.4 Extension to Multiple Dimensions

Extension of FR to multiple dimensions requires formulating multi-dimensional interpolation functions and correction functions that satisfy boundary conditions equivalent to those in Eqn. (2.23) for each type of element.

Interpolation bases for quadrilaterals and hexahedra can be obtained via tensor products of the 1-dimensional interpolation basis. In HiFiLES, the solution in hexahedra is discretized in the following way

$$\hat{u}^\delta(\xi, \eta, \zeta) = \sum_{i=1}^{p+1} \sum_{j=1}^{p+1} \sum_{k=1}^{p+1} \hat{u}_{i,j,k}^\delta l_i(\xi) l_j(\eta) l_k(\zeta), \quad (2.30)$$

where i, j, k index the solution points along the ξ, η, ζ directions, respectively. The flux is discretized similarly.

The interpolation basis for triangles and tetrahedra are described in detail by Hesthaven and Warburton [34]. Figure 4.3 shows a possible configuration of internal and boundary points. The extension of interpolation polynomials to prisms is obtained via tensor products of the 1-dimensional basis with the triangular basis [16].

The most general polynomial discretization of a k -dimensional solution scalar field and flux vector field in an arbitrary reference element is

$$\hat{u}^\delta(\boldsymbol{\xi}) = \sum_{i=1}^{N_s} \hat{u}_i^\delta \phi_i(\boldsymbol{\xi}), \quad (2.31)$$

$$\hat{\mathbf{f}}^\delta(\boldsymbol{\xi}) = \sum_{i=1}^{N_s} \hat{\mathbf{f}}_i^\delta \phi_i(\boldsymbol{\xi}), \quad (2.32)$$

where N_s is the number of solution points in an element, $\phi_i(\boldsymbol{\xi})$ is a polynomial basis function associated with solution point i constructed such that $\phi_i(\boldsymbol{\xi}_j) = \delta_{ij}$ and $i, j = 1, \dots, N_s$.

By letting $\vec{u} = \langle \hat{u}_1^\delta, \hat{u}_2^\delta, \dots, \hat{u}_{N_s}^\delta \rangle^T$, $\vec{\mathbf{f}} = \langle \hat{\mathbf{f}}_1^\delta, \hat{\mathbf{f}}_2^\delta, \dots, \hat{\mathbf{f}}_{N_s}^\delta \rangle^T$, and $\vec{\phi} = \langle \phi_1, \phi_2, \dots, \phi_{N_s} \rangle^T$ the discretization can be written more concisely as

$$\begin{aligned} \hat{u}^\delta(\boldsymbol{\xi}) &= \vec{u}^T \cdot \vec{\phi}(\boldsymbol{\xi}) = \vec{\phi}(\boldsymbol{\xi})^T \cdot \vec{u}, \\ \hat{\mathbf{f}}^\delta(\boldsymbol{\xi}) &= \vec{\mathbf{f}}^T \cdot \vec{\phi}(\boldsymbol{\xi}) = \vec{\phi}(\boldsymbol{\xi})^T \cdot \vec{\mathbf{f}}. \end{aligned} \quad (2.33)$$

Note that we are using the boldface and arrow notation to denote vectors. Boldface vectors have a number of entries equal to the number of dimensions of the problem domain. Arrow vectors are vectors of general dimensions.

In the general FR approach, the boundary conditions for the correction functions in multiple dimensions can be formulated as

$$\mathbf{h}_i(\boldsymbol{\xi}_j) \cdot \mathbf{n}_j = \delta_{ij}, \quad (2.34)$$

where \mathbf{h}_i is the correction vector function associated with interface point i , $\boldsymbol{\xi}_j$ is the location vector of the j^{th} interface point, and \mathbf{n}_j is the outward unit normal at interface point j . Interface (or boundary) points are located on the boundary of an

element.

One of the challenges in the FR approach is finding correction functions that not only satisfy Equation (2.34) but also guarantee stability in the linear advection-diffusion case. Correction functions that guarantee such stability exist for 1-dimensional segments[85], triangles[18, 96], and tetrahedra[97]. FR schemes with these correction functions comprise the ESFR family of schemes.

The update step at solution point i and element n in the FR approach for the multidimensional advection equation $\frac{\partial u}{\partial t} + \nabla \cdot \vec{f} = 0$ becomes

$$\begin{aligned} \frac{du_i^\delta}{dt} &= -\frac{1}{\det(\tilde{J}_n)} \nabla \cdot \mathbf{f}_i^C(\boldsymbol{\xi}_i) = -\frac{1}{\det(\tilde{J}_n)} \nabla \cdot \left(\mathbf{f}_i^\delta(\boldsymbol{\xi}_i) + \sum_{j=1}^{N_f} \left[(\hat{\mathbf{f}}_j^{\delta I} - \hat{\mathbf{f}}_j^b) \cdot \mathbf{n}_j \right] \vec{\mathbf{h}}_j(\boldsymbol{\xi}_i) \right) \\ &= -\frac{1}{\det(\tilde{J}_n)} \left(\nabla \cdot \mathbf{f}_i^\delta(\boldsymbol{\xi}_i) + \sum_{j=1}^{N_f} \left[(\hat{\mathbf{f}}_j^{\delta I} - \hat{\mathbf{f}}_j^b) \cdot \mathbf{n}_j \right] \nabla \cdot \vec{\mathbf{h}}_j(\boldsymbol{\xi}_i) \right) \\ &= -\frac{1}{\det(\tilde{J}_n)} \left(\vec{\nabla} \phi(\boldsymbol{\xi}_i)^T \cdot \vec{\mathbf{f}} + \sum_{j=1}^{N_f} \left[(\hat{\mathbf{f}}_j^{\delta I} - \hat{\mathbf{f}}_j^b) \cdot \mathbf{n}_j \right] \nabla \cdot \vec{\mathbf{h}}_j(\boldsymbol{\xi}_i) \right), \end{aligned} \quad (2.35)$$

where N_f is the number of interface points, \tilde{J}_n is the Jacobian matrix, $\vec{\nabla} \phi(\boldsymbol{\xi}_i)$ is the vector of gradients of each ϕ_i function evaluated at $\boldsymbol{\xi}_i$, $\vec{\mathbf{f}}^\delta$ is a vector of flux vectors, and $\hat{\mathbf{f}}_j^b$ is the flux vector at the j^{th} interface point (obtained via extrapolation).

Note that it is possible to evaluate each of the terms in Equation (2.35) for all $i = 1, \dots, N_s$ with a series of matrix-vector multiplications.

In terms of time integration, HiFiLES uses an explicit Adaptive RK 45 (RK45) Method and local or global time stepping. Currently, a polynomial multigrid to improve the code convergence is being validated.

2.3.2 Shock Capturing and Stabilization Models

We use the method of concentration described in [75] for detecting shocks on meshes with quadrilateral elements. We are still in the process of extending the method of

concentration to triangles and are currently using Persson and Peraire's method [69, 68] for the same. We have explored both selective addition of artificial viscosity as well as modal order reduction for capturing the detected shocks effectively. Persson and Peraire have used this shock capturing tool as a stabilization method as well in their turbulence calculations. Here we show a viscous case on quadrilateral elements using the concentration method (reproduction of the result in [75]) and an inviscid case on triangles using Persson and Peraire's method.

Figures 2.1 and 2.2 show the density and energy plots for a Ma 1.2 flow over a NACA 0012 airfoil at a 5° angle of attack. The flow is at Re of 60000 and we have used 6th order polynomial interpolation in the elements for the computation. There is a bow shock in front of the airfoil and we see fish-tail shocks at the trailing edge. We can also see boundary layer formation and a Λ -shock structure on the upper side of the airfoil. Here we have used simple modal order reduction in elements with shock sensor value above a threshold. Figure 2.3 shows the elemental shock sensor values. We can see the shock sensor is able to distinguish between shocks and other smooth regions enabling the structure of the vortices and boundary layer to be preserved.

Figure 2.4 shows an inviscid flow of $\text{Ma} = 1.6$ over a NACA 0012 airfoil at 0° angle of attack on a triangle mesh. Here we use Persson and Peraire's method for shock detection and can see that we the shock has been detected and captured well. A few oscillations still remain near the strong bow shock in front of the airfoil even after enforcement of continuity of the artificial viscosity coefficients. Figures 2.5 and 2.6 show the artificial viscosity being added element-wise and after continuity enforcement respectively.

2.3.3 SA Turbulence Model and Negative $\tilde{\nu}$ Modification

The one equation SA turbulence model is one of the more commonly used turbulence models used to solve attached and moderately separated aerodynamic flows [78]. The added equation directly solves for turbulent eddy viscosity via advection, diffusion,

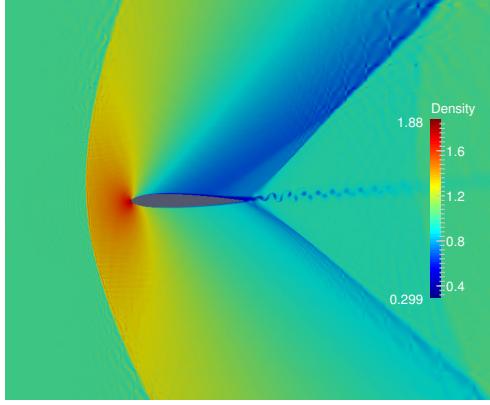


Figure 2.1: density contours for viscous flow at $\text{Ma} = 1.2$ over a NACA 0012 airfoil at 5° AoA with polynomial order 6

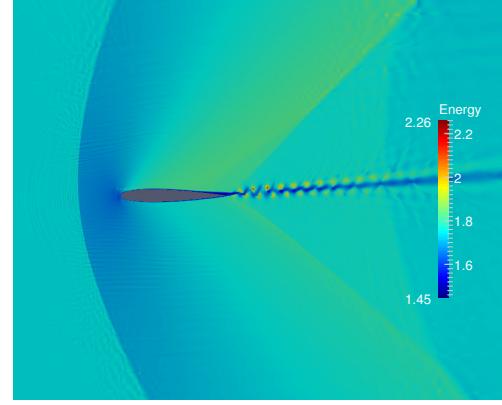


Figure 2.2: Energy contours

production and dissipation. A modified form of the equation can be written as [13, 63, 58]:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\tilde{\nu}) + \nabla \cdot (\rho\tilde{\nu}\mathbf{u}) &= c_{b_1}\tilde{S}\rho\nu\psi + \frac{1}{\sigma}[\nabla \cdot ((\mu + \mu\psi)\nabla\tilde{\nu}) + c_{b_2}\rho\nabla\tilde{\nu} \cdot \nabla\tilde{\nu}] \\ &\quad - c_{w_1}\rho f_w \left(\frac{\nu\psi}{d}\right)^2 \end{aligned} \quad (2.36)$$

where $\tilde{\nu}$ is a modified version of the kinematic eddy viscosity and ν is the kinematic viscosity. The other variables are defined as:

$$\mu_t = \begin{cases} \rho\tilde{\nu}f_{v_1} & \text{if } \tilde{\nu} \geq 0 \\ 0 & \text{if } \tilde{\nu} < 0 \end{cases} \quad \text{where} \quad f_{v_1} = \frac{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3}{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3 + c_{v_1}^3} \quad (2.37)$$

$$\tilde{S} = \begin{cases} S + \bar{S} & \text{if } \bar{S} \geq -c_{v_2}S \\ S + \frac{S(c_{v_2}^2S + c_{v_3}\bar{S})}{(c_{v_3} - 2c_{v_2})S - \bar{S}} & \text{if } \bar{S} \leq -c_{v_2}S \end{cases} \quad (2.38)$$

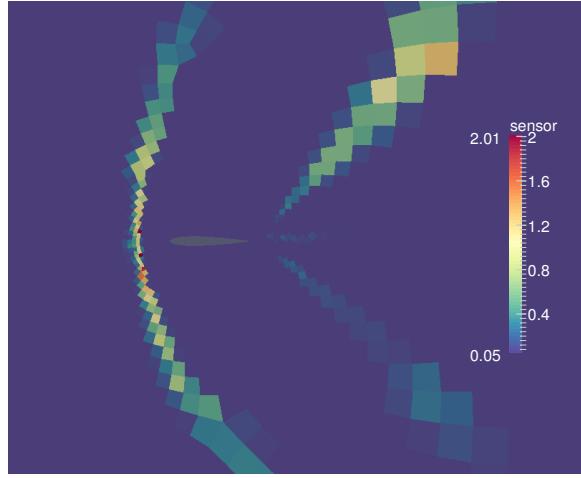


Figure 2.3: Figure shows the elemental shock “sensor” for the $\text{Ma} = 1.2$ viscous case shown in figure 2.1. The shock sensor is just the maximum value of the enhanced kernel in each element

$$S = \sqrt{\boldsymbol{\omega} \cdot \boldsymbol{\omega}} \quad \bar{S} = \frac{(\nu\psi)^2 f_{v_2}}{\kappa^2 d^2} \quad (2.39)$$

$$f_{v_2} = 1 - \frac{\psi}{1 + \psi f_{v_1}} \quad (2.40)$$

$$f_w = g \left[\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6} \right]^{1/6} \quad g = r + c_{w_2}(r^6 - r) \quad r = \frac{\nu\psi}{\tilde{S}\kappa^2 d^2} \quad (2.41)$$

where S is the magnitude of vorticity, d is the closest distance to a wall, $c_{b1} = 0.1355$, $\sigma = \frac{2}{3}$, $c_{b2} = 0.622$, $K = 0.41$, $\text{Pr}_t = 0.9$, $c_{v1} = 7.1$, $c_{v2} = 0.7$, $c_{v3} = 0.9$, $c_{w1} = \frac{c_{b1}}{K^2} + \frac{(1+c_{b2})}{\sigma}$, $c_{w2} = 0.3$, $c_{w3} = 2$.

The diffusion term, $\nabla \cdot (\rho \tilde{\nu} \mathbf{u})$, may become discontinuous in the first derivative leading to oscillations in high-order polynomials. This can lead to large negative values of the modified eddy viscosity term, $\tilde{\nu}$, significant enough to cause an unbounded

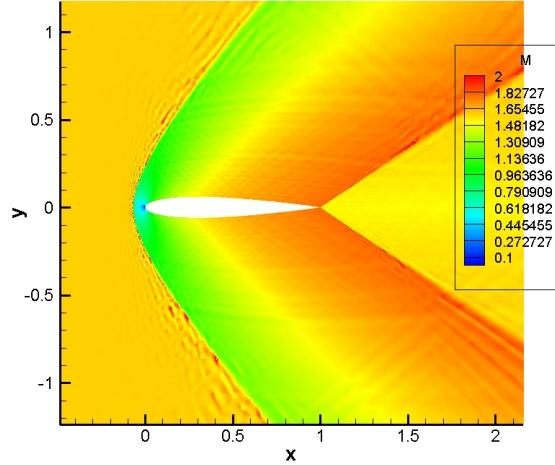


Figure 2.4: Ma contours for inviscid flow over NACA0012 at $\text{Ma} = 1.6$ and $\text{AoA} = 0^\circ$ on a triangle-mesh using Persson and Peraire's method and using artificial viscosity

solution. To prevent this, the following modification is introduced [58].

$$\psi = \begin{cases} 0.05\log(1.0 + e^{(20.0\chi)}) & \text{if } \chi \leq 10.0, \\ \chi & \text{if } \chi > 10.0, \end{cases} \quad (2.42)$$

$$\chi = \frac{\tilde{\nu}}{\nu} \quad (2.43)$$

2.3.4 Large Eddy Simulation

In order to resolve all the scales of motion in a high Re number turbulent flow, the computational mesh would have to be exceedingly fine. A practical solution is to employ the LES formulation, which only resolves the larger scales of motion and thus allows for the use of coarser meshes.

The effect of the unresolved or SGS dynamics on the solution is accounted for by an SGS model for the subgrid-scale stress τ_{ij} , which is added to the viscous stress tensor σ_{ij} given by (2.6):

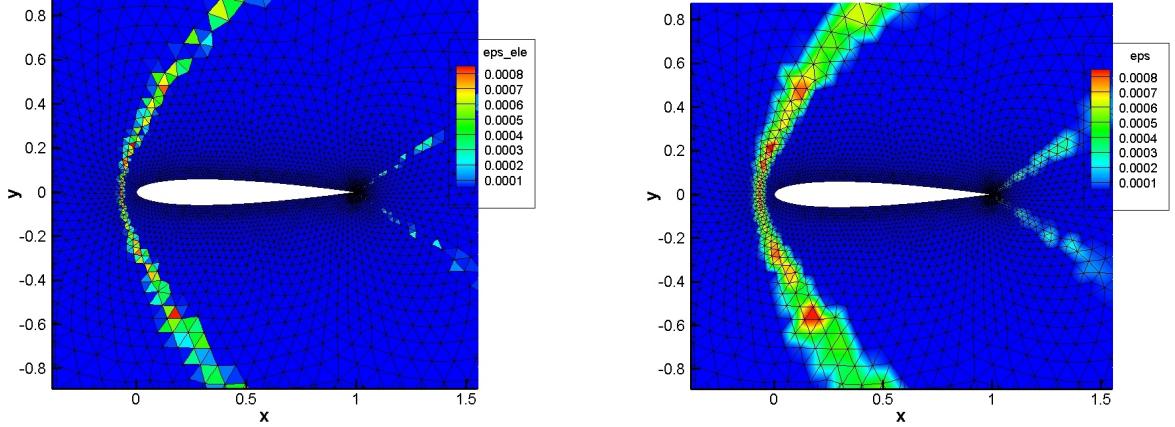
Figure 2.5: Element-wise AV coefficients for the inviscid $\text{Ma} = 1.6$ case

Figure 2.6: AV coefficients with continuity enforcement

$$\sigma_{ij} = 2\mu S_{ij}^d + \tau_{ij}, \quad (2.44)$$

$$S_{ij}^d = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right). \quad (2.45)$$

The standard Smagorinsky model [76] is available in HiFiLES:

$$\tau_{ij} = 2\mu_t S_{ij}^d, \quad (2.46)$$

$$\mu_t = \rho C_S^2 \Delta^2 |S^d|, \quad (2.47)$$

$$|S^d| = \sqrt{2S_{ij}^d S_{ij}^d}, \quad (2.48)$$

where μ_t is the eddy viscosity, $C_S = 0.1$ is the Smagorinsky coefficient and Δ is the filter width. In HiFiLES the filter width is given by (in 3D):

$$\Delta = \alpha(\text{vol})^{1/3}, \quad (2.49)$$

where $\alpha \geq 1$ is a user-defined scaling factor and vol is the element volume.

HiFiLES also includes the WALE model [61] and the Similarity model [5]. The Similarity model incorporates a low-pass filtering operator, for which several choices

are available in HiFiLES: a discrete Gaussian filter[48], a high-order commuting Vasilyev-type filter[81, 82] and a modal Vandermonde-type filter[8].

The modal filter can be used on unstructured tetrahedral meshes. For details of these operators, see Lodato, Castonguay and Jameson [48] and Bull and Jameson [11]. One can combine the similarity model with the Smagorinsky or WALE model to form a mixed SGS model. The WSM model, first proposed by Lodato et al. [49], was used in simulations of the flow over a square cylinder (see Section 2.4.7).

2.3.5 Computing Architecture and Scalability

The HiFiLES code has been designed to work on multi-CPU as well as multi-CPU-GPU platforms. The FR method in its current form with explicit time-stepping has a great potential for parallelization. Since the solution points are not explicitly shared between elements, most of the computations are element-local enabling an efficient use of shared memory on GPUs. Also, several computations are independent for each solution point and the highly parallelizable nature of GPUs becomes very useful. A detailed description of the parallelization of the FR method, along with scalability and performance analysis has been performed in [16].

2.4 Verification and Validation

2.4.1 Method of Manufactured Solutions

This section describes the test of HiFiLES’s spatial order of accuracy using the MMS in 2D and 3D for viscous flows. As shown by Salari et. al [74], the MMS test rigorously assesses the correctness of implementation of a solver of Partial Differential Equation (PDE)s. Simplex elements are crucial for simulations in unstructured meshes and have a more complex implementation than squares and hexahedra. As a result, we perform the MMS test in grids using simplex elements.

The MMS test for NS solvers requires checking the solver’s solution against an exact solution. Such exact solution can be chosen arbitrarily. The NS equations can

be satisfied with this arbitrary solution by including a time-dependent source term in the equations. Then, we solve

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = S \quad (2.50)$$

For the following tests, we selected a smooth exact solution, so aliasing does not pollute the results. We picked

$$U_{2D} = \begin{pmatrix} \sin(k(x+y) - \omega t) + a \\ \sin(k(x+y) - \omega t) + a \\ \sin(k(x+y) - \omega t) + a \\ (\sin(k(x+y) - \omega t) + a)^2 \end{pmatrix} \quad (2.51)$$

$$U_{3D} = \begin{pmatrix} \sin(k(x+y+z) - \omega t) + a \\ (\sin(k(x+y+z) - \omega t) + a)^2 \end{pmatrix}$$

To find the value of S , we plug the values of our selected U into the left-hand side of Equation (2.51) and simplify. The resulting expression is S . We let $\text{Pr} = 0.72, \gamma = 1.4, k = \pi, \omega = \pi, a = 3.0$ and $\mu = 0.001$.

The meshes used have dimensions $[-1, 1] \times [-1, 1]$ in 2D and $[-1, 1] \times [-1, 1] \times [-1, 1]$ in 3D. Periodic boundary conditions were applied on the boundaries of the square and cube domains. Uniform square and cubic meshes were created and then each element was subdivided into triangles or tetrahedra. Two triangles were created from each square, and six tetrahedra were created from each cube. Consequently, in 2D a $N \times N$ mesh contains $2N^2$ triangles, and in 3D a $N \times N \times N$ mesh contains $6N^3$ tetrahedra.

In 3D, the time step was $1e-4$ seconds and 10 seconds of flow were simulated. In 2D, the time step was $1e-6$ seconds and 1 second of flow was simulated. The time-stepping scheme used was the low-storage, 4th order accurate RK45 method.

Polynomial Order	Mesh:	2x2x2	4x4x4	8x8x8	16x16x16	Overall Order of Accuracy
$p = 1$	L_2 error	5.76e-01	1.35e-01	3.22e-02	7.90e-03	
	$\mathcal{O}(L_2)$		2.10	2.06	2.03	2.06
$p = 2$	L_2 error	4.09e-01	5.52e-02	6.87e-03	8.53e-04	
	$\mathcal{O}(L_2)$		2.89	3.01	3.01	2.97
$p = 3$	L_2 error	9.77e-02	5.97e-03	3.78e-04		
	$\mathcal{O}(L_2)$		4.03	3.98		4.01
$p = 4$	L_2 error	1.12e-02	6.39e-04	2.07e-05		
	$\mathcal{O}(L_2)$		4.13	4.95		4.54
$p = 5$	L_2 error	1.53e-01	5.08e-03	6.92e-05		
	$\mathcal{O}(L_2)$		4.91	6.20		5.55

Table 2.1: Accuracy of HiFiLES for NS equations with source term in tetrahedral meshes at $t = 10$. L_2 error is the L_2 -norm of the error in the energy field: ρe

Polynomial Order	Mesh:	2x2x2	4x4x4	8x8x8	16x16x16	Overall Order of Accuracy
$p = 1$	L_2 error	1.98e+01	9.57e+00	4.55e+00	2.19e+00	
	$\mathcal{O}(L_2)$		1.05	1.07	1.06	1.06
$p = 2$	L_2 error	1.17e+01	2.98e+00	7.10e-01	1.71e-01	
	$\mathcal{O}(L_2)$		1.97	2.07	2.06	2.03
$p = 3$	L_2 error	3.17e+00	3.81e-01	4.73e-02		
	$\mathcal{O}(L_2)$		3.06	3.01		3.03
$p = 4$	L_2 error	5.21e-01	4.27e-02	2.69e-03		
	$\mathcal{O}(L_2)$		3.61	3.99		3.80
$p = 5$	L_2 error	3.20e+00	1.88e-01	4.79e-03		
	$\mathcal{O}(L_2)$		4.09	5.29		4.69

Table 2.2: Accuracy of HiFiLES for NS equations with source term in tetrahedral meshes at $t = 10$. L_2 error is the L_2 -norm of the error in the gradient of the energy field: $\frac{\partial}{\partial x_i}(\rho e)$

Polynomial Order	Mesh:	4x4	8x8	16x16	32x32	64x64	Overall Order of Accuracy
$p = 1$	L_2 error	7.92e-01	1.84e-01	4.36e-02	1.07e-02	2.68e-03	2.05
	$\mathcal{O}(L_2)$		2.10	2.08	2.03	2.00	
$p = 2$	L_2 error	1.29e-01	1.61e-02	1.95e-03	2.33e-04	2.86e-05	3.04
	$\mathcal{O}(L_2)$		3.00	3.05	3.06	3.03	
$p = 3$	L_2 error	1.01e-02	9.25e-04	5.71e-05	3.65e-06	2.35e-07	3.88
	$\mathcal{O}(L_2)$		3.45	4.02	3.97	3.96	
$p = 4$	L_2 error	2.60e-03	6.33e-05	2.00e-06	6.49e-08	3.62e-09	4.88
	$\mathcal{O}(L_2)$		5.36	4.98	4.95	4.16	
$p = 5$	L_2 error	7.15e-05	3.87e-06	6.31e-08			5.07
	$\mathcal{O}(L_2)$		4.21	5.94			

Table 2.3: Accuracy of HiFiLES for NS equations with source term in triangular meshes at $t = 1$. L_2 error is the L_2 -norm of the error in the energy field: ρe

Tables (2.3) and (2.1) show the spatial order of accuracy achieved when calculating the energy fields ρe in 2D and 3D, respectively. Tables (2.4) and (2.2) show the order of accuracy for the gradient of the energy field $\frac{\partial}{\partial x_i}(\rho e)$ in 2D and 3D, respectively. Because of the exact solutions that were picked, the exact values of the gradients of ρe in the x, y, z directions are equal.

As expected[34], the order of accuracy of the solution is $p + 1$ and the order of accuracy of the gradient of the solution is p , where p is the order of the polynomial used to approximate the solution fields. In the fifth order simulations, the relatively large time step introduces errors larger than the spatial discretization errors. Hence we observe sub-optimal orders of convergence in the coarsest meshes.

2.4.2 Subsonic laminar flat-plate

Computations of the flow over a subsonic flat-plate have been performed and validated against the Blasius' solution for laminar boundary layer. The flow conditions are Ma 0.5, 0° angle of attack and $Re = 1 \cdot 10^6$ based on the plate length. The governing equations are the 2D NS equations with constant ratio of specific heats of 1.4, Prandtl number of 0.72 and constant dynamic viscosity of $1.827 \cdot 10^{-5} Pa \cdot s$.

Polynomial Order	Mesh:	4x4	8x8	16x16	32x32	64x64	Overall Order of Accuracy
$p = 1$	L_2 error	1.61e+01	8.31e+00	3.81e+00	1.71e+00	7.84e-01	
	$\mathcal{O}(L_2)$		0.96	1.12	1.15	1.13	1.10
$p = 2$	L_2 error	4.05e+00	8.16e-01	1.90e-01	4.54e-02	1.11e-02	
	$\mathcal{O}(L_2)$		2.31	2.11	2.06	2.04	2.12
$p = 3$	L_2 error	4.71e-01	6.39e-02	7.03e-03	7.75e-04	8.84e-05	
	$\mathcal{O}(L_2)$		2.88	3.18	3.18	3.13	3.11
$p = 4$	L_2 error	1.01e-01	4.30e-03	2.31e-04	1.41e-05	5.27e-06	
	$\mathcal{O}(L_2)$		4.56	4.22	4.04	1.42	3.67
$p = 5$	L_2 error	5.04e-03	2.50e-04	7.80e-06			
	$\mathcal{O}(L_2)$		4.33	5.00			4.67

Table 2.4: Accuracy of HiFiLES for NS equations with source term in triangular meshes at $t = 1$. L_2 error is the L_2 -norm of the error in the gradient of the energy field: $\frac{\partial}{\partial x_i}(\rho e)$

Mesh	First cell height	# of cells in boundary layer	p_3	p_4	p_5	p_6
Mesh a0 (140 = 14x10)	0.00075	2	✗	✗	✗	✓
Mesh a1 (560 = 28x20)	0.000375	4	✗	✗	✓	✓
Mesh a2 (2240 = 56x40)	0.0001875	8	✗	✓	✓	✓
Mesh a3 (8960 = 112x80)	0.0000935	16	✓	✓	✓	✓

Table 2.5: HiFiLES convergence using different grids and polynomial order. ✗ / ✓ indicates not converged/converged resp.

The objective of this study is to determine the minimum number of elements and the order of polynomial required to converge the flat-plate simulation using HiFiLES. Four different numerical grids have been used in this study (2, 4, 8, 16 elements inside the boundary layer) and four polynomial orders (p_3 – p_6). The results, summarized in Table 2.5, show that a minimum number of elements is needed in the boundary layer depending on the polynomial order to obtain satisfactory convergence (free from inter-element jumps).

The results are compared with the Blasius' solution for laminar boundary layer with satisfactory results, and some details of the solutions are presented in Fig. 2.7

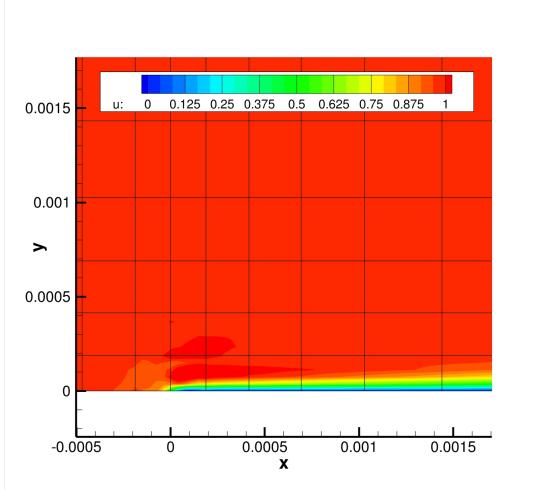


Figure 2.7: Detail of the flat-plate leading edge ($x=0.0$, mesh a2).

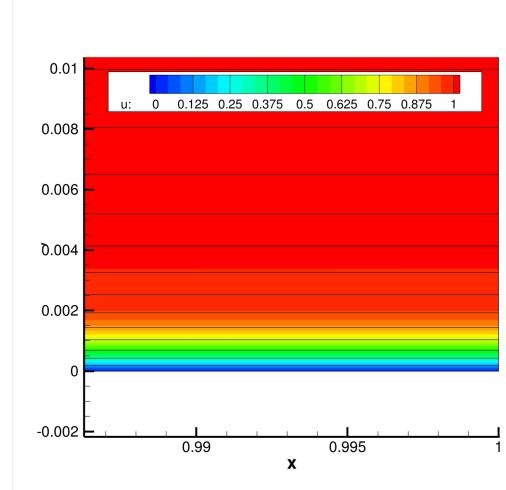


Figure 2.8: Flow solution at the end of the flat-plate ($x=1.0$, mesh a2).

(leading edge), and Fig. 2.8 (end of the flat-plate). It is important to note that in this particular case (mesh a2) the flat-plate boundary layer is captured using 8 elements, while in a second order solver it would be necessary of the order of 30 elements inside the boundary layer.

To finalize, it is critical to note that the absence of a local time stepping technique in HiFiLES increases the required number of iterations to obtain a converged solution. However, we have noticed an improvement of the rate of convergence as we refine the grid (see Fig. 2.9). The obtained convergence rate is comparable to a second order numerical code (e.g. SU2 [65, 64]) running using a similar numerical time integration (see Fig. 2.10).

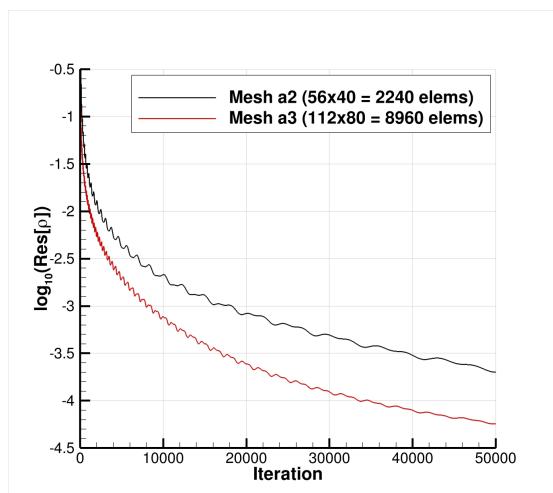


Figure 2.9: Convergence comparison (3^{rd} order, finest grids).

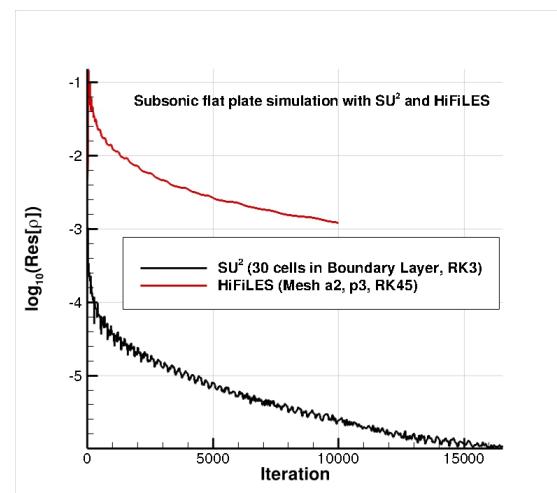


Figure 2.10: Comparison of HiFiLES with SU2 using a similar time integration scheme.

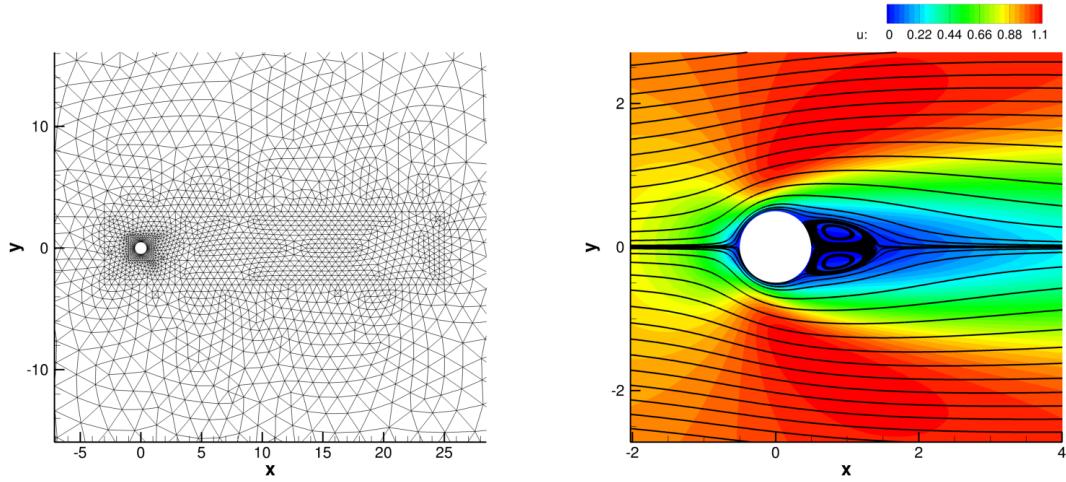
2.4.3 Circular Cylinder

The classic test case of laminar flow past a circular cylinder at low Re number has also been chosen as a verification and validation case for the 2D NS equations in HiFiLES, and the results are compared to existing experimental data and simulation results [66]. Two separate cases are computed: first, the steady flow past the cylinder at $Re = 20$, and second, the unsteady flow past the cylinder at $Re = 100$, where Re is based upon the diameter of the cylinder. For both cases, Ma number is set to 0.1 in order to recover nearly incompressible flow for comparisons with the existing incompressible results. The remaining flow conditions are 0° angle of attack, a constant ratio of specific heats of 1.4, a Prandtl number of 0.72, a free-stream temperature of $300K$, and a free-stream dynamic viscosity of $1.853 \cdot 10^{-5} Pa \cdot s$ (laminar viscosity varies according to Sutherland's law during the simulation).

The two simulations are performed with third order polynomials on a mesh with 4988 total elements that contains quadrilateral elements near the body of the cylinder and triangular elements out to the far-field. There is a small refinement box immediately downstream of the cylinder to help resolve features in the wake. The rectangular far-field boundaries are located approximately 30 diameters away from the cylinder in the upstream, upward, and downward directions and 50 diameters away in the downstream direction. A view of the mesh near the cylinder surface is show in Fig. 2.11.

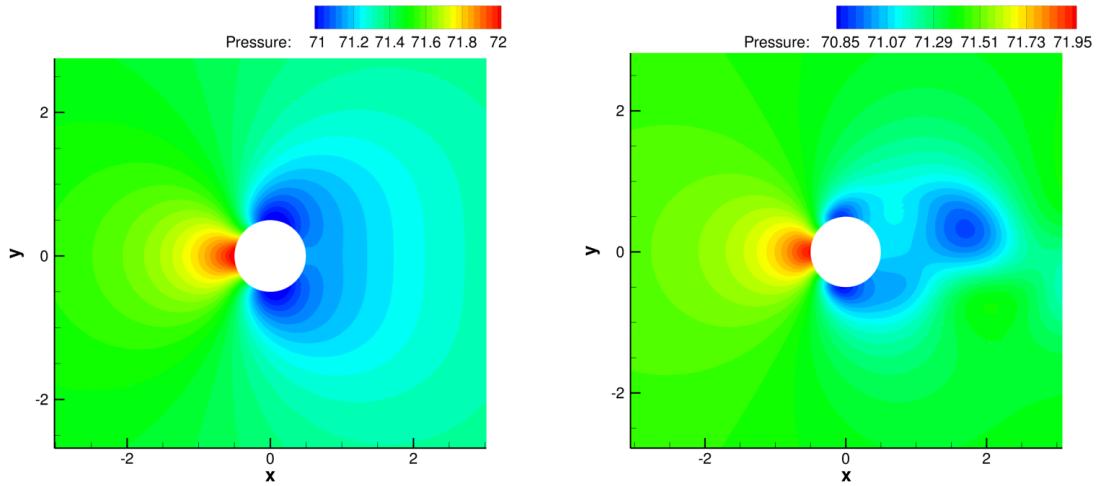
The flow around the cylinder for $Re = 20$ is steady, and it features a large recirculation region behind the cylinder. Fig. 2.11 presents x-velocity contours around the cylinder along with streamlines. The length of the recirculation region can be determined from the streamlines, and a length of approximately one cylinder diameter agrees well with reported results for $Re = 20$. The coefficient of drag computed by HiFiLES is 2.043, which is close to the value of 2.01 reported by Park et al. [66] Pressure contours around the cylinder are shown in Fig. 2.12.

When Re is increased to 100, the flow around the cylinder becomes unsteady and exhibits periodic vortex shedding. This periodic shedding in the wake behind the cylinder can be seen in the instantaneous contours of x-velocity and vorticity



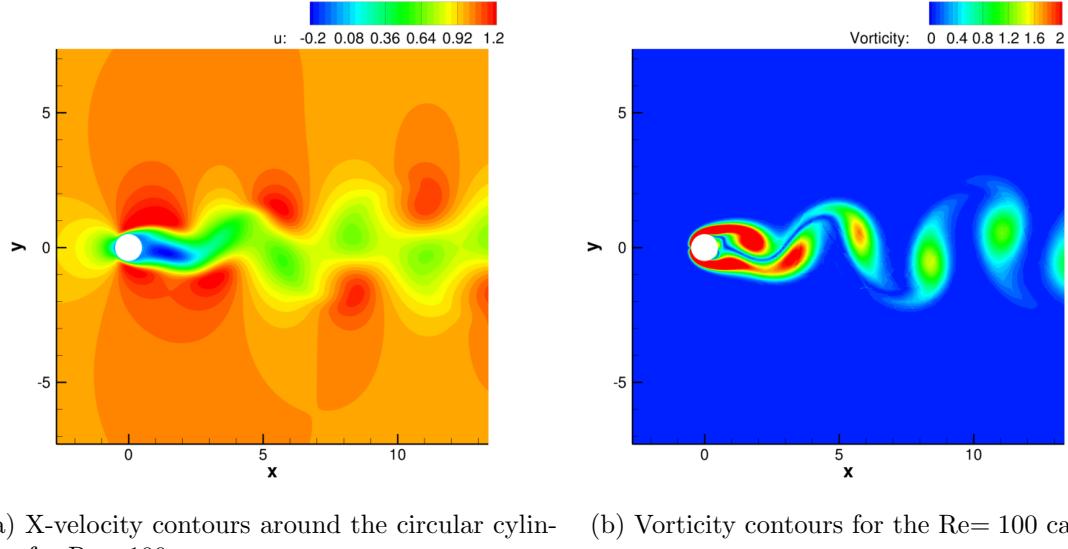
(a) Zoom view of the mesh near the cylinder. (b) X-velocity contours and streamlines around the circular cylinder for $Re = 20$.

Figure 2.11: The mesh for the circular cylinder simulations along with x-velocity contours for the $Re = 20$ case.



(a) Pressure contours for the $Re = 20$ case. (b) Pressure contours for the $Re = 100$ case.

Figure 2.12: Pressure contours for the steady and unsteady (instantaneous) cylinder cases.



(a) X-velocity contours around the circular cylinder for $\text{Re} = 100$. (b) Vorticity contours for the $\text{Re} = 100$ case.

Figure 2.13: Instantaneous solution contours for the unsteady cylinder case.

in Fig. 2.13, and it also results in periodic fluctuations in the force coefficients on the cylinder. HiFiLES reports an average drag coefficient of 1.339 with a maximum deviation from this value of 0.0092, which agree excellently with the values reported by Park et al. [66] of 1.33 and 0.0091 for the average C_d and maximum deviation from it, respectively. Instantaneous pressure contours for the $\text{Re} = 100$ case can be seen in Fig. 2.12. The asymmetry that is visible in the pressure contours contributes to the variability in the drag coefficient.

2.4.4 SD7003 airfoil at 4° angle of attack

Abundant literature documents flow around a SD7003 infinite wing and airfoil. Hence, physical experiments [62, 70] and numerical simulations [30, 86, 17, 67, 80] of flow over this geometry can be used to benchmark HiFiLES.

The simulations on the 2D geometry were performed on a circular domain with a radius of $50c$, where c is the airfoil's cord length, centered at the leading edge of the airfoil. The boundary conditions are characteristic on the outer edge and adiabatic no-slip wall on the airfoil. The Mach number for all simulations was $\text{Ma} = 0.2$. The reported lift and drag coefficients in Table (2.6) correspond to the average of lift and drag coefficients over 13 periods after the flow reached a pseudo-periodic state. More details are provided by Williams [95].

Source	$Re = 10K$		$Re = 22K$		$Re = 22K$	
	C_L	C_D	C_L	C_D	C_L	C_D
Uranga et al.[80]	0.3755	0.04978	0.6707	0.04510	0.5730	0.02097
c_{dg}, κ_{dg}	0.3719	0.04940	0.6722	0.04295	0.5831	0.01975
c_+, κ_+	0.3713	0.04935	0.6655	0.04275	0.5774	0.02005

Table 2.6: Time-averaged values of the lift and drag coefficients for the SD7003 airfoil flows with $Re = 10,000, 22,000, 60,000$

The average lift and drag coefficients are in close agreement with the results by Uranga el. al [80]. The density contours in Figures (2.14),(2.15), and (2.16) show that vortical structures are captured for a reasonable distance away from the airfoil despite the fact that elements are coarser away from the airfoil.

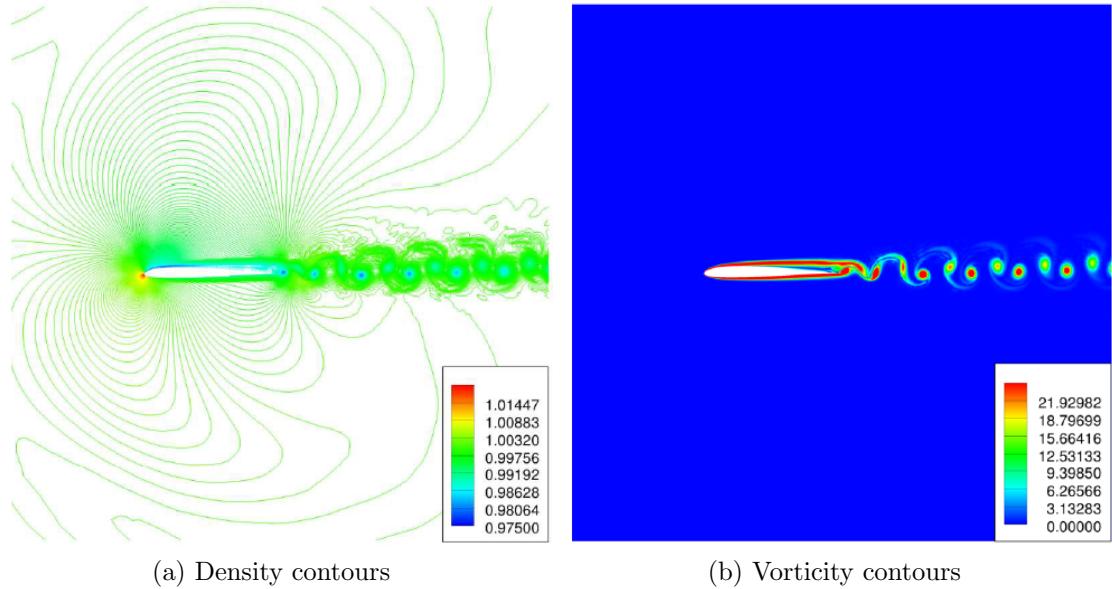


Figure 2.14: Density and vorticity contours for the flow with $Re = 10,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements

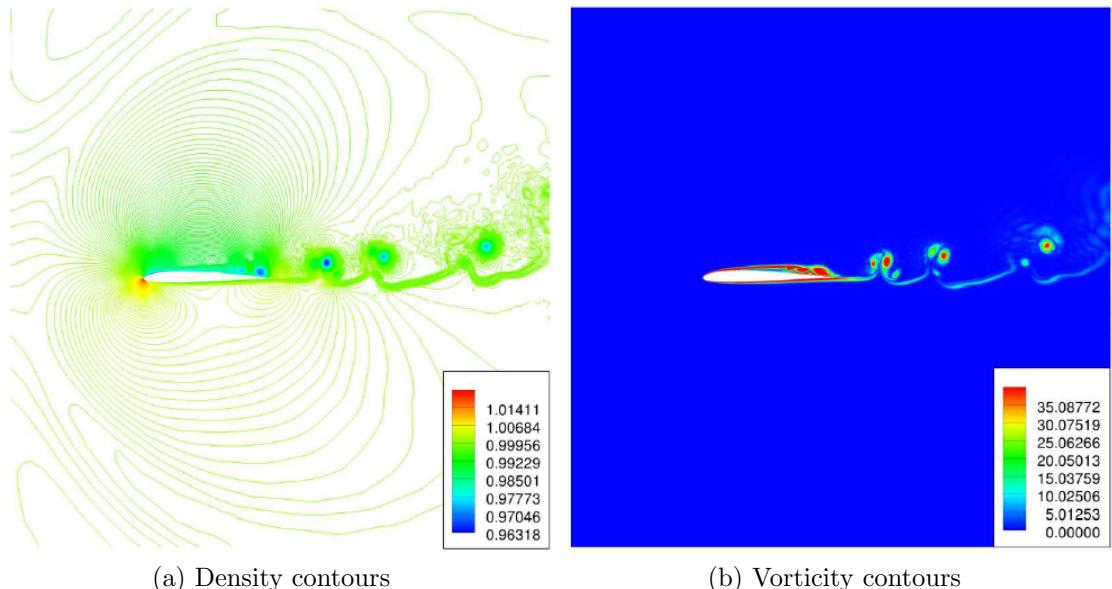


Figure 2.15: Density and vorticity contours for the flow with $Re = 22,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements

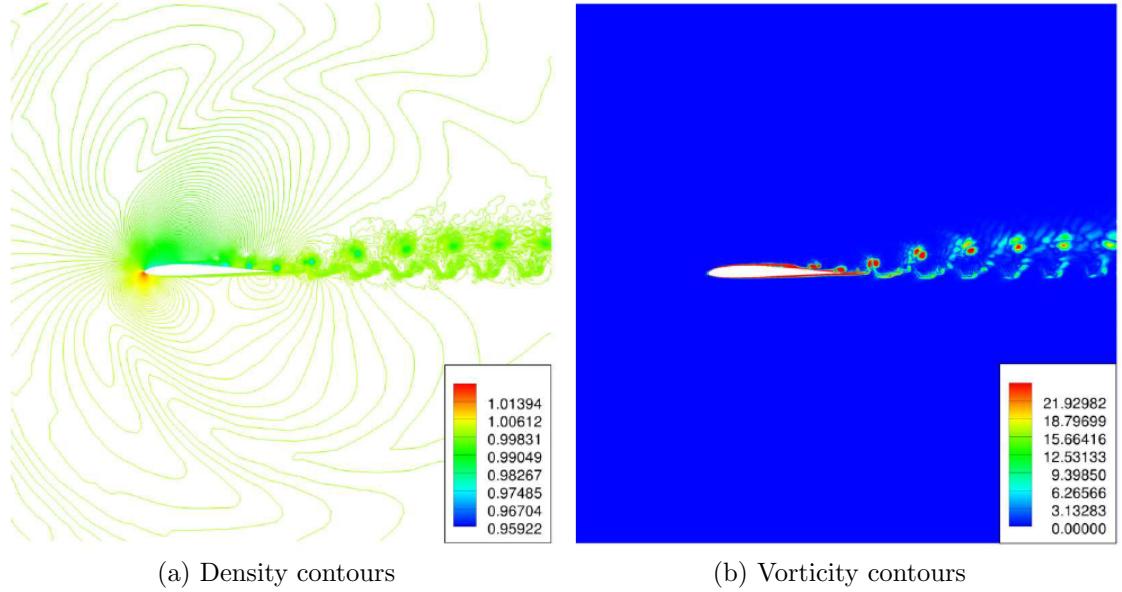


Figure 2.16: Density and vorticity contours for the flow with $Re = 60,000$ around the SD7003 airfoil. $p = 2$ on unstructured triangular grid with $N = 25,810$ elements

2.4.5 SD7003 wing section at 4° angle of attack

To validate HiFiLES’s performance in 3D simulations, we extrude the SD7003 geometry from Section(2.4.4) by $0.2c$ in the z -direction and apply periodic boundary conditions at $z = 0$ and $z = 0.2c$. Table (2.7) shows the time-averaged lift and drag coefficients.

Source	$Re = 10K$	
	C_L	C_D
Uranga et al.[80]	0.3743	0.04967
c_{dg}, κ_{dg}	0.3466	0.04908
c_+, κ_+	0.3454	0.04903

Table 2.7: Time-averaged values of the lift and drag coefficients for the SD7003 wing-section in a flow with $Re = 10,000$

It is worth noting that the vortical structures are preserved better than in the 2D case. Table (2.7) demonstrates that HiFiLES provides average lift and drag coefficient estimates in close agreement with experiments.

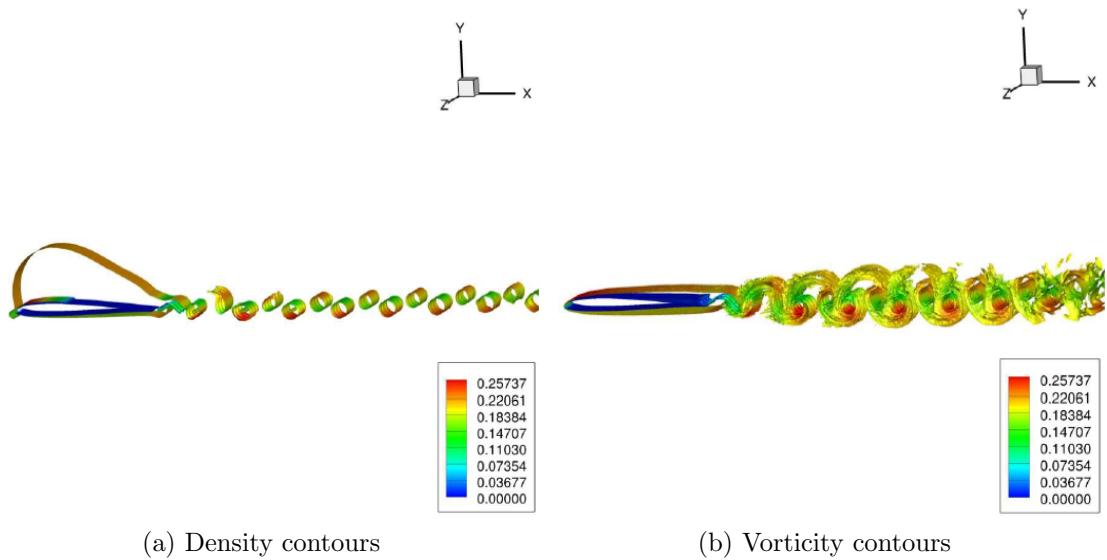


Figure 2.17: Density and vorticity isosurfaces colored by Mach number for the flow with $Re = 10,000$ around the SD7003 wing-section. $p = 3$ on unstructured tetrahedral grid with $N = 711,332$ elements

2.4.6 Taylor-Green Vortex at $\text{Re} = 1,600$

The TGV is a simple test of the resolution of the small scales of a turbulent flow by a numerical method. The compressible TGV at $\text{Re}= 1600$ was one of the benchmark problems in the 1st and 2nd International Workshops on High-Order CFD Methods [91]. A reference solution was computed by Debonis [26] using a high-order Dispersion Relation-Preserving (DRP) scheme on a mesh of 512^3 elements. The results presented here were obtained by Bull and Jameson using FR to recover the fourth-order-accurate DG and SD schemes in HiFiLES [11, 12]. We also compare our results to those of Beck and Gassner [7], who used a fourth-order filtered DG method on a mesh of 64^3 elements. From a simple initial condition in a triply-periodic box of dimensions $[0 : 2\pi]^3$, interactions between vortices cause the flow to develop in a prescribed manner into a mass of elongated vortices across a range of scales. The initial condition is specified as

$$u(t_0) = u_0 \sin(x/L) \cos(y/L) \cos(z/L), \quad (2.52)$$

$$v(t_0) = -u_0 \cos(x/L) \sin(y/L) \cos(z/L), \quad (2.53)$$

$$w(t_0) = 0, \quad (2.54)$$

$$p(t_0) = p_0 + \frac{\rho_0 V_0^2}{16} \left[\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right] \left[\cos\left(\frac{2z}{L}\right) + 2 \right], \quad (2.55)$$

where $L = 1$, $u_0 = 1$, $\rho_0 = 1$ and $p_0 = 100$. Ma is set to 0.08 (consistent with the initial pressure p_0) and the initial temperature is 300K.

Figs. 2.18 (a) and (b) show the volume-averaged kinetic energy $\langle k \rangle$ on (a) hexahedral meshes of 16^3 , 32^3 and 64^3 elements and (b) tetrahedral meshes (formed by splitting the hexahedral meshes). The reference solution, labeled as ‘DRP-512’ is plotted for comparison. Figs. 2.18 (c) and (d) show the kinetic energy dissipation rate, given by $\epsilon = -d\langle k \rangle/dt$ versus the reference solution and the results of Beck and Gassner [7], labeled as ‘Beck-DG-64x4’. On the finest hexahedral and tetrahedral meshes the kinetic energy and dissipation rate predictions match the reference solution, demonstrating that the high-order numerical scheme is able to resolve the important flow dynamics on a relatively coarse mesh. As a qualitative measure of

the resolution of the turbulent flow structures, Figure 2.19 shows isosurfaces of the q criterion at four times during the simulation. The evolution of complex small scale structures is evident.

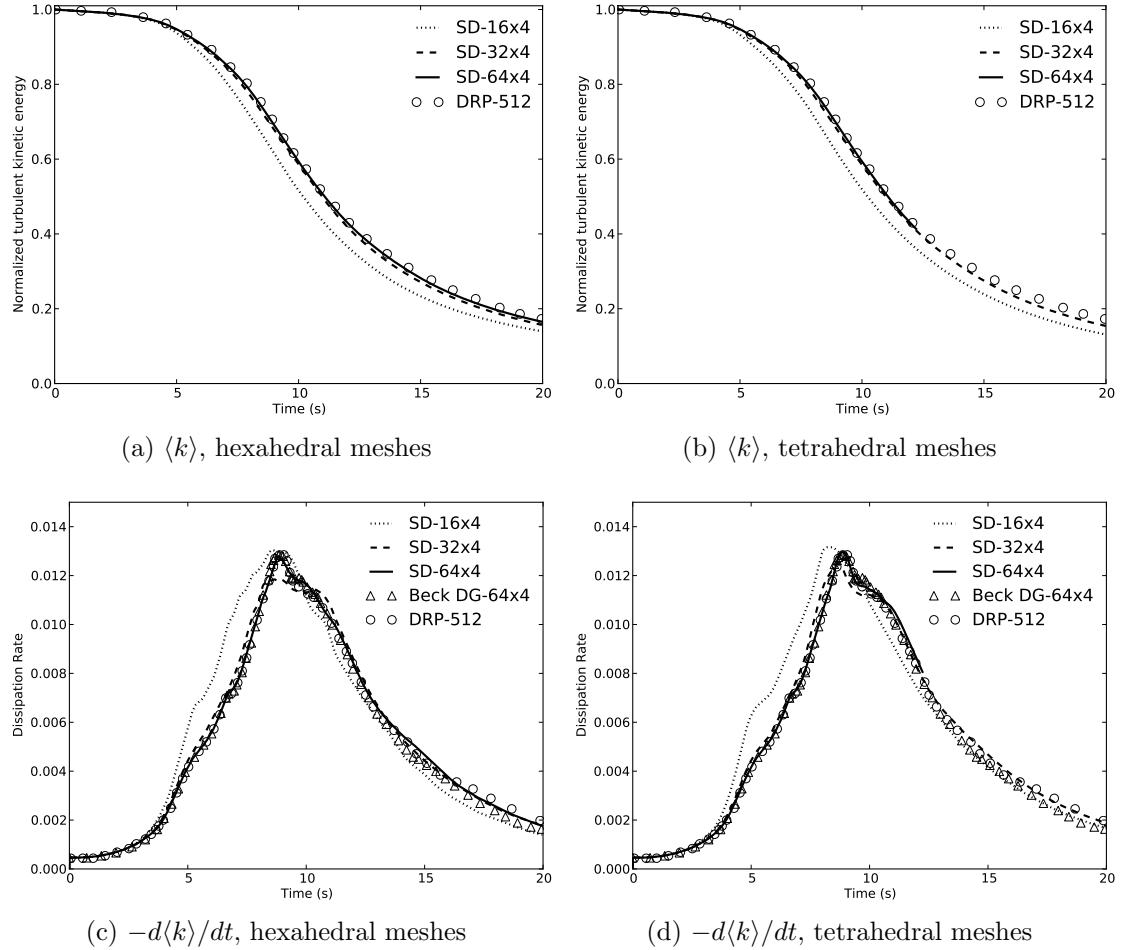


Figure 2.18: Taylor-Green vortex results on hexahedral and tetrahedral meshes from Bull and Jameson [11]. (a, b) Evolution of average kinetic energy $\langle k \rangle$; (c, d) dissipation rate $-d\langle k \rangle/dt$. ‘SD- $M \times N$ ’ refers to M^3 mesh, N th-order accurate SD scheme. (---) 4th-order DG on 64^3 mesh [7]; (○) DNS [26].

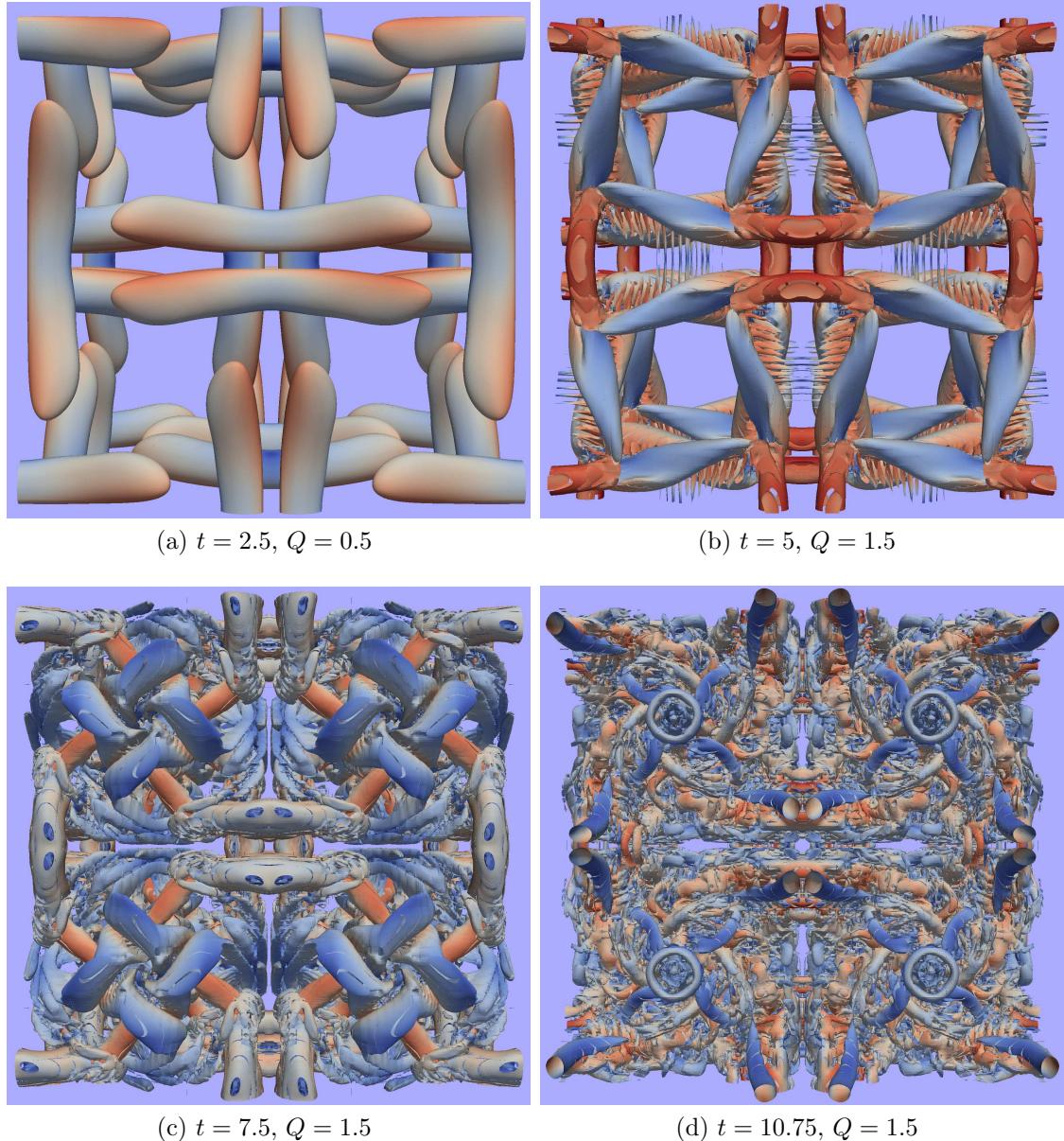


Figure 2.19: TGV solution on the fine mesh using fourth order accurate DG method, showing isosurfaces of q criterion colored by velocity magnitude at time $t = 2.5$ to 10.75 seconds.

2.4.7 LES of Flow Over a Square Cylinder at $Re = 21,400$

Using the FR method to recover the fourth-order accurate SD scheme, the flow over a square cylinder of side D in a domain of $21D \times 12D \times 3.2D$ (see Figure 2.20) at $Re = 21,400$ and $Ma = 0.3$ was simulated, for which LDV experimental data is available [56, 55]. A tetrahedral mesh of 87,178 elements was generated giving a total of 1.74M Degrees of Freedom (DoF) since there are 20 solution points per element at fourth order accuracy. Time discretization was by the fourth-order five-stage explicit RK scheme. A total time of 250 seconds was simulated and time-averaged quantities were calculated over the last 100 seconds (approx. 5 flow-through periods). The WSM model (see Section 2.3.4) based on the modal Vandermonde filter [11] was used with the Breuer-Rodi three-layer wall model [9] within $0.2D$ of the wall. The computation took around 60 hours on 7 GPUs in the lab's own cluster. Figure 2.20 shows the computational mesh including all the DoF. Figure 2.21 shows an isosurface of the q -criterion colored by velocity magnitude, illustrating the structures present in the turbulent boundary layer and wake. Figures 2.22 (a, b) show the normalized mean streamwise and vertical velocity components $\langle u \rangle / u_B$ and $\langle v \rangle / u_B$ respectively along several vertical lines in the wake. Figures 2.22 (c, d) show the normalized mean Reynolds stress components $\langle u' u' \rangle / u_B^2$ and $\langle u' v' \rangle / u_B^2$ along the same lines. For comparison, high-order LES results computed by Lodato and Jameson [48] using the SD method and the WSM model on a hexahedral mesh of 2.3M DoF are plotted. Mean velocities are accurately predicted although the accuracy is reduced near the cylinder owing to the coarse tetrahedral resolution in the boundary layer. The Reynolds stresses are less accurately predicted than the mean velocities but are broadly correct. These results highlight the advantages of using HiFiLES for LES of turbulent flows: the ability to obtain good results on coarse meshes and the ability to use unstructured tetrahedral meshes.

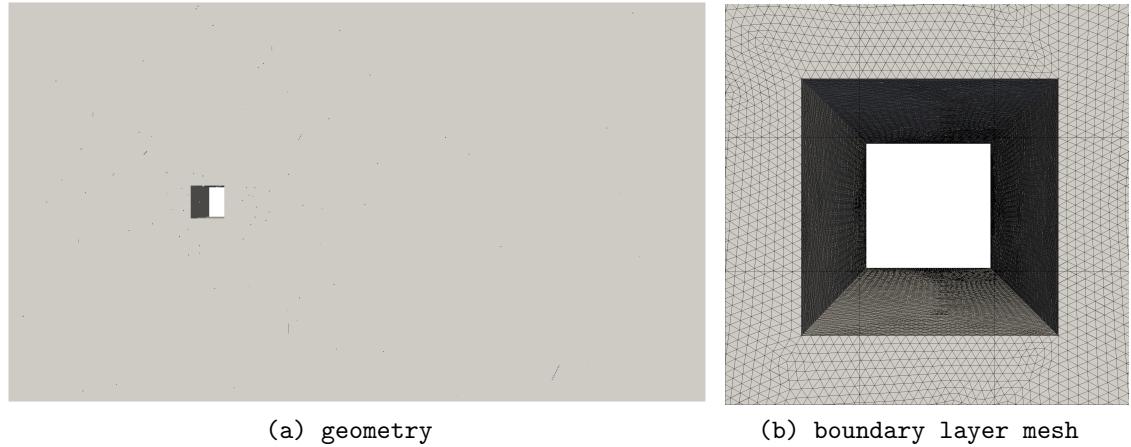


Figure 2.20: Square cylinder geometry and tetrahedral boundary layer mesh showing all degrees of freedom



Figure 2.21: Isosurface of the q -criterion colored by velocity magnitude showing the wake behind the square cylinder

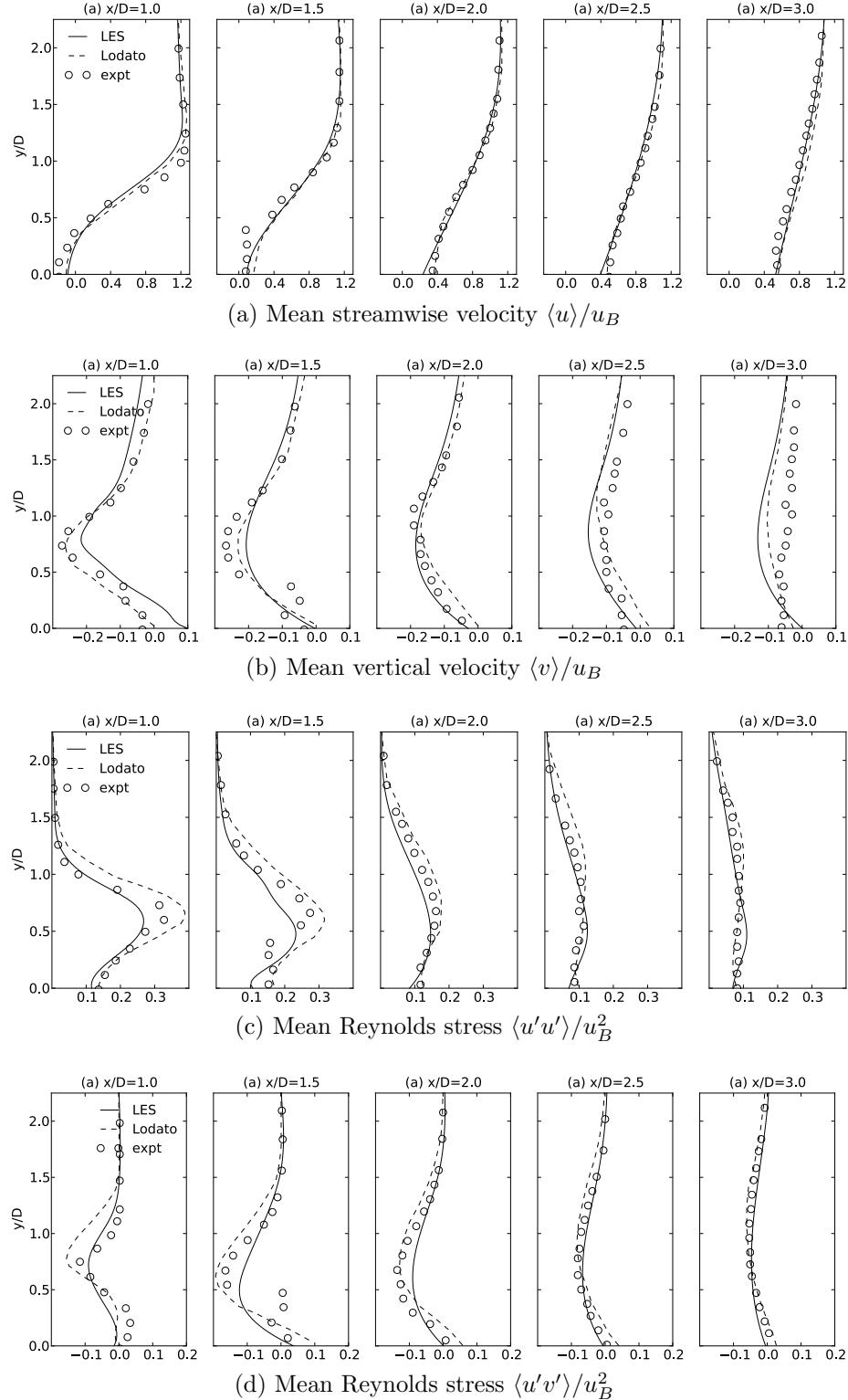
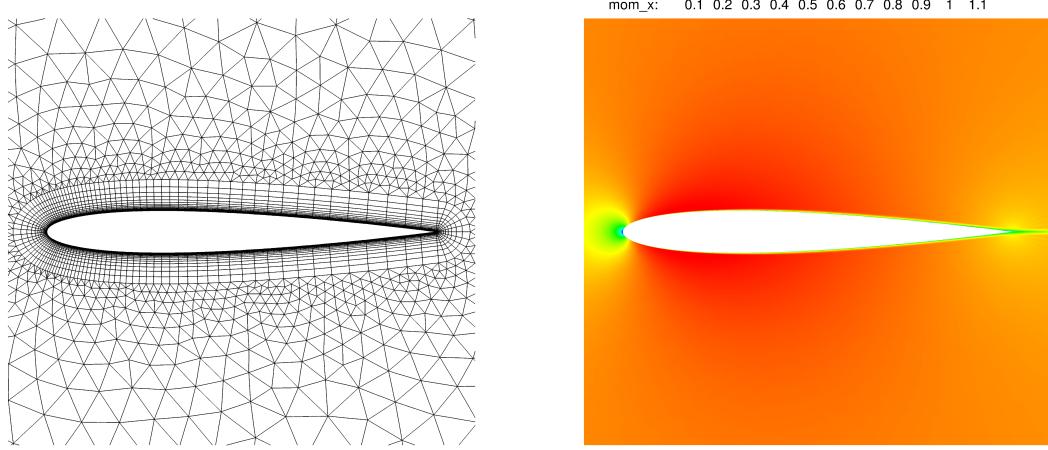


Figure 2.22: (a) Mean streamwise and vertical velocity and mean Reynolds stresses along vertical lines in the wake. (—) current results, (- - -) 4th order SD+WSM on hexahedral mesh by Lodato and Jameson [48], (○) LDV experiments by Lyn et al. [56, 55].



(a) Zoomed view of the mixed element mesh near the NACA 0012 airfoil.
(b) X-momentum contours near the NACA 0012 airfoil

Figure 2.23: Turbulent flow past a NACA 0012 airfoil at $\text{Re} = 6$ million, $\text{Ma} = 0.15$, $\alpha = 0^\circ$ using FR to recover 4th order accurate DG method and the SA turbulence model.

2.4.8 NACA 0012 airfoil at 0° angle of attack, $\text{Re} = 6$ million, $\text{Ma} = 0.15$

In this section, the NACA 0012 airfoil is used to study the accuracy of the SA turbulence model coupled with FR. The NACA 0012 is commonly used as a validation case for all turbulence models and a large database of results are available at the NASA Turbulence Modeling Resource website. A 6,539 element quad/triangle mixed mesh is used with a NACA 0012 airfoil of chord length 1.0 and a far field boundary 20 chord lengths away. The results are compared with CFL3D and experimental results from Gregory & O'Reilly [32].

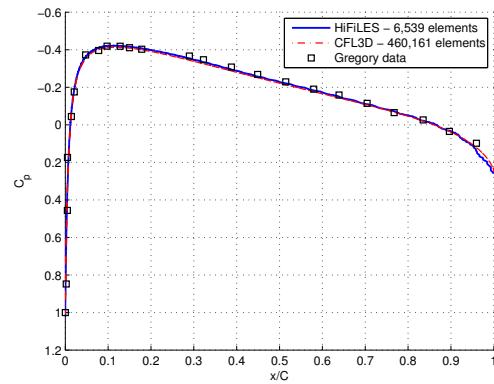


Figure 2.24: Pressure coefficient on the NACA 0012 airfoil at $\text{Re} = 6$ million, $\text{Ma} = 0.15$, $\alpha = 0^\circ$ using FR to recover 4th order accurate DG method and the SA turbulence model.

2.5 Conclusion

In this chapter, we have presented a comprehensive description, verification and validation of the HiFiLES solver. In its first version, HiFiLES offers to its users an optimal implementation of the FR methodology in unstructured 3D grids using GPUs or traditional Message Passing Interface (MPI). The implementation has been verified via MMS. The code has been tested in some difficult NS and LES problems with very satisfactory results.

The power of the FR method is in its flexibility, efficiency and accuracy. Different high-order schemes can be recovered by choosing a single parameter, allowing the numerical behavior to be fine-tuned. Though the use of explicit timestepping sets limits on the CFL condition, the fact that HiFiLES can be run on high performance multi-GPU platforms more than compensates for this.

Despite considerable advances in the accuracy and versatility of SGS models, current industrial CFD codes are restricted in their ability to perform LES of turbulent flows by the use of highly dissipative second-order numerical schemes. Therefore, in order to advance the state of the art in industrial CFD, it is necessary to move to high-order accurate numerical methods. The ESFR family of schemes are ideal for resolving turbulent flows due to low numerical dissipation and high-order accurate representation of solution gradients at the small scales. Advanced subgrid-scale models have been implemented in HiFiLES for all element types, enabling simulation of turbulent flows over complex geometry. The development of the first high-order accurate solver for unstructured meshes incorporating LES modeling capabilities represents a significant step towards tackling challenging compressible turbulent flow problems of practical interest. Future additions will include optimization of the ESFR schemes for turbulence resolution, moving mesh capabilities, multigrid convergence acceleration, and advanced turbulence modeling.

Chapter 3

Flux Reconstruction Schemes with Corrected Fluxes Continuous in m Derivatives

3.1 Introduction

The FR approach to high-order methods provides a unifying framework to analyze and implement a large set of high-order schemes, including the nodal DG and SD methods. The unification occurs through the formulation of flux correction functions. The main appeal of FR is its differential formulation, which is ideal for highly-parallel computational architectures. ESFR provides the added benefit of guaranteeing linear stability while having variable dispersion and dissipation properties parameterized by a single constant. Asthana [4] found optimal values for such constant and found that the FR scheme could be optimized further if the scheme's formulation were not constrained by this parameter.

With the intention of providing a framework whereby more parameters are introduced while linear stability is guaranteed, we formulate the C^m Flux Reconstruction (CMFR) set of families of schemes. The main difference between ESFR and CMFR is that the flux correction functions in CMFR are forced to be continuous among elements in an arbitrary number of derivatives, while ESFR requires C^0 and C^{p+1}

continuity only –where p is the degree of the polynomial used to discretize the solution.

In this article we present the proof of linear stability of the CMFR set of families of schemes, the derivation of C1FR –a FR scheme with C^1 correction functions–, and promising results for energy preservation of underresolved wavenumbers with C1FR.

3.2 Background

High-order numerical methods for unstructured grids promise to offer better accuracy than low-order schemes for a comparable computational cost. Their relative lower dispersion and dissipation make them prime candidates for use in LES [50]. Below is a brief history on the developments in high-order numerical schemes that form the foundations for this chapter.

The work of Reed and Hill [71] in the 70’s introduced the DG method to solve PDEs in variational form. Cockburn and Shu formulated the DG method for conservation laws and advanced its theoretical foundations [22, 21, 20, 25, 24]. As a way to reduce the computational cost of the original DG scheme, researchers developed a nodal variant. Hesthaven and Warburton give a through exposition of this method in their book [34]. Kopriva and Kolias [43] developed a staggered grid method based on the differential form of the equations. This method was later named SD and was thoroughly studied by Liu et al. [47] and Wang et al. [92]. Wang [90] has also introduced the popular spectral volume method.

Noting the similarities between nodal DG and the SD schemes, Huynh introduced the FR approach to high-order methods [36, 37]. With this approach, it is possible to analyze and implement multiple high-order schemes within a unifying framework, including the SD method and a variant of nodal DG for the linear advection equation. Furthermore, Huynh used the FR approach to create a variety of new high-order schemes with different stability and accuracy properties. Vincent et al. [85], building on Jameson’s proof of stability of the SD scheme [38], formulated a class of ESFR schemes. These schemes are provably stable at all orders of accuracy for the linear advection-diffusion equations [15]. Williams et al. and Castonguay et al. have proved

this stability for all orders in tetrahedral [97] and triangular [18, 96] meshes.

In the $p+1$ th order FR and ESFR schemes for the 1-dimensional linear advection equation, the solution is represented by a polynomial of degree p and the flux is represented by a polynomial of degree $p+1$. In this article, we show that it is possible to represent the flux with a polynomial of degree m – where $m \geq p+1-$, evaluate such flux at the regular $p+1$ points, retain the provable linear stability, and obtain the expected $m+1$ th order convergence (limited only by interpolation errors). An important part of the proof consists of ensuring that the flux polynomial representations have continuous arbitrary derivatives at the flux points connecting interfaces between elements, hence we call these schemes C^m Flux Reconstruction (CMFR) schemes. This general framework recovers ESFR schemes and provides an arbitrary number of parameters that can be used to optimize the dispersion and dissipation properties of the schemes. The work by Asthana [4] shows how it is possible to optimize the dispersion and dissipation properties of ESFR schemes. His work inspired the formulation of a provably stable scheme with more optimizable parameters, and thus the creation of the CMFR schemes presented here.

The article starts with the formulation of the CMFR schemes. We then present the proof of stability for all orders of accuracy of the CMFR schemes for the linear advection equation. The proof of stability requires the formulation of some “correction functions”, so the discussion continues with the general procedure for finding these. To illustrate the process of scheme creation, we show the development of the CMFR scheme that has fluxes continuous in the 0th and 1st derivatives: the $C^{0,1}$ FR (or C1FR) scheme. We then perform numerical experiments that demonstrate the schemes’ $p+1$ th order convergence when we use an p th order polynomial to represent the flux. The discussion is followed by showing how the CMFR schemes can preserve the energy of high-frequency waves better than the nodal DG method at the price of exchanging flux derivative information across element interfaces and additional work in the simulation pre-processing stage. We conclude by showing potential avenues of future research.

3.3 The C^m Flux Reconstruction Approach

3.3.1 Preliminaries: the general advection equation

Suppose we would like to solve the one-dimensional conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad (3.1)$$

in the domain Ω . x is the spatial coordinate, t is time, $u = u(x, t)$ is the conserved scalar quantity (or solution), and $f = f(u)$ is the flux.

To discretize the equation, let us partition Ω into N non-overlapping elements $\Omega_n = \{x | x_n < x < x_{n+1}\}$ and approximate both the solution u and flux f within each Ω_n with polynomials. In each element, solution points are the locations where the solution values are stored and advanced; flux points in each element serve the same purpose for the flux values. In the Flux Reconstruction family of schemes, the solution and flux points are collocated in order to not have to compute additional flux interpolations.

As is customary, let us map the approximated solution and flux from the physical domain Ω_n in x -coordinates to the reference domain $\hat{\Omega} = \{\xi | -1 < \xi < 1\}$ in ξ -coordinates. We can then write the approximations in the following form

$$\hat{u} = \sum_{p=1}^{P+1} \hat{u}_p l_p(\xi) \quad (3.2)$$

$$\hat{f} = \sum_{p=1}^{P+1} \hat{f}(\hat{u}_p) l_p(\xi) \quad (3.3)$$

where P is the polynomial order used to represent the solution, l_p is the Lagrange polynomial that equals 1 at the solution/flux point p and 0 at the others, and \hat{u}_p is the solution value at the point p . Note that both u and f are potentially discontinuous across elements. The circumflex $\hat{\cdot}$ means that the polynomial or entity below it is written or defined in reference domain coordinates.

Understanding that $J_n = \frac{\partial x}{\partial \xi}|_n$, we can rewrite Eqn. (3.1) the reference domain

coordinates

$$\frac{\partial \hat{u}}{\partial t} + \frac{1}{J_n} \frac{\partial \hat{f}}{\partial \xi} = 0 \quad (3.4)$$

Let us clearly show the values of the desired m^{th} order derivatives at the left interfaces of the n^{th} element in both the physical and reference domains.

$$f|_{L,n}^I = \hat{f}|_{L,n}^I \quad (3.5)$$

$$\left. \frac{\partial f}{\partial x} \right|_{L,n}^I = \frac{1}{J_n} \left. \frac{\partial \hat{f}}{\partial \xi} \right|_{L,n}^I \quad (3.6)$$

$$\left. \frac{\partial^m f}{\partial x^m} \right|_{L,n}^I = \frac{1}{J_n^m} \left. \frac{\partial^m \hat{f}}{\partial \xi^m} \right|_{L,n}^I \quad (3.7)$$

Here the symbol $|_{n,L}^I$ denotes that the quantity to its left is being evaluated at the *left (L) interface (I)* of element n , and J_n represents the Jacobian of element n . Note that the desired interface values $\left. \frac{\partial^m \hat{f}}{\partial \xi^m} \right|_{L,n}^I$ and $\left. \frac{\partial^m \hat{f}}{\partial \xi^m} \right|_{R,n}^I$ will be defined later on when proving linear stability of the scheme.

It is important to note that

$$\left. \frac{\partial^m f}{\partial \xi^m} \right|_{R,n}^I = \left. \frac{\partial^m f}{\partial \xi^m} \right|_{L,n+1}^I \quad (3.8)$$

Eventually, we would like to add a polynomial to the flux in order to guarantee continuity of arbitrary derivatives across elements. To that end, let us define the following correction constants at the left interface of element n –the correction constants

at the right interface are defined in the same way by replacing L with R –

$$\begin{aligned}
 \text{for } c^0 \text{ continuity: } I_{0_{L,n}} &= f|_{L,n}^I - f|_{L,n} \\
 &= \hat{f}|_{L,n}^I - \hat{f}|_{L,n} \\
 &\vdots \\
 \text{for } c^m \text{ continuity: } I_{m_{L,n}} &= \frac{\partial^m f}{\partial x^m} \Big|_{L,n}^I - \frac{\partial^m f}{\partial x^m} \Big|_{L,n} \\
 &= \frac{1}{J_n^m} \left(\frac{\partial^m \hat{f}}{\partial \xi^m} \Big|_{L,n}^I - \frac{\partial^m \hat{f}}{\partial \xi^m} \Big|_{L,n} \right)
 \end{aligned} \tag{3.9}$$

These constants are the difference between the desired flux derivative and the derivative of the flux polynomial at the interface of interest.

We can now introduce the correction functions that will enforce flux derivative continuity across elements. To guarantee that we have full control over which derivatives will be continuous at both the left (L) and right (R) interfaces at each element, we set the following conditions on the correction functions $g_{L_i}(\xi)$ and $g_{R_i}(\xi)$ defined in element n as follows:

$$\begin{aligned}
 \frac{\partial^j g_{L_i}}{\partial x^j}(-1) &= \delta_{ij} ; \quad \frac{\partial^j g_{L_i}}{\partial x^j}(1) = 0 \\
 \frac{\partial^j g_{R_i}}{\partial x^j}(-1) &= 0 ; \quad \frac{\partial^j g_{R_i}}{\partial x^j}(1) = \delta_{ij}
 \end{aligned} \tag{3.10}$$

where δ_{ij} is the Kronecker delta. i and j belong to the set of derivatives in which we wish to have continuity. For example, if we desire flux continuity in the zeroth and third derivatives, $i, j \subset \{0, 3\}$. Note that the correction function polynomials must be of order greater than or equal to $2s$, where s is the number of derivative continuities desired, due to the existence of two constraints per correction function. In the previous example, $s = 2$.

Putting all the definitions together, we can now define the corrected flux in element n as

$$\hat{f}^c = \hat{f} + \sum_{i=0}^m \{ I_{i_{L,n}} g_{L_i} + I_{i_{R,n}} g_{R_i} \} \tag{3.11}$$

where m is the highest derivative in which continuity is desired. The superscript c is used to make it explicit that the quantity over which it appears has been made continuous across element interfaces. In the example, $m = 3$, $I_{1L,n} = I_{1R,n} = I_{2L,n} = I_{2R,n} = 0$. The semi-discrete form of the update step in element n is

$$\frac{d\hat{u}_p}{dt} = -\frac{1}{J_n} \left[\frac{\partial \hat{f}}{\partial \xi}(\xi_p) + \sum_{i=0}^m \left\{ I_{iL,n} \frac{\partial g_{L_i}}{\partial \xi}(\xi_p) + I_{iR,n} \frac{\partial g_{R_i}}{\partial \xi}(\xi_p) \right\} \right] \quad (3.12)$$

for $p = 1, \dots, P+1$. Note that the correction functions are being sampled at the same points as the flux and solution, so the fact that we are using polynomials of high orders as correction functions does not add computational complexity to the update step. In vector form,

$$\frac{d\vec{\hat{u}}}{dt} = -\frac{1}{J_n} \left[\vec{\frac{\partial \hat{f}}{\partial \xi}} + \sum_{i=0}^m \left\{ I_{iL,n} \vec{\frac{\partial g_{L_i}}{\partial \xi}} + I_{iR,n} \vec{\frac{\partial g_{R_i}}{\partial \xi}} \right\} \right] \quad (3.13)$$

Here we see more clearly that the scheme maintains the desired computational parallelizability of the Flux Reconstruction family of schemes, as the only element specific values are the scalars J_n , $I_{iR,n}$ and $I_{iL,n}$.

3.3.2 CMFR schemes for the general advection-diffusion equation

The extension of a FR scheme for the advection equation to the advection-diffusion equation is straightforward. If we want to solve

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial^2 h}{\partial x^2} = 0 \quad (3.14)$$

where f and h are functions of u , we can introduce an auxiliary variable q so the equation becomes

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{\partial q}{\partial x} &= 0 \\ q &= f + \frac{\partial h}{\partial x}\end{aligned}\tag{3.15}$$

Note that in the linear case, $h = -\beta u$ and $f = au$.

When solving any PDE, the CMFR schemes correct all functions dependent on u whose first derivatives need to be found. In this general advection-diffusion case, q and h need to be corrected following the form in Eqn. (3.11). The main difference between this and the advection equation CMFR scheme is that in this case there are two corrections necessary and the interface values $I_{i_{L,n}}, I_{i_{R,n}}$ used to correct q will be a function of $q = f + \frac{\partial h}{\partial x}$ and not just f . The correction functions found for Eqn. (3.11) remain exactly the same.

3.4 Linear stability of C^m continuous flux reconstruction ($\hat{f} = a\hat{u}$)

In this section we show that the CMFR schemes are stable in the 1-D linear advection equation in the following Sobolev-type norm:

$$\begin{aligned}||u||_m^2 &= \sum_{n=1}^N \int_{x_n}^{x_{n+1}} \left\{ \sum_{r=0}^m \frac{c_r}{2} \left(\frac{\partial^r u}{\partial x^r} \right)^2 \right\} dx \\ &= \sum_{n=1}^N \sum_{r=0}^m c_r \left(\frac{1}{J_n^{2r}} \right) \int_{-1}^1 \left\{ \frac{1}{2} \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \right\} J_n \cdot d\xi\end{aligned}\tag{3.16}$$

where c_r for $0 \leq r \leq m$ are arbitrary constants. It is possible to find ranges of values of each c_r for which Eqn. (3.16) is a norm. As shown later in section 3.6.2, negative values of c_r are possible given that u is a polynomial. To establish stability, we need

to show that

$$\frac{d}{dt} \|u\|_m^2 = \sum_{n=1}^N \sum_{r=0}^m c_r \left(\frac{1}{J_n^{2r}} \right) \frac{d}{dt} \int_{-1}^1 \left\{ \frac{1}{2} \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \right\} J_n \cdot d\xi \leq 0 \quad (3.17)$$

To that end, consider the FR scheme in element n for linear advection:

$$\frac{d\hat{u}}{dt} = -\frac{1}{J_n} \left[a \frac{\partial \hat{u}}{\partial \xi} + \sum_{i=0}^m \left\{ I_{i_{L,n}} \frac{\partial g_{L_i}}{\partial \xi} + I_{i_{R,n}} \frac{\partial g_{R_i}}{\partial \xi} \right\} \right] \quad (3.18)$$

To express Eqn. (3.16) in known terms, differentiate Eqn. (3.18) r times, where $0 \leq r \leq m$, with respect to ξ ; multiply by $\frac{\partial^r \hat{u}}{\partial \xi^r}$; and integrate from -1 to 1 to obtain

$$\frac{d}{dt} \int_{-1}^1 \left\{ \frac{1}{2} \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \right\} d\xi = -\frac{1}{J_n} [\textcircled{1} + \textcircled{2} + \textcircled{3}] \quad (3.19)$$

where

$$\textcircled{1} = a \int_{-1}^1 \frac{1}{2} \frac{\partial}{\partial \xi} \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 d\xi = \frac{a}{2} \left[\left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \Big|_{R,n} - \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \Big|_{L,n} \right] \quad (3.20)$$

$$\textcircled{2} = \sum_{i=0}^m I_{i_{L,n}} \int_{-1}^1 \frac{\partial^r \hat{u}}{\partial \xi^r} \cdot \frac{\partial^{r+1} g_{L_i}}{\partial \xi^{r+1}} d\xi \quad (3.21)$$

$$\textcircled{3} = \sum_{i=0}^m I_{i_{R,n}} \int_{-1}^1 \frac{\partial^r \hat{u}}{\partial \xi^r} \cdot \frac{\partial^{r+1} g_{R_i}}{\partial \xi^{r+1}} d\xi \quad (3.22)$$

It is possible to simplify Eqn. (3.21) further. Integrating by parts,

$$\textcircled{2} = \sum_{i=0}^m I_{i_{L,n}} \left\{ \int_{-1}^1 \left(\frac{\partial}{\partial \xi} \left[\frac{\partial^r \hat{u}}{\partial \xi^r} \frac{\partial^r g_{L_i}}{\partial \xi^r} \right] - \frac{\partial^r g_{L_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} \right) d\xi \right\} \quad (3.23)$$

$$\textcircled{2} = \sum_{i=0}^m I_{i_{L,n}} \left[\frac{\partial^r \hat{u}}{\partial \xi^r} \frac{\partial^r g_{L_i}}{\partial \xi^r} \right]_{-1}^1 - \sum_{i=0}^m I_{i_{L,n}} \int_{-1}^1 \frac{\partial^r g_{L_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} d\xi \quad (3.24)$$

By using the boundary conditions on g_{L_i} given in Eqn. (3.10),

$$\textcircled{2} = -I_{r_{L,n}} J_n^r \frac{\partial^r \hat{u}}{\partial \xi^r} \Big|_{L,n} - \sum_{i=0}^m I_{i_{L,n}} \int_{-1}^1 \frac{\partial^r g_{L_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} d\xi \quad (3.25)$$

Proceeding similarly with term $\textcircled{3}$

$$\textcircled{3} = I_{r_{R,n}} J_n^r \frac{\partial^r \hat{u}}{\partial \xi^r} \Big|_{R,n} - \sum_{i=0}^m I_{i_{R,n}} \int_{-1}^1 \frac{\partial^r g_{R_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} d\xi \quad (3.26)$$

Note the difference in signs between $\textcircled{2}$ and $\textcircled{3}$.

By replacing the expression from Eqn. (3.19) into Eqn. (3.17), we obtain

$$\begin{aligned} \frac{d}{dt} \|u\|_m^2 &= \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \left(-\frac{1}{J_n^{2r}} \right) \frac{a}{2} \left[\left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \Big|_{R,n} - \left(\frac{\partial^r \hat{u}}{\partial \xi^r} \right)^2 \Big|_{L,n} \right] \right. \\ &\quad + \sum_{r=0}^m c_r \left(-\frac{1}{J_n^{2r}} \right) \left[-I_{r_{L,n}} J_n^r \frac{\partial^r \hat{u}}{\partial \xi^r} \Big|_{L,n} - \sum_{i=0}^m I_{i_{L,n}} \int_{-1}^1 \frac{\partial^r g_{L_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} d\xi \right] \\ &\quad \left. + \sum_{r=0}^m c_r \left(-\frac{1}{J_n^{2r}} \right) \left[I_{r_{R,n}} J_n^r \frac{\partial^r \hat{u}}{\partial \xi^r} \Big|_{R,n} - \sum_{i=0}^m I_{i_{R,n}} \int_{-1}^1 \frac{\partial^r g_{R_i}}{\partial \xi^r} \frac{\partial^{r+1} \hat{u}}{\partial \xi^{r+1}} d\xi \right] \right\} \end{aligned} \quad (3.27)$$

To de-clutter the notation, let us define

$$\frac{\partial^r *}{\partial \xi^r} = *^{(r)}$$

and re-arrange Eqn. (3.27) to obtain

$$\frac{d}{dt} \|\hat{u}\|_m^2 = \textcircled{a} + \textcircled{b} + \textcircled{c} \quad (3.28)$$

where

$$\begin{aligned} \textcircled{a} &= - \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \frac{a}{2} \left[\hat{u}^{(r)2} \Big|_{R,n} - \hat{u}^{(r)2} \Big|_{L,n} \right] \left(\frac{1}{J_n^{2r}} \right) \right. \\ &\quad + \sum_{r=0}^m -c_r I_{r_{L,n}} J_n^r \hat{u}^{(r)} \Big|_{L,n} \left(\frac{1}{J_n^{2r}} \right) \\ &\quad \left. + \sum_{r=0}^m c_r I_{r_{R,n}} J_n^r \hat{u}^{(r)} \Big|_{R,n} \left(\frac{1}{J_n^{2r}} \right) \right\} \end{aligned} \quad (3.29)$$

$$\textcircled{b} = \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \sum_{i=0}^m I_{i_{L,n}} \int_{-1}^1 g_{L_i}^{(r)} \hat{u}^{(r+1)} d\xi \left(\frac{1}{J_n^{2r}} \right) \right\} \quad (3.30)$$

$$\textcircled{c} = \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \sum_{i=0}^m I_{i_{R,n}} \int_{-1}^1 g_{R_i}^{(r)} \hat{u}^{(r+1)} d\xi \left(\frac{1}{J_n^{2r}} \right) \right\} \quad (3.31)$$

We show stability of CMFR in the two following steps:

1. We show that for a selection of interface values $I_{i_{L,n}}$ and $I_{i_{R,n}}$,

$$\textcircled{a} \leq 0 \quad (3.32)$$

2. We explain how to find functions g_{L_i} and g_{R_i} , for $i = 0, \dots, m$ satisfying conditions (3.10) such that

$$\begin{aligned} \textcircled{b} &= 0 \\ \textcircled{c} &= 0 \end{aligned} \quad (3.33)$$

for arbitrary $c_r, r = 1, \dots, m$.

By showing that expressions (3.32) and (3.33) hold, we conclude, from Eqn. (3.28) that

$$\frac{d}{dt} \|\hat{u}\|_m \leq 0 \quad (3.34)$$

3.4.1 Part 1.

In this part of the proof, we aim to show that the term ④ in Eqn. (3.28) is non-positive.

Rearranging and factoring terms in ④, Eqn. (3.29) becomes

$$\begin{aligned} \textcircled{4} = & - \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \left(\frac{1}{J_n^{2r}} \right) \left(\frac{a}{2} \left[\hat{u}^{(r)2} \Big|_{R,n} - \hat{u}^{(r)2} \Big|_{L,n} \right] \right. \right. \\ & \left. \left. - I_{r_{L,n}} J_n^r \cdot \hat{u}^{(r)} \Big|_{L,n} + I_{r_{R,n}} J_n^r \cdot \hat{u}^{(r)} \Big|_{R,n} \right) \right\} \end{aligned} \quad (3.35)$$

Recall the definition of the correction constants $I_{r_{L,n}}$ and $I_{r_{R,n}}$ in element n from Eqn. (3.9),

$$\begin{aligned} I_{r_{L,n}} J_n^r &= f^{(r)} \Big|_{L,n}^I - f^{(r)} \Big|_{L,n} \\ I_{r_{R,n}} J_n^r &= f^{(r)} \Big|_{R,n}^I - f^{(r)} \Big|_{R,n} \end{aligned} \quad (3.36)$$

In the case of linear advection, $\hat{f}^\delta = a\hat{u}$, so the correction constants become

$$\begin{aligned} I_{r_{L,n}} J_n^r &= \hat{f}^{(r)} \Big|_{L,n}^I - a\hat{u}^{(r)} \Big|_{L,n} \\ I_{r_{R,n}} J_n^r &= \hat{f}^{(r)} \Big|_{R,n}^I - a\hat{u}^{(r)} \Big|_{R,n} \end{aligned} \quad (3.37)$$

Let us introduce the following generalized Roe flux at the interfaces, so

$$\begin{aligned} \hat{f}^{(r)} \Big|_{L,n}^I &= \frac{1}{2} \left[a\hat{u}^{(r)} \Big|_{L,n} + a\hat{u}^{(r)} \Big|_{R,n-1} \right] \\ &\quad - \frac{1-\alpha_r}{2} |A_{r_{L,n}}| \left[\hat{u}^{(r)} \Big|_{L,n} - \hat{u}^{(r)} \Big|_{R,n-1} \right] \\ \hat{f}^{(r)} \Big|_{R,n}^I &= \frac{1}{2} \left[a\hat{u}^{(r)} \Big|_{L,n+1} + a\hat{u}^{(r)} \Big|_{R,n} \right] \\ &\quad - \frac{1-\alpha_r}{2} |A_{r_{R,n}}| \left[\hat{u}^{(r)} \Big|_{L,n+1} - \hat{u}^{(r)} \Big|_{R,n} \right] \end{aligned} \quad (3.38)$$

Where $A_{r_{L,n}}$ and $A_{r_{R,n}}$ are the Jacobian matrices corresponding to the r^{th} derivative of the flux at the left (L) and right (R) interfaces of element n . Equivalently,

$$\begin{aligned} A_{r_{L,n}} &= \left[\frac{d(\hat{f}^{(r)})}{d(\hat{u}^{(r)})} \right]_{L,n} \\ A_{r_{R,n}} &= \left[\frac{d(\hat{f}^{(r)})}{d(\hat{u}^{(r)})} \right]_{R,n} \end{aligned} \quad (3.39)$$

Numerically, $A_{r_{L,n}}$ and $A_{r_{R,n}}$ can be evaluated for the linear advection equation as

$$\begin{aligned} A_{r_{L,n}} &= \frac{a\hat{u}^{(r)}|_{L,n} - a\hat{u}^{(r)}|_{R,n-1}}{\hat{u}^{(r)}|_{L,n} - \hat{u}^{(r)}|_{R,n-1}} = a \\ A_{r_{R,n}} &= \frac{a\hat{u}^{(r)}|_{L,n+1} - a\hat{u}^{(r)}|_{R,n}}{\hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n}} = a \end{aligned} \quad (3.40)$$

Note that $A_{r_{L,n}} = A_{r_{R,n-1}}$ even for non-linear fluxes by construction.

Plugging these values of $A_{r_{L,n}}$ and $A_{r_{R,n}}$ into Eqn. (3.38), and substituting the updated interface flux values $\hat{f}^{(r)}|_{L,n}^I$ and $\hat{f}^{(r)}|_{R,n}^I$ into the definition of the interface correction values in Eqn. (3.37), we obtain

$$\begin{aligned} I_{r_{L,n}} J_n^r &= \frac{1}{2} \left\{ a\hat{u}^{(r)}|_{L,n} + a\hat{u}^{(r)}|_{R,n-1} \right\} \\ &\quad - \frac{1 - \alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n} - \hat{u}^{(r)}|_{R,n-1} \right\} - a\hat{u}^{(r)}|_{L,n} \\ I_{r_{R,n}} J_n^r &= \frac{1}{2} \left\{ a\hat{u}^{(r)}|_{L,n+1} + a\hat{u}^{(r)}|_{R,n} \right\} \\ &\quad - \frac{1 - \alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n} \right\} - a\hat{u}^{(r)}|_{R,n} \end{aligned} \quad (3.41)$$

Simplifying,

$$\begin{aligned} I_{r_{L,n}} J_n^r &= \frac{a}{2} \left\{ -\hat{u}^{(r)}|_{L,n} + \hat{u}^{(r)}|_{R,n-1} \right\} \\ &\quad - \frac{1-\alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n} - \hat{u}^{(r)}|_{R,n-1} \right\} \\ I_{r_{R,n}} J_n^r &= \frac{a}{2} \left\{ \hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n} \right\} \\ &\quad - \frac{1-\alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n} \right\} \end{aligned} \quad (3.42)$$

Using the updated values of $I_{r_{L,n}}$ and $I_{r_{R,n}}$ from Eqn. (3.42), Eqn. (3.35) becomes

$$\begin{aligned} @ = - \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \left(\frac{1}{J_n^{2r}} \right) \left(\frac{a}{2} \left[\hat{u}^{(r)2}|_{R,n} - \hat{u}^{(r)2}|_{L,n} \right] \right. \right. \\ &\quad - \left[\frac{a}{2} \left\{ -\hat{u}^{(r)}|_{L,n} + \hat{u}^{(r)}|_{R,n-1} \right\} \right. \\ &\quad \left. \left. - \frac{1-\alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n} - \hat{u}^{(r)}|_{R,n-1} \right\} \right] \cdot \hat{u}^{(r)}|_{L,n} \right. \\ &\quad + \left[\frac{a}{2} \left\{ \hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n} \right\} \right. \\ &\quad \left. \left. - \frac{1-\alpha_r}{2} |a| \left\{ \hat{u}^{(r)}|_{L,n+1} - \hat{u}^{(r)}|_{R,n} \right\} \right] \cdot \hat{u}^{(r)}|_{R,n} \right\} \end{aligned} \quad (3.43)$$

Distributing the $\left(\frac{1}{J_n^{2r}} \right)$ term to convert the derivatives with respect to ξ into

derivatives with respect to x , and factoring out the $\frac{a}{2}$ term,

$$\begin{aligned} \textcircled{a} = & - \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \frac{a}{2} \left(\left[u^{(r)2} \Big|_{R,n} - u^{(r)2} \Big|_{L,n} \right] \right. \right. \\ & - \left[\left\{ -u^{(r)} \Big|_{L,n} + u^{(r)} \Big|_{R,n-1} \right\} \right. \\ & - \frac{1 - \alpha_r}{a} |a| \left\{ u^{(r)} \Big|_{L,n} - u^{(r)} \Big|_{R,n-1} \right\} \left. \right] \cdot u^{(r)} \Big|_{L,n} \\ & + \left[\left\{ u^{(r)} \Big|_{L,n+1} - u^{(r)} \Big|_{R,n} \right\} \right. \\ & \left. \left. - \frac{1 - \alpha_r}{a} |a| \left\{ u^{(r)} \Big|_{L,n+1} - u^{(r)} \Big|_{R,n} \right\} \right] \cdot u^{(r)} \Big|_{R,n} \right\} \end{aligned} \quad (3.44)$$

Note that all terms in Eqn. (3.44) are defined at element interfaces. More explicitly,

$$\begin{aligned} u^{(r)} \Big|_{L,n} &= u_n(x_n) \\ u^{(r)} \Big|_{R,n} &= u_n(x_{n+1}) \end{aligned} \quad (3.45)$$

where u_n is the polynomial representing the solution in element n and x_n is the location of the n^{th} interface in physical coordinates. Using the identities in Eqn. (3.45), Eqn. (3.44) becomes

$$\begin{aligned} \textcircled{a} = & - \sum_{n=1}^N \left\{ \sum_{r=0}^m c_r \frac{a}{2} \left(\left[\left\{ u_n^{(r)}(x_{n+1}) \right\}^2 - \left\{ u_n^{(r)}(x_n) \right\}^2 \right] \right. \right. \\ & - \left[\left\{ -u_n^{(r)}(x_n) + u_{n-1}^{(r)}(x_n) \right\} \right. \\ & - \frac{1 - \alpha_r}{a} |a| \left\{ u_n^{(r)}(x_n) - u_{n-1}^{(r)}(x_n) \right\} \left. \right] \cdot u_n^{(r)}(x_n) \\ & + \left[\left\{ u_{n+1}^{(r)}(x_{n+1}) - u_n^{(r)}(x_{n+1}) \right\} \right. \\ & \left. \left. - \frac{1 - \alpha_r}{a} |a| \left\{ u_{n+1}^{(r)}(x_{n+1}) - u_n^{(r)}(x_{n+1}) \right\} \right] \cdot u_n^{(r)}(x_{n+1}) \right\} \end{aligned} \quad (3.46)$$

Let us do the following substitutions to simplify algebraic manipulations

$$\begin{aligned} B_n = \sum_{r=0}^m c_r \frac{a}{2} & \left(\left[u_{n-1}^{(r)}(x_n) \right]^2 \right. \\ & + \left[\left\{ u_n^{(r)}(x_n) - u_{n-1}^{(r)}(x_n) \right\} \right. \\ & \left. \left. - \frac{1-\alpha_r}{a} |a| \left\{ u_n^{(r)}(x_n) - u_{n-1}^{(r)}(x_n) \right\} \right] \cdot u_{n-1}^{(r)}(x_n) \right) \end{aligned} \quad (3.47)$$

$$\begin{aligned} D_n = \sum_{r=0}^m c_r \frac{a}{2} & \left(- \left[u_n^{(r)}(x_n) \right]^2 \right. \\ & - \left[\left\{ -u_n^{(r)}(x_n) + u_{n-1}^{(r)}(x_n) \right\} \right. \\ & \left. \left. - \frac{1-\alpha_r}{a} |a| \left\{ u_n^{(r)}(x_n) - u_{n-1}^{(r)}(x_n) \right\} \right] \cdot u_n^{(r)}(x_n) \right) \end{aligned} \quad (3.48)$$

We can then rewrite Eqn. (3.46) as

$$\textcircled{a} = - \sum_{n=1}^N \{B_{n+1} + D_n\} \quad (3.49)$$

Let us manipulate (3.49) to combine the two summations into one whose terms have the same unshifted index

$$\begin{aligned} \textcircled{a} &= - \sum_{n=1}^N D_n - \sum_{n=1}^N B_{n+1} \\ \textcircled{a} &= - \sum_{n=1}^N D_n - \sum_{n=2}^{N+1} B_n \\ \textcircled{a} &= -D_1 - \sum_{n=2}^N D_n - \sum_{n=2}^N B_n - B_{N+1} \\ \textcircled{a} &= -D_1 - \sum_{n=2}^N \{D_n + B_n\} - B_{N+1} \end{aligned} \quad (3.50)$$

Note that various terms in B_n and D_n cancel when summed (compare (3.47) and (3.48)). After doing such cancellations, Eqn. (3.46) becomes

$$\textcircled{a} = -D_1 - \sum_{n=2}^N \sum_{r=0}^m c_r \cdot \frac{1-\alpha_r}{2} |a| \left(u_{n-1}^{(r)}(x_n) - u_n^{(r)}(x_n) \right)^2 - B_{N+1} \quad (3.51)$$

The value of the solution and its derivatives at x_1 and x_{N+1} are set by the desired boundary conditions. Both D_1 and B_{N+1} depend exclusively on such pre-determined conditions:

$$D_1 = - \sum_{r=0}^m c_r \frac{a}{2} \left[u_1^{(r)}(x_1) \right]^2 \quad (3.52)$$

$$B_{N+1} = \sum_{r=0}^m c_r \frac{a}{2} \left[u_N^{(r)}(x_{N+1}) \right]^2 \quad (3.53)$$

To not introduce/extract energy into/from the solution, let us set periodic boundary conditions in all derivatives,

$$u_1^{(r)}(x_1) = u_N^{(r)}(x_{N+1}) \quad (3.54)$$

Consequently, $D_1 + B_{N+1} = 0$, and Eqn. (3.46) becomes simply

$$\textcircled{a} = - \sum_{n=2}^N \sum_{r=0}^m c_r \cdot \frac{1-\alpha_r}{2} |a| \left(u_{n-1}^{(r)}(x_n) - u_n^{(r)}(x_n) \right)^2 \quad (3.55)$$

Knowing that the following holds,

$$\begin{aligned} c_r &\geq 0 \\ 0 &\leq \alpha_r \leq 1 \\ \left(u_{n-1}^{(r)}(x_n) - u_n^{(r)}(x_n) \right)^2 &\geq 0 \text{ for } 2 \leq n \leq N \end{aligned} \quad (3.56)$$

we conclude that

$$\textcircled{a} \leq 0 \quad (3.57)$$

We have shown that the term \textcircled{a} in Eqn. (3.28) is non-positive, concluding this part of the proof.

3.4.2 Part 2.

We wish to find functions g_{L_i} and g_{R_i} , for $i = 0, \dots, m$, which satisfy the boundary conditions in Eqn. (3.10) and the stability conditions of Eqn. (3.33) ($\textcircled{b} = 0$ and $\textcircled{c} = 0$) for arbitrary $c_r, r = 1, \dots, m$.

We start by finding g_{L_i} . Let us manipulate \textcircled{b} in Eqn. (3.30) to find restrictions on g_{L_i} ,

$$\begin{aligned}\textcircled{b} &= \sum_{n=1}^N \left\{ \sum_{r=0}^m \left[c_r \left(\frac{1}{J_n^{2r}} \right) \sum_{i=0}^m \left(I_{i_{L,n}} \int_{-1}^1 g_{L_i}^{(r)} \hat{u}^{(r+1)} d\xi \right) \right] \right\} = 0 \\ &= \sum_{n=1}^N \left\{ \sum_{i=0}^m \left[I_{i_{L,n}} \sum_{r=0}^m \left(c_r \frac{1}{J_n^{2r}} \int_{-1}^1 g_{L_i}^{(r)} \hat{u}^{(r+1)} d\xi \right) \right] \right\} = 0\end{aligned}\tag{3.58}$$

To satisfy Eqn. (3.58), either

$$\sum_{i=0}^m \left(I_{i_{L,n}} \int_{-1}^1 g_{L_i}^{(r)} \hat{u}^{(r+1)} d\xi \right) = 0\tag{3.59}$$

or

$$\sum_{r=0}^m \left(c_r \frac{1}{J_n^{2r}} \int_{-1}^1 g_{L_i}^{(r)} \hat{u}^{(r+1)} d\xi \right) = 0\tag{3.60}$$

We then have two options for finding g_{L_i} . Observing that the only term in the summand in (3.60) that changes as the solution evolves is \hat{u} itself, we realize that a generic g_{L_i} with special polynomial orthogonality properties would satisfy Eqn. (3.58). As a result, we could pre-compute a non-changing g_{L_i} to run a linearly stable scheme. On the other hand, the option given by (3.59) obliges us to find g_{L_i} at every time-step because the scalar $I_{i_{L,n}}$ changes with the solution. Therefore, we choose to find g_{L_i} using (3.60).

Rewriting \hat{u} as a sum of scaled Lagrange polynomials, condition (3.60) becomes

$$\sum_{p=1}^{P+1} \hat{u}_p \sum_{r=0}^m \frac{c_r}{J_n^{2r}} \left(\int_{-1}^1 g_{L_i}^{(r)} l_p^{(r+1)} d\xi \right) = 0 \quad (3.61)$$

This implies that

$$\sum_{r=0}^m \frac{c_r}{J_n^{2r}} \int_{-1}^1 g_{L_i}^{(r)} l_p^{(r+1)} d\xi = 0 \quad (3.62)$$

for $p = 1, \dots, P + 1$, (recall P is the order of the polynomial used to represent \hat{u}). Let us expand the Lagrange polynomials l_p and the correction functions g_{L_i} into monomials,

$$\begin{aligned} l_p &= \sum_{j=0}^P \zeta_{pj} \xi^j \\ g_{L_i} &= \sum_{k=0}^S \theta_{ik} \xi^k \end{aligned} \quad (3.63)$$

Note that ζ_{pj} are known, unchanging scalars while θ_{ik} are the unknowns we are trying to find. S is the desired polynomial order of the correction function.

Eqn. (3.62) becomes

$$\sum_{r=0}^m \frac{c_r}{J_n^{2r}} \int_{-1}^1 \left(\sum_{k=r}^S \theta_{ik} \frac{k!}{(k-r)!} \xi^{k-r} \right) \left(\sum_{j=r+1}^P \zeta_{pj} \frac{j!}{(j-r-1)!} \xi^{j-r-1} \right) d\xi = 0$$

After some algebraic manipulation, we arrive at the conditions that each g_{L_i} must satisfy so the CMFR scheme maintains linear stability:

$$\sum_{r=0}^m \frac{c_r}{J_n^{2r}} \left[\sum_{k=r}^S \theta_{ik} \frac{k!}{(k-r)!} \left(\sum_{j=r+1}^P \zeta_{pj} \frac{j!}{(j-r-1)!} \int_{-1}^1 \xi^{j+k-2r-1} d\xi \right) \right] = 0 \quad (3.64)$$

Recall that i in θ_{ik} indexes the correction function corresponding to each specific derivative in which continuity is desired. p indexes each solution point in the reference domain.

With Eqn. (3.64) and the constraints given in Eqn. (3.10) we can construct a system of equations to solve for g_{L_i} in each element. More specifically, if we would like to ensure s different flux continuities between elements, the boundary constraints in Eqn. (3.10) produce $2s$ equations, and the conditions in Eqn. (3.64) produce P equations (not necessarily all independent), where P is the order of the solution representation \hat{u} .

We can recover the ESFR scheme by ensuring continuity in the 0th and $P + 1$ th derivatives. These two desired flux continuities ($s = 2$) force the flux to be of order, at least, $P + 1$. This is the order of the correction functions suggested by Vincent [85]. The ESFR scheme is one of the cases in which Eqn. (3.62) produces $P - 1$ independent equations instead of P . This can be seen by the fact that the $P + 1$ th and $P + 2$ th derivatives of a Lagrange polynomial l_p of order P is zero.

Once we find g_{L_i} , we find g_{R_i} automatically by setting

$$g_{R_i} = (-1)^i g_{L_i}(-\xi) \quad (3.65)$$

3.5 C^1 linearly stable flux reconstruction family of schemes

In this section, we illustrate the process of creating a scheme that ensures C^1 flux continuity to solve the advection and advection-diffusion equation. This scheme is called C01FR, or C1FR. We use this notation because it is possible to create a scheme whose reconstructed flux is continuous in the, say, zeroth and third derivatives without enforcing continuity in the first and second derivatives, and such scheme would have a $C^{0,3}$ flux and be called C03FR. When a scheme ensures continuity in the $0, \dots, M$ derivatives of the flux, we call it CMFR to simplify the name, which could as well be C012...MFR.

3.5.1 C1FR for the linear advection equation

To create the scheme with C^1 flux, we only need to find correction functions g_{L_i} that satisfy (3.64) and boundary conditions (3.10). Let c_0 and c_1 be non-zero in Eqn. (3.64). Expanding the summation over r , we have the following $P + 1$ equations for each i

$$\begin{aligned} c_0 \left[\sum_{k=0}^S \theta_{ik} \left(\sum_{j=1}^P \zeta_{pj} \cdot j \cdot \int_{-1}^1 \xi^{j+k-1} d\xi \right) \right] \\ + \frac{c_1}{J_n^2} \left[\sum_{k=1}^S k \theta_{ik} \left(\sum_{j=2}^P \zeta_{pj} \cdot j \cdot (j-1) \cdot \int_{-1}^1 \xi^{j+k-3} d\xi \right) \right] = 0 \end{aligned} \quad (3.66)$$

for $p = 1, \dots, P + 1$.

Rearranging terms we obtain

$$\begin{aligned} \sum_{j=1}^P \zeta_{pj} \left[\sum_{k=0}^S \theta_{ik} \left(c_0 \cdot j \int_{-1}^1 \xi^{j+k-1} d\xi \right. \right. \\ \left. \left. + \frac{c_1}{J_n^2} k j \cdot (j-1) \int_{-1}^1 \xi^{j+k-3} d\xi \right) \right] = 0 \end{aligned} \quad (3.67)$$

and, therefore we have the following P equations, where $j = 1, \dots, P$,

$$\sum_{k=0}^S \theta_{ik} \left(c_0 \cdot j \int_{-1}^1 \xi^{j+k-1} d\xi + \frac{c_1}{J_n^2} k j \cdot (j-1) \int_{-1}^1 \xi^{j+k-3} d\xi \right) = 0 \quad (3.68)$$

We can evaluate the integral terms understanding that for positive integer q

$$\int_{-1}^1 \xi^q d\xi = \begin{cases} \frac{2}{q+1} & , q \text{ even} \\ 0 & , q \text{ odd} \end{cases} \quad (3.69)$$

Recalling the constraints on g_{L_i} ,

$$g_{L_i}^{(j)}(-1) = \delta_{ij} J_n^j ; \quad g_{L_i}^{(j)}(+1) = 0 \quad (3.70)$$

we can obtain additional equations for each θ_{ik} . Specifically, for g_{L_0} ,

$$\begin{aligned} \sum_{k=0}^S \theta_{0k}(-1)^k &= 1 & \sum_{k=0}^S \theta_{0k}(1)^k &= 0 \\ \sum_{k=1}^S k\theta_{0k}(-1)^{k-1} &= 0 & \sum_{k=1}^S k\theta_{0k}(1)^{k-1} &= 0 \end{aligned} \quad (3.71)$$

and for g_{L_1} ,

$$\begin{aligned} \sum_{k=0}^S \theta_{1k}(-1)^k &= 0 & \sum_{k=0}^S \theta_{1k}(1)^k &= 0 \\ \sum_{k=1}^S k\theta_{1k}(-1)^{k-1} &= J_n & \sum_{k=1}^S k\theta_{1k}(1)^{k-1} &= 0 \end{aligned} \quad (3.72)$$

From Eqn. (3.65), we can find g_{R_i} by setting

$$g_{R_i} = (-1)^i \sum_{k=0}^S \theta_{ik}(-\xi)^k = \sum_{k=0}^S \theta_{ik}(-1)^{i+k} \xi^k \quad (3.73)$$

As a result, for each g_{L_i} we have $P + 4$ equations. By setting S , the order of the correction function, to be $P + 3$, we have $P + 4$ unknowns and can solve a $(P + 4) \times (P + 4)$ system of equations to have a one-parameter, linearly stable, C^1 flux, FR scheme.

Letting the polynomial order P be 3, we obtain the correction functions seen in Fig. 3.1 for varying values of c_1 . We let $c_0 = 1$ without loss of generality (as seen by the fact that in Eqn. (3.68) only the ratio c_1/c_0 modifies the correction functions). We note that the correction functions that ensure the flux is continuous in its first derivative, namely g_{L_1} and g_{R_1} are unaffected by changes in c_1 . On the other hand, g_{L_0} and g_{R_0} change dramatically with changes in c_1 . In section 3.6 we will see how these variations affect dispersion and dissipation properties qualitatively.

In Fig. 3.2 we see the correction functions corresponding to the C012FR, or C2FR, scheme with $P = 3$. It is interesting to note that changes in c_1 affect g_{L_0} and g_{R_0}

only, changes in c_2 affect g_{L_1} and g_{R_1} only, and g_{L_2} and g_{R_2} are not affected by c_1 or c_2 .

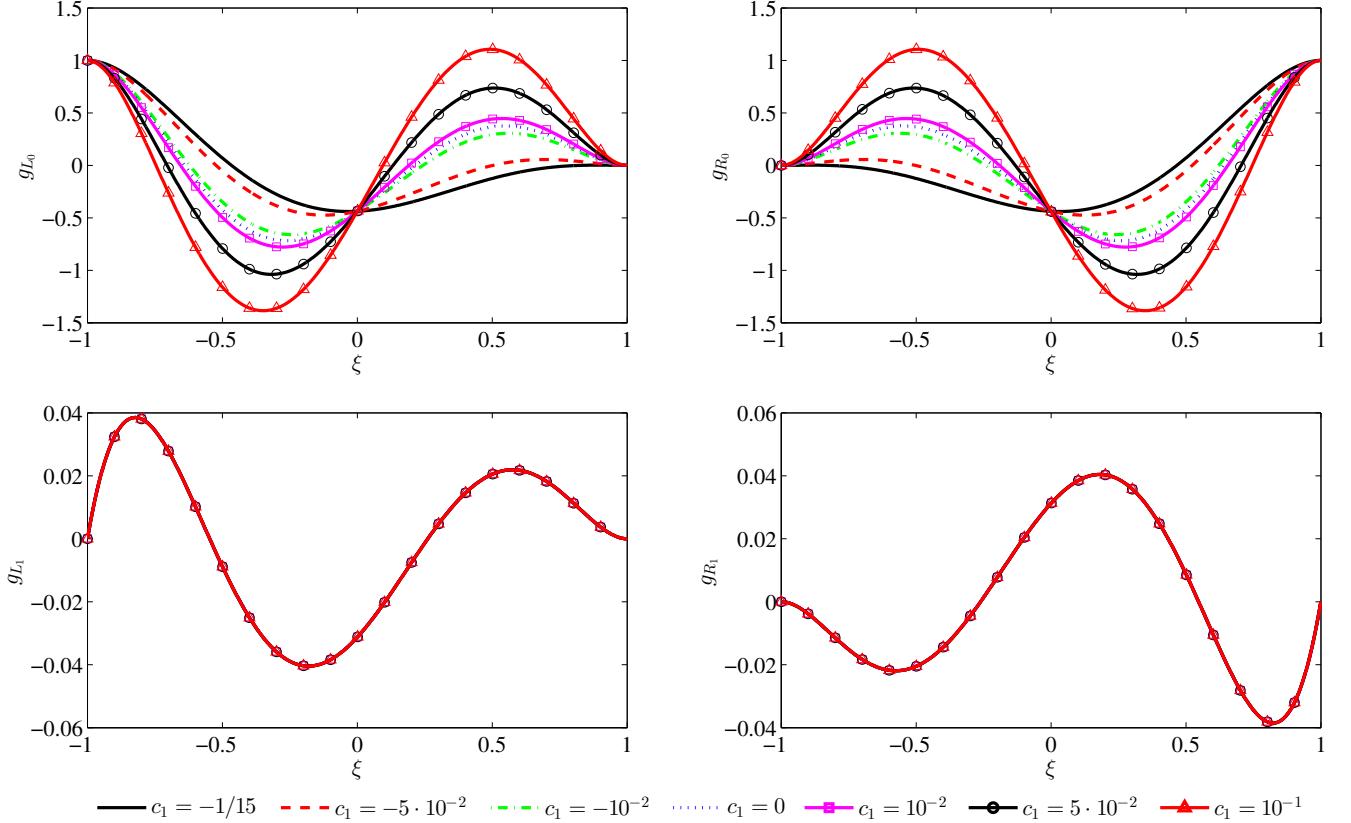


Figure 3.1: Left and right correction functions for the C1FR scheme with $P = 3$, in which the zeroth and first derivatives of the corrected flux are continuous

3.6 Numerical test cases

In this section we present solutions to the linear advection equation using the C1FR scheme to show that it is stable and achieves the theoretical order of convergence of $P + 1$ when the solution is discretized with an order P polynomial. In addition, we present solutions to the linear advection-diffusion equation to demonstrate the ability to change the scheme's dispersion and dissipation properties while maintaining

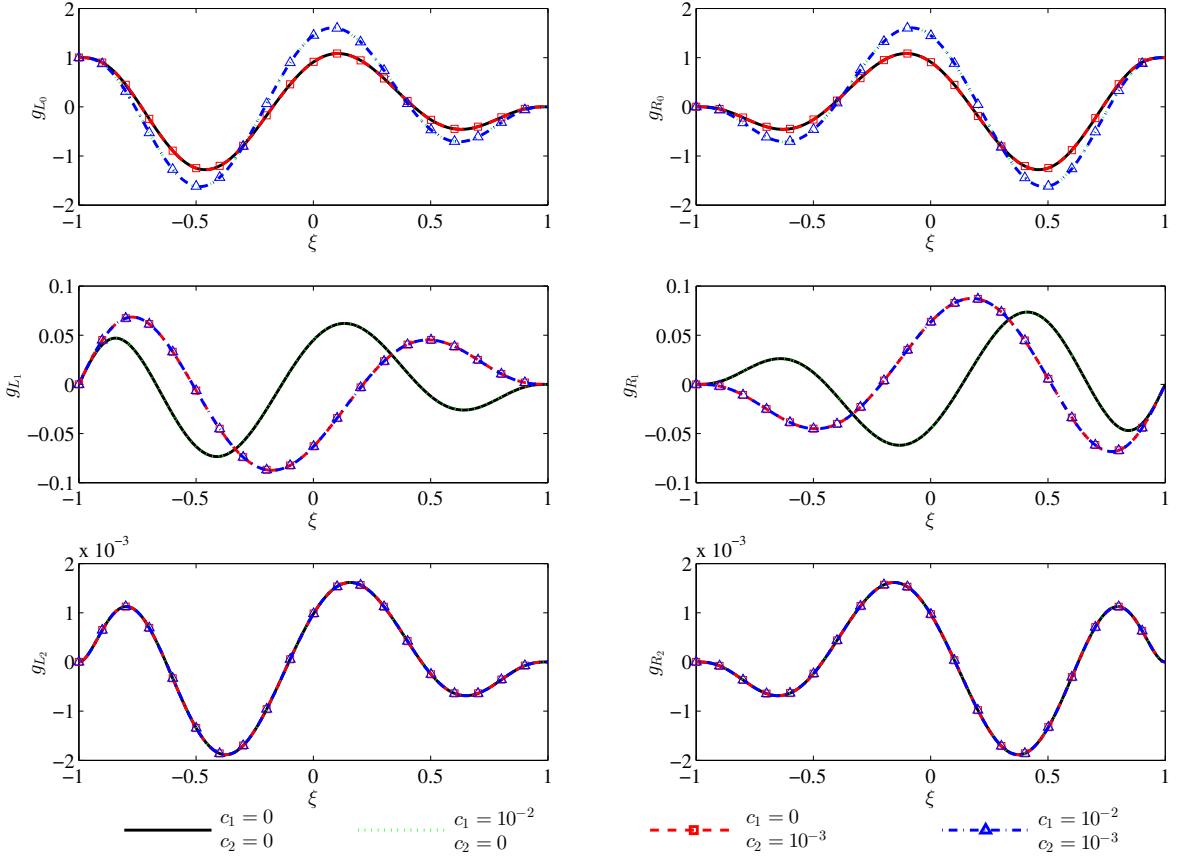


Figure 3.2: Left and right correction functions for the C2FR scheme with $P = 3$, in which the zeroth, first, and second derivatives of the corrected flux are continuous

stability. As a test of the scheme's ability to solve non-linear systems of equations, solutions to Sod's Shock Tube problem [72] and Einfeldt et al.'s 123 Problem [28] are presented. The solution of these challenging, difficult to stabilize initial conditions of the Euler equations would be a testament to the CMFR's promise as a numerical scheme with built-in stability.

As can be seen from the derivation of the C1FR scheme, the interface flux constants α_0 and α_1 , the norm constants c_0 and c_1 , and the location of the solution points at each element are variable. In this exposition, we will not modify the location of the solution points and use the standard zeroes of the Legendre polynomials. We note

that the values of c_1 have a direct impact on the scheme's dispersion and dissipation, while –as expected– the α_r values affect the dissipation only. A future rigorous Fourier analysis would reveal wiser choices for the c_1 parameter.

In the case of the Euler equations, the solution points in an element are located at the zeroes of the Legendre polynomials plus the end-points. It was found that the absence of solution points at, or very close to, the edges of an element would destabilize the scheme. This is to be expected, given that extrapolation of a polynomial is prone to large over and under-shoots of the real values.

3.6.1 Order of Accuracy of C1FR

3.6.1.1 Setup

The 1-D experiments follow the procedure suggested by Vincent et al. [84] to estimate a scheme's order of accuracy isolating interpolation errors. We solve the linear advection equation with advection speed of $a = 1$. The domain was $\Omega = [-10, 10]$ and was discretized in $n = 10, 15, 24, 38, 60$ equispaced elements of orders $P = 1, 2, 3$. The initial condition was a sine wave with wavenumber $k = 2\pi/20 \approx 0.63$. The advection speed was 1 and fully upwinded fluxes $\alpha_0 = 0, \alpha_1 = 0$ in Eqn. (3.38)) were used. The boundary conditions were periodic. The simulation advanced using a fourth order RK scheme with a time-step of order 10^{-3} .

The initial condition was advected for a full domain length, using either standard nodal DG or C1FR, and the resulting solution was taken as the reference solution u_{ref} . The wave was advected for a further full domain length to obtain the final solution u_{final} . The error was calculated by taking the L-2 norm of $u_{ref} - u_{final}$.

3.6.1.2 Results and discussion

Figures 3.3, 3.4, and 3.5 show the rate of convergence of the solution and its derivative obtained by discretizing the solution with polynomials of order $P = 1, 2, 3$, respectively. The slopes of the best fit lines are presented in each figure's caption.

The fact that we recover the expected nodal DG's $2P + 1$ order of convergence found by Vincent et al. [84] for $P = 1, 2, 3$ validates the experimental setup. It is

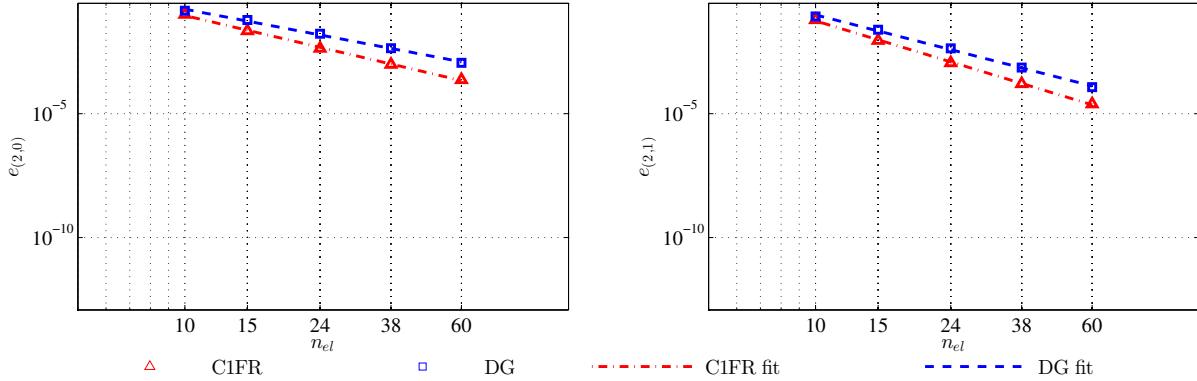


Figure 3.3: L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 1$. Order of accuracy in solution: DG: 2.728, C1FR: 3.374. Order of accuracy in first derivative: DG: 2.691, C1FR: 3.359.

interesting to note that C1FR retains FR's even-odd order of convergence behavior: when P is odd, the order of convergence is $2P + 1$; while when P is even, the order of convergence is $2P$.

This numerical experiment does not replace a von Neumann analysis, but does show that the scheme is stable, consistent, and maintains the desired order of accuracy. Although we would not expect the scheme to maintain super-convergence properties in real applications –as the interpolation errors are themselves of order $P + 1$ –, this experiment relieves worries about C1FR's introducing lower order errors.

3.6.2 Advection-Diffusion Energy Preservation

Motivated by the fact that in turbulent simulations the preservation of energy at different scales (or wavenumbers) is of paramount importance, we wanted to explore the potential benefit of having sets of families of stable numerical schemes with modifiable dispersion and dissipation properties.

By solving the linear advection-diffusion equation we are able to assess how much dissipation in different scales is due to numerics as opposed to the nature of the equation.

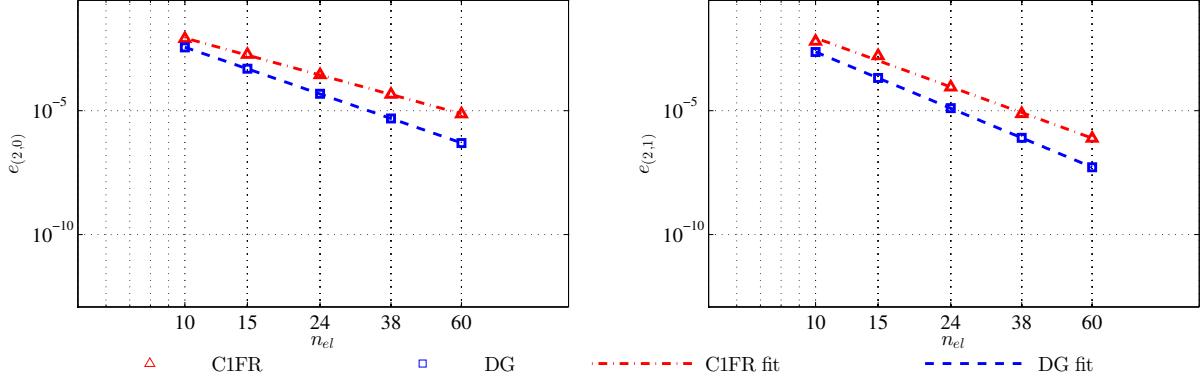


Figure 3.4: L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 2$. Order of accuracy in solution: DG: 4.960, C1FR: 3.917 . Order of accuracy in first derivative: DG: 4.971, C1FR: 4.178.

3.6.2.1 Setup

In these numerical experiments we solve the linear advection-diffusion equation using the C1FR and nodal DG schemes following the approach described by Huynh [37]. In essence, we re-write the diffusion-advection equation as a system of two first order PDEs as follows

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial q}{\partial x} &= 0 \\ q - au + \kappa \frac{\partial u}{\partial x} &= 0 \end{aligned} \tag{3.74}$$

where a is the advection speed, κ is the diffusion coefficient, and q is a dummy variable. The desired scheme is used to discretize the spatial differentiation.

In this section, we let $a = 1$, $\kappa = 10^{-2}$. The domain was $\Omega = [-10, 10]$ and was discretized in $n = 20$ equispaced elements of polynomial order $P = 5$. The boundary conditions were periodic. The initial conditions were sine waves with low, medium, and high wavenumbers. The wavenumbers were chosen relative to the Nyquist limit of the discretization:

$$k = \rho(P+1)\pi/h$$

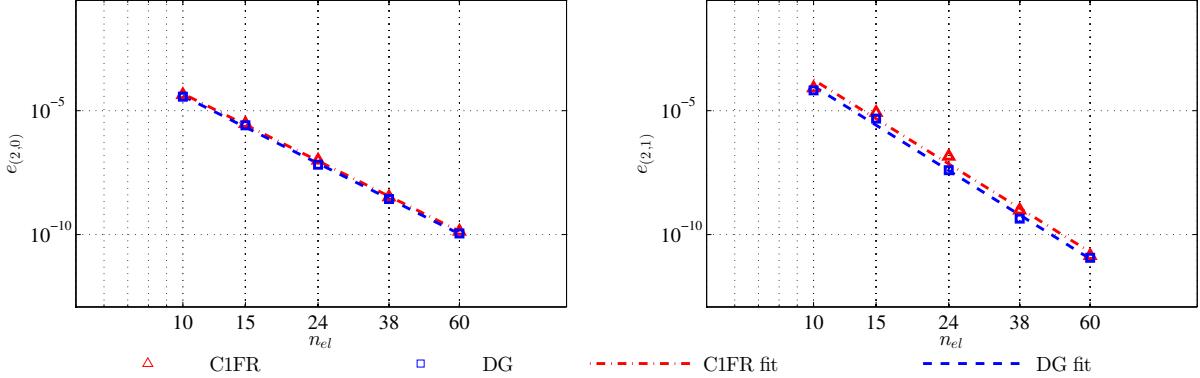


Figure 3.5: L-2 norm of error of advected sine wave and its derivative, $e_{(2,0)}$ and $e_{(2,1)}$ respectively, versus number of elements, for linear advection with polynomial discretization of order $P = 3$. Order of accuracy in solution: DG: 7.119, C1FR: 7.187. Order of accuracy in first derivative: DG: 7.908, C1FR: 7.882.

where ρ is a non-dimensional constant, $P + 1$ is the number of solution points in each element of polynomial degree P , and h is the size of the element. Note that when $\rho = 1$, the Nyquist limit is reached exactly if the solution points are spaced evenly.

In our experiments, for the low wavenumber $\rho = 0.25$; medium wavenumber $\rho = 0.5$; high wavenumber $\rho = 0.75$. The fluxes were all fully upwinded and in the C1FR scheme, $c_1 = -5 \cdot 10^{-3}$. The solution is advanced with a standard RK4 time-stepping scheme. A CFL of 0.3 is used for both schemes. At each timestep, we calculate the square of the L-2 norm of the solution and its derivative, and compare it to the exact corresponding values. $\|u\|_{(2,m)}$ is the L-2 norm of the m^{th} derivative of solution u .

3.6.2.2 Results and discussion

Fig. 3.6 shows that both schemes preserve the exact solution and derivative norms of the low wavenumber. On the other hand, 3.8 shows that both schemes suffer from aliasing and deviate significantly from the exact L-2 norms when the initial solution is a high wavenumber. C1FR is somewhat closer to the exact values than nodal DG both before and after the norms of the numerical solutions intersect the exact solution's L-2 norm.

Fig. 3.7 presents a promising result. C1FR preserves the correct L-2 norms of

the solution while nodal DG's numerical dissipation affects the energy content of the wave. The L-2 norm of C1FR's solution derivative oscillates around the exact value, while nodal DG's oscillates with similar magnitude trending further below the exact values.

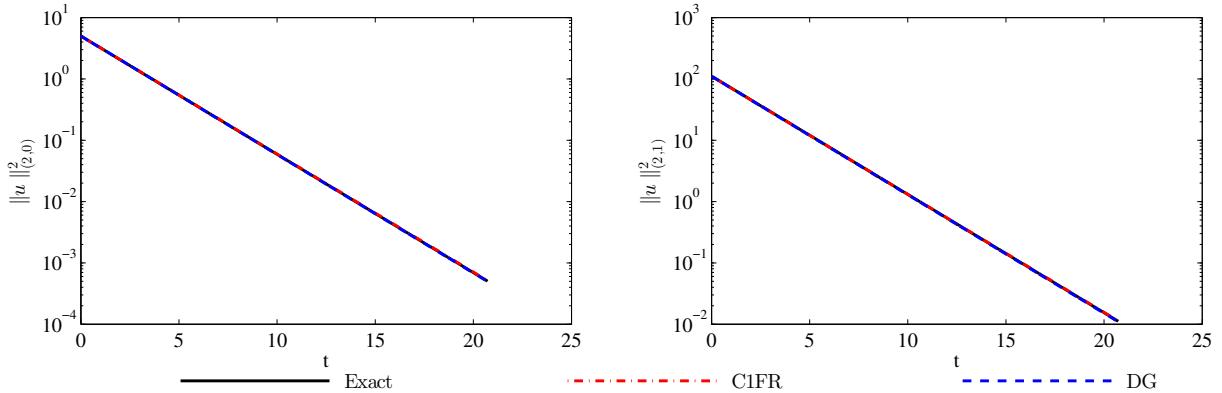


Figure 3.6: Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with low wavenumber: $k = 0.25(P + 1)\pi/h$, $P = 3$, $h = 1$.

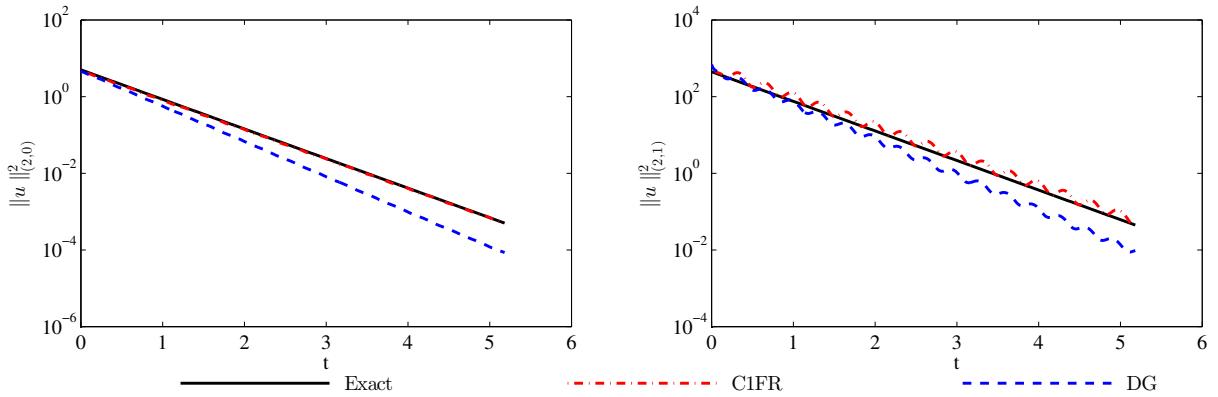


Figure 3.7: Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with medium wavenumber: $k = 0.5(P + 1)\pi/h$, $P = 3$, $h = 1$.

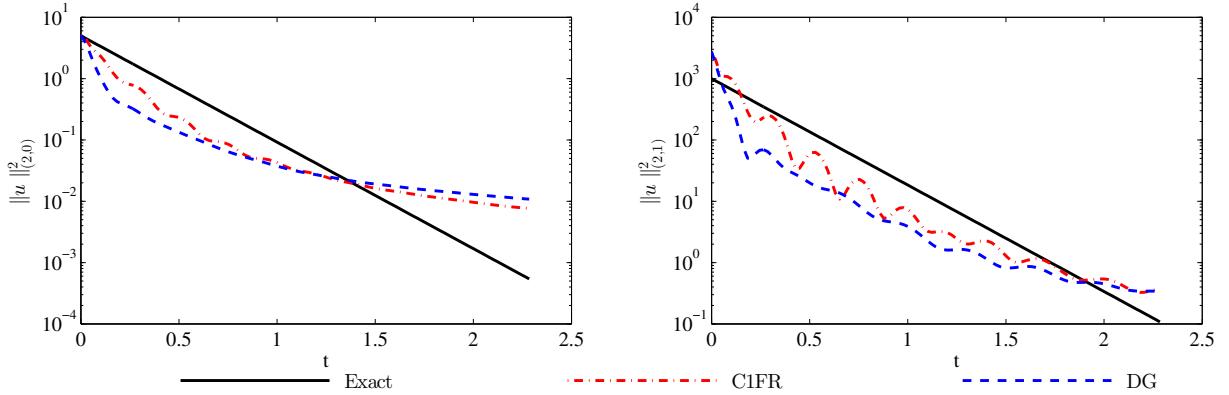


Figure 3.8: Time history of norms of numerical solutions to the advection-diffusion equation and their first derivative. Initial condition is a sine wave with high wavenumber: $k = 0.75(P + 1)\pi/h$, $P = 3$, $h = 1$.

3.6.3 Solutions to the Euler Equations

3.6.3.1 Setup

Solving the 1-D Euler equations is a good test of a scheme's robustness and potential for solving challenging Navier-Stokes cases. In this section we compare the performance of C1FR to that of unmodified DG. The equations in conservative form are

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad (3.75)$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix}, \quad (3.76)$$

$$E = \left(e + \frac{1}{2}u^2 \right) \rho, \quad (3.77)$$

and

$$p = (\gamma - 1) \left(E - \frac{1}{2}\rho u^2 \right). \quad (3.78)$$

γ , ρ , e are the usual symbols of ratio of specific heat capacities of the gas, density, and specific internal energy, respectively.

We can rewrite the system of equations so only three variables appear:

$$U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix}, \quad F = \begin{pmatrix} U_2 \\ \frac{U_2^2}{U_1} + p \\ \frac{U_2}{U_1}(U_3 + p) \end{pmatrix}, \quad p = (\gamma - 1) \left(U_3 - \frac{1}{2} \frac{U_2^2}{U_1} \right) \quad (3.79)$$

It is possible to find exact solutions to problems with initial conditions of the form

$$\rho(x, 0) = \begin{cases} \rho_L & \text{if } x < x_{ref} \\ \rho_R & \text{if } x \geq x_{ref} \end{cases},$$

$$p(x, 0) = \begin{cases} p_L & \text{if } x < x_{ref} \\ p_R & \text{if } x \geq x_{ref} \end{cases}, \text{ and}$$

$$u(x, 0) = \begin{cases} u_L & \text{if } x < x_{ref} \\ u_R & \text{if } x \geq x_{ref} \end{cases}.$$
(3.80)

where subscripts R, L mean the value is constant to the right and left, respectively, of the point x_{ref} . A thorough description on how to find such exact solutions is in Section 4.2 in [77].

High order methods are known to not perform well in the presence of shocks. This is specially true when the initial condition is discontinuous. C1FR schemes are not impervious to this problem and become unstable at any CFL with discontinuous initial conditions such as those in Eqn. (3.80).

In order to produce a solution, the initial discontinuity is “thickened” by using a hyperbolic tangent (\tanh) function, as opposed to a Heaviside step function, to step from the left value to the right value. For the following results,

$$y = \frac{y_R - y_L}{2} \tanh(K(x - x_{ref})) + \frac{y_R + y_L}{2} \quad (3.81)$$

where y is the quantity being initialized (ρ, p, u) and K modifies the sharpness of the step. When $K \rightarrow \infty$, we recover the Heaviside step function. A value of $K = 90$ produced a subjectively appropriate sharpness and allowed the numerical solution

to develop shocks by itself. We are interested in seeing how the numerical scheme handles the latter.

Even though the initial conditions were modified, it is still possible to observe that the C1FR functions have enhanced built-in resilience relative to regular DG.

No modifications to the interface flux definitions were made for the following results. It can be argued that a different selection of fluxes and the use of limiters or filters could improve the results of both C1FR and DG. However, the goal of the following exposition is not to present the “best” or a “better” numerical scheme for the solution of the Euler equations, but rather to assess the behavior of a general scheme like C1FR in a challenging situation that could appear in a flow of engineering interest. The idea behind this goal is that if an untuned scheme performs well in a challenging scenario, it is reasonable to expect the engineer will not need to spend much time and effort tuning the simulation parameters to obtain a useful solution to a problem.

3.6.3.2 Results and discussion

3.6.3.2.1 Sod’s Shock Tube

The initial conditions for this problem are, as shown in [72],

$$\rho(x, 0) = \begin{cases} 1 & \text{if } x < 0.5 \\ 0.123 & \text{if } x \geq 0.5 \end{cases}, \quad p(x, 0) = \begin{cases} 1 & \text{if } x < 0.5 \\ 0.1 & \text{if } x \geq 0.5 \end{cases}, \quad \text{and } u(x, 0) = 0.$$

Figure 3.9 plots the initial conditions with the “thickened” discontinuity.

Figure 3.10 shows the results to this problem at $t = 0.25$ s using C1FR. The flux used in the 0th derivative is central ($\alpha_0 = 1$ in Eqn. (3.38)) and the flux used in the 1st derivative is upwinded ($\alpha_1 = 0$ in Eqn. (3.38)). The correction functions are created with $c_1 = 1e - 2$ in Eqn. (3.68). The time-stepping method was RK4, and the CFL for the C1FR and DG cases was $2.5e - 2$. The timestep was set to

$$\Delta t = \frac{\text{CFL}h}{|a| + |u|} \tag{3.82}$$

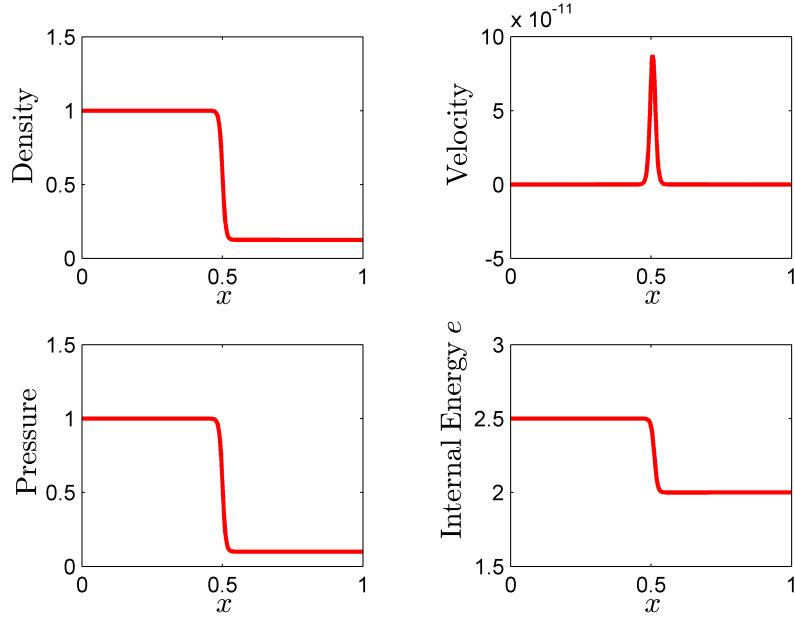


Figure 3.9: Sod’s Shock Tube Problem with “thickened” discontinuity at $t = 0$.

where h is the element size and $|a| = \sqrt{\gamma \frac{p}{\rho}}$ is the speed of sound.

The solution to the Shock Tube problem with C1FR exhibits oscillations at the contact discontinuities, as expected. During the run of this simulation, the magnitude of these oscillation increased and decreased. The maximum value of the internal energy was not achieved. It is worth noting that slight oscillations are also present at the plateaus. This is an unexpected result, as a central flux causes large oscillations beyond the discontinuity points. It could be surmised that the upwinding of the first derivative flux acted as a limiter.

Figure 3.11 shows the solution with regular, unfiltered, non-limited DG with the upwinded Rusanov flux ($\alpha_0 = 0$ in Eqn. (3.38)).

The solution to the Shock Tube problem with DG exhibits larger oscillations at the contact discontinuities than with C1FR. The maximum value of the internal energy was achieved, albeit with very large oscillations at the discontinuity. Results by Lv et al. [54] show that even some bounding strategies produce similar overshoot magnitudes at the discontinuities. Visible oscillations are also present at the plateaus, contrary to the results with C1FR. This behavior is expected; DG has been used

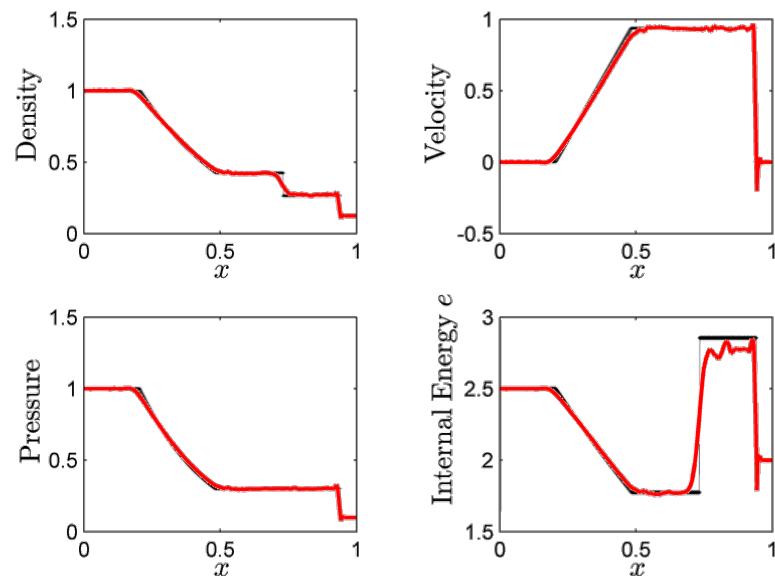


Figure 3.10: Sod’s Shock Tube Problem at $t = 0.25$ solved with C1FR. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the C1FR scheme and “thickened” discontinuity in the initial conditions shown in Figure 3.9. $N = 71, P = 3, c_1 = 1e - 2, \alpha_0 = 1, \alpha_1 = 0, \text{CFL} = 2.5e - 2$

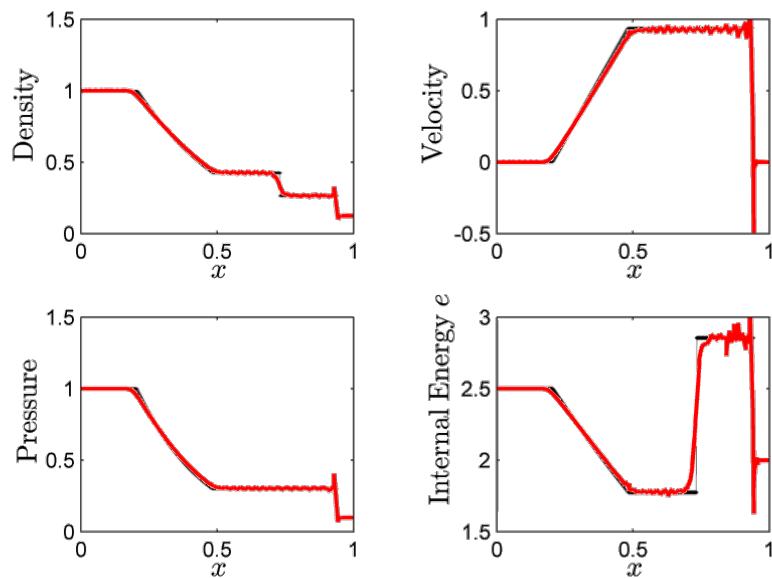


Figure 3.11: Sod’s Shock Tube Problem at $t = 0.25$ solved with regular DG. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the DG scheme and “thickened” discontinuity in the initial conditions shown in Figure 3.9. $N = 70, P = 3, \alpha_0 = 0$

to solve the 1-D Euler equations neatly in the presence of shocks when combined with limiters [89] or filters [3]. It could be surmised that the upwinding of the first derivative flux acted as a limiter.

3.6.3.2.2 123 Problem

The 123 Problem was designed to reach conditions in which the Euler equations cannot be linearized and, hence, the Roe Flux does not provide a physical answer. It is important to note that this case is challenging for even low-order methods created specifically for the Euler equations, as seen in [35], where the Harten-Lax-van Leer and Einfeldt (HLLE) scheme [27] and Advection Upstream Splitting Method Plus (AUSM+) [46] are compared. The initial conditions are, as shown in [28],

$$\rho(x, 0) = 1, \quad p(x, 0) = 0.4, \quad \text{and} \quad u(x, 0) = \begin{cases} -2 & \text{if } x < 0.5 \\ 2 & \text{if } x \geq 0.5 \end{cases}.$$

Figure 3.12 plots the initial conditions with the “thickened” discontinuity.

Figure 3.13 shows the results at $t = 0.1$ s using C1FR. The parameters for C1FR in this case are the same as in Section 3.6.3.2.1, including the CFL.

Stable results with unmodified DG could not be obtained. The reader is directed to Figure 3.4 in [89] and Figure 4.3 in [35] to appreciate that accurate solutions to this problem are particularly difficult. [89] needed to design fluxes for DG to obtain a reasonable result.

The most common challenge for the schemes in the aforementioned references is that the internal energy at $x = 0$ is always over-estimated. For example, as shown in [35], HLLE overestimates $e(x = 0, t = 0.1)$ by 0.1; AUSM+ by 0.8; and [89] by at least 0.2, even though Wang’s simulations used DG with $P = 1$ (second order accurate in space).

Given how challenging this problem is, C1FR performed exceedingly well without the use of limiters or filters. $e(x = 0, t = 0.1)$ is overestimated by 0.1 and shows some oscillations around $x = 0$. The plateau in $u(x = 0, t = 0.1)$ is missed (likely because of the “thickened” discontinuous initial conditions). This type of result suggests that

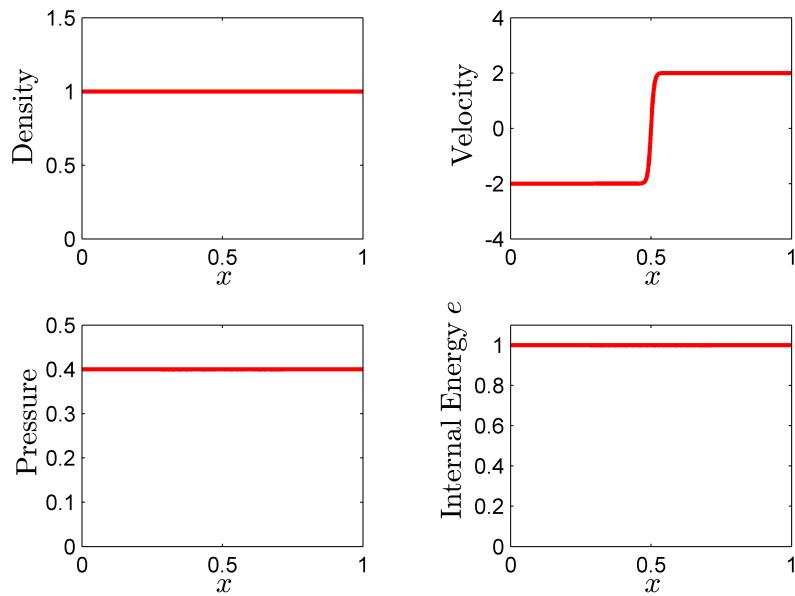


Figure 3.12: 123 Problem with “thickened” discontinuity at $t = 0$ solved with C1FR. Solid black line is the exact solution to the original problem with discontinuous initial conditions. Superimposed solid red line is the solution obtained with the C1FR scheme and “thickened” discontinuity in the initial conditions shown in Figure 3.12. $N = 71, P = 3, c_1 = 1e - 2, \alpha_0 = 1, \alpha_1 = 0, \text{CFL} = 2.5e - 2$

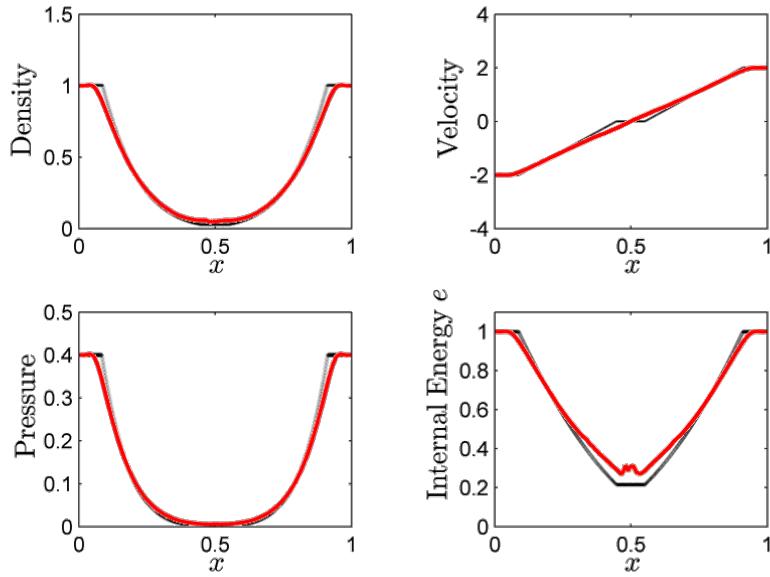


Figure 3.13: 123 problem at $t = 0.1$. Solid black line is the exact solution. Superimposed solid red line is the solution obtained with the C1FR scheme.

C1FR could be robust enough for flows of engineering interest.

3.7 Conclusions

We have presented a natural extension of the FR approach. The CMFR schemes guarantee 1-D linear stability and introduce an arbitrary number of parameters that modify the scheme's dispersive and dissipative properties. The addition of these parameters require the representation of the reconstructed flux to be $p + 1$ or higher, where p is the order of the polynomial used to represent the conservative solution.

We have shown the derivation of the C1FR scheme, which has reconstructed fluxes continuous in the zeroth and first derivatives across elements. This scheme, with a particular selection of its free parameter, was able to preserve the energy of an advected and diffused wave with a medium wavenumber better than the nodal DG scheme. Certainly, this one example cannot be said to be generalizable. Nevertheless, it calls for a Von Neumann analysis to assess the impact of the free parameter on

the scheme's properties. As the general CMFR schemes have arbitrarily many such parameters, this analysis should be generalizable.

A major complication with the general CMFR schemes is that the correction functions do depend on the element's Jacobian, so they are not as general and element-agnostic as those in the original FR schemes. However, as shown by Allaneau [2], it is possible to formulate some FR schemes as filtered DG schemes without the need to find the correction functions explicitly. A similar analysis with the FR schemes could yield element-dependent filtered DG schemes whose properties can be understood or optimized in the CMFR framework.

C1FR solutions to the 1-D Euler equations suggest that the scheme is more robust than regular DG in problems likely to arise in engineering applications. This is not to mean that solutions cannot be obtained with DG, but rather that the built-in robustness in C1FR could allow it to provide good solutions to challenging flow problems without the need of tuning, limiting, or filtering. Certainly, performing any of the latter could enhance the accuracy and robustness of the scheme, as it does in DG.

Future work should include numerical experiments in 2-D and 3-D using tensor product elements to assess the extent to which the stability guarantees in 1-D translate to other dimensions. In addition, it is still unclear if the degree of continuity of the corrected fluxes could be beneficial in the solution of high order partial differential equations like the heat equation.

Chapter 4

Local Fourier Spectral Filters

4.1 Introduction

Low-order methods are ubiquitous in industry and academia. Regardless of the mesh quality and flow conditions, commercial CFD packages output an answer. It is up to the informed user to decide if such answer is believable or accurate to her satisfaction. This is not so true of high-order methods: they are still sensitive to starting conditions, mesh quality, and the non-linearity of the flow, i.e. how high Re is. If any of these parameters is not chosen well, the simulation will halt prematurely and no result, not even a rough estimate, will be provided. Certainly, this is not acceptable in an industrial setting.

The LFS filters being proposed in this chapter are an attempt at tackling the low robustness of high-order methods from the perspective of flow physics, rather than the classical frameworks of polynomial order reduction, artificial viscosity, or limiting. Very promising results have been shown by Asthana and the author [3] for 1D non-linear advection-diffusion problems and 2D inviscid high- Ma flows with quadrilateral, unstructured, coarse grids, and polynomial discretizations of up to order 8. The present paper extends their formulation to arbitrary elements in arbitrary dimensions and shows results in high- Re flows without turbulence modeling.

It is well known that turbulent flows –which tend to be high- Re –, exhibit an energy cascade: the energy from large scales is transferred to smaller scales due to

natural dissipation. In very crude terms, large vortices become ever smaller vortices until they reach dimensions proportional to the Kolmogorov length-scale [42]. This phenomenon is well captured by the NS equations. Hence, a good NS solver would see large scales become ever smaller. In general, low-order methods in grids not created for DNS introduce enough numerical dissipation that the ever shrinking scales are dissipated before they become aliased (or under-sampled).

We postulate that it is precisely this very natural energy cascade which is destabilizing high-order numerical methods. Because high-order methods introduce little numerical dissipation, the ever shrinking scales may not be dissipated before they become aliased: they re-appear as larger scales that, naturally, become smaller later on. This vicious cycle introduces non-physical energy into the flow until the simulation is de-stabilized. Thus, removing the small scales before they become aliased would, in theory, stabilize the solution.

LFS filters target scales relative to the element size, as the filtering operation happens in the reference element. The smaller the element, the smaller the scale being filtered. In addition, the filters help satisfy boundary conditions. The results presented here show that the LFS filters can stabilize not only high-Re flows but also moderate-Ma and low-Ma flows in coarse grids, which opens the door to using the filters as pre-conditioners or in multigrid cycles. All simulations being presented ran from start to finish without intervention.

Many stabilization schemes created for high-order methods have focused on shock-capturing. A concise review of the shock-capturing literature can be seen in Section 3.4 in [83]. An AV-based shock capturing approach that has gained popularity because of its ease of implementation and increase of robustness was suggested by Persson and Peraire [69]. A similar approach for high-Re flows with turbulence modeling has been proposed by Nguyen et al. [60]. Lodato [50] has used filtering in the formulation of SGS models for LES with high-order SD schemes. His work inspired the formulation of the filters presented here. A stabilization strategy based on optimization was suggested by Guba et al. [33] shows great promise. A limiter-based stabilization strategy easily implemented in DG-type methods was proposed by Kuzmin [44] and Lv et al. [54].

The main reason we decided to find a stabilization strategy that could be posed as a matrix multiplication and requires a very local stencil arises from the fact that HiFiLES performs best on GPUs. GPUs require a low-communication, highly-parallel implementation with organized memory accesses and homogeneous computations. Implementation of AV would have required elements adjacent to each other to share information about the AV that each requires, leading to inevitable additional inter-element communication.

Section 2.3.1 presented a general description of the FR method. Section 2.3.1.4 showed how this scheme relies on matrix multiplications and, hence, a filter that costs two small matrix multiplications per element is ideal for FR. Section 4.2 describes the properties being sought in the LFS filters and the mechanics of their implementation. Section 4.3 provides visualizations of the filters and their effects on a polynomial solution. Section 4.4 presents the results of 2D simulations, in unstructured coarse grids, of flows where $\text{Re} = 1e6$. This Re number was selected because of the availability of experimental data for the case of the circular cylinder and HiFiLES had not been run at these Re numbers. In addition, Section 4.4.7 shows results that isolate the effects of filtering from the effects of coarsening a grid or changing the spatial order of accuracy to demonstrate that LFS filters preserve element-wise spectral properties.

4.2 Local Fourier Spectral Filters

4.2.1 Desired Properties

When designing the LFS Filters presented in this paper, we considered the following properties as desirable:

1. The filter should have spectral interpretation so there is control over the physical scales being smoothed
2. The filtering operation must have a local stencil: only interior and boundary solution values can be used to filter the solution in the interior of the element

3. The filter should preserve boundary conditions
4. Filtering the solution at the interior points should be influenced by the solution values at the element's boundary
5. The influence of a boundary point on the internal point being filtered should be inversely proportional to the distance between them
6. To limit computational cost, the filter should be applied in the reference domain and involve matrix-vector multiplications exclusively
7. Filtering strategy should be generalizable to any type of element in unstructured grids

In the early stages of the filter design, we noted that Properties 3 and 4 could be overly stringent, so we re-phrased them as the following, more relaxed condition:

8. If solution values at the boundary lie on a hyperplane in the space of spatial coordinates and solution values, the filtering operation should bring the internal solution values closer to such plane. In other words, if the solution values at the boundaries can be determined from a linear function of their location, the filter should bring the internal solution values closer to satisfying such linear function.

Condition 8 may not allow a filtered solution to satisfy the boundary conditions always. Nevertheless, in the limit of infinite mesh refinement the solutions at the boundary of every element will be coplanar, even in the presence of shocks in the NS equations. Hence, Condition 8 satisfies conditions 3 and 4 asymptotically. This re-formulation was inspired by the Essentially-Local-Extremum-Diminishing (ELED) property of the Jameson-Schmidt-Turkel (JST) scheme[39] for finite volume methods. Whether or not (some) LFS filters satisfy ELED properties shall be left for a future study.

4.2.2 Mechanics of LFS Filters

The core idea of the LFS filtering technique is that the solution is filtered with a matrix whose entries depend on the element interface values, the basis functions of

the solution, and a selected Fourier filtering kernel.

Suppose we wish to filter the scalar field v inside an arbitrary element. This field could be density, momentum in a specific direction, or energy. We can represent v , as usual, as a weighted sum of basis functions.

$$v(\boldsymbol{\xi}) = \sum_{i=1}^{N_s} v_i \phi_i(\boldsymbol{\xi}), \quad (4.1)$$

where v_i is the i^{th} weight, $\boldsymbol{\xi}$ is a vector of coordinates in a reference domain, and $\phi_i(\boldsymbol{\xi})$ is the i^{th} basis function.

The key component of the LFS technique is that the Fourier filtering operation over the element is approximated at each internal point i with coordinates $\boldsymbol{\xi}_i$. Suppose we wish to use filtering kernel $G(\boldsymbol{\xi})$ to filter $v(\boldsymbol{\xi})$, then the filtered value at internal point i becomes

$$\begin{aligned} \bar{v}_j &= (G * v)(\boldsymbol{\xi}_j) \\ &= \int_{\mathbb{R}^d} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) v(\boldsymbol{\xi}) d\boldsymbol{\xi} \\ &= \int_{\mathbb{R}^d} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) \left(\sum_{i=1}^{N_s} v_i \phi_i(\boldsymbol{\xi}) \right) d\boldsymbol{\xi} \\ &= \sum_{i=1}^{N_s} v_i \int_{\mathbb{R}^d} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) \phi_i(\boldsymbol{\xi}) d\boldsymbol{\xi}, \end{aligned} \quad (4.2)$$

where Ω is the entire problem domain, not just the element domain, N_s is the number of solution points, and the overbar means the quantity is filtered. Filtering over the entire domain would be computationally expensive, but would certainly smoothen the solution field and can be done while maintaining the order of accuracy of the underlying numerical scheme[57, 79, 88]. In order to use only a local (element-wise) stencil, it is possible to break the integration as follows

$$\int_{\mathbb{R}^d} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) \phi_i(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int_{\Omega_n} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) \phi_i(\boldsymbol{\xi}) d\boldsymbol{\xi} + \int_{\mathbb{R}^d \setminus \Omega_n} G(\boldsymbol{\xi}_j - \boldsymbol{\xi}) \phi_i(\boldsymbol{\xi}) d\boldsymbol{\xi}, \quad (4.3)$$

where Ω_n is the domain of element n (where the ϕ_i values are known) and $\Omega \setminus \Omega_n$ is the complement of Ω_n in Ω . As G and ϕ_i are defined in Ω_n , it is possible to calculate the first integral in Equation (4.3) analytically or via an adaptive quadrature algorithm like `quanc8` [29].

A design choice arises in defining ϕ_i in the domain $\Omega \setminus \Omega_n$. Asthana et al. [3] initially decided to let ϕ_i equal a constant in 1D elements, and a combination of constant and polynomial in quadrilaterals. Non-tensor-product elements became problematic for this kind of formulation. This paper presents a more generic filter design method based on the desired properties in Section 4.2.1.

Let us describe the mechanics of the two components of the integral in Equation (4.3) in a subsection each. We can call the filter associated with the term $\int_{\Omega_n} G(\boldsymbol{\xi}_j - \boldsymbol{\xi})\phi_i(\boldsymbol{\xi})d\boldsymbol{\xi}$ “Internal Filtering Component” and the filter associated with the term $\int_{\mathbb{R}^d \setminus \Omega_n} G(\boldsymbol{\xi}_j - \boldsymbol{\xi})\phi_i(\boldsymbol{\xi})d\boldsymbol{\xi}$ “Boundary Filtering Component”.

The filtering operation being sought follows this format

$$\vec{v} = \alpha \mathcal{T} \vec{v} + (1 - \alpha) \mathcal{B} \vec{v}^*, \quad (4.4)$$

where \mathcal{T} is the internal filtering component, as it acts on the solution at internal points \vec{v} , and \mathcal{B} is the boundary filtering component, as it acts on the solution at boundary points \vec{v}^* , and α is a scalar between 0 and 1 that determines the relative influence the internal and boundary components have on the internal solution values.

4.2.3 Internal Filtering Component

There are two steps to the creation of the internal filtering component \mathcal{T} . First we create the filtering matrix with the spectral interpretation, and then we normalize it so if the quantity being filtered is a constant, the filter preserves the constant.

4.2.3.1 Matrix formation

It can be seen from the last line in Equation (4.2) that finding the vector of solution values filtered with the internal component can be posed as a matrix-vector

multiplication:

$$\begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \vdots \\ \bar{v}_{N_s} \end{bmatrix} = \begin{bmatrix} \int_{\Omega_n} G(\xi_1 - \xi) \phi_1(\xi) d\xi & \int_{\Omega_n} G(\xi_1 - \xi) \phi_2(\xi) d\xi & \cdots & \int_{\Omega_n} G(\xi_1 - \xi) \phi_{N_s}(\xi) d\xi \\ \int_{\Omega_n} G(\xi_2 - \xi) \phi_1(\xi) d\xi & \int_{\Omega_n} G(\xi_2 - \xi) \phi_2(\xi) d\xi & \cdots & \int_{\Omega_n} G(\xi_2 - \xi) \phi_{N_s}(\xi) d\xi \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\Omega_n} G(\xi_{N_s} - \xi) \phi_1(\xi) d\xi & \int_{\Omega_n} G(\xi_{N_s} - \xi) \phi_2(\xi) d\xi & \cdots & \int_{\Omega_n} G(\xi_{N_s} - \xi) \phi_{N_s}(\xi) d\xi \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_s} \end{bmatrix}.$$

More compactly,

$$\vec{\bar{v}} = \mathcal{F}\vec{v} \quad (4.5)$$

we use over-arrow to denote column vectors. Interesting choices for $G(\xi)$ depend on a desired wavenumber threshold h :

I. Gaussian function:

$$G(\xi) = h \exp(h||\xi||), \quad (4.6)$$

where $||\xi||$ is a measure of the length of ξ .

II. Multidimensional indicator function:

$$G(\xi) = h I_{[0,1]}(h||\xi||), \quad (4.7)$$

where $I_{[0,1]}(x) = 1$ if $0 \leq x \leq 1$, and 0 otherwise.

III. Multidimensional sharp-spectral function:

$$G(\xi) = h||\xi||^{-n/2} J_{n/2}(h||\xi||), \quad (4.8)$$

where J_α is a Bessel function of the first kind and n is the number of dimensions of ξ .

In these functions, the larger the value of h , the thinner the filter is in physical space, so the fewer wavenumbers it dampens. Note that as $h \rightarrow \infty$, $\mathcal{F}_{ij} \rightarrow \phi_j(\xi_i)$, so $\vec{\bar{v}} \rightarrow \vec{v}$. The integrals in Equation (4.2.3.1) can be evaluated to arbitrary accuracy with `quanc8` [29] in a pre-processing stage.

4.2.3.2 Enforcing Conservation of a Constant Quantity

The requirement to preserve a constant can be posed in an equation as

$$\bar{v}_i = \sum_{j=1}^{N_s} \mathcal{F}_{ij} v_{\text{const}} = v_{\text{const}}, \quad (4.9)$$

where v_{const} is a constant scalar. As a result, each row of \mathcal{F} must satisfy

$$\sum_{j=1}^{N_s} \mathcal{F}_{ij} = 1, \quad (4.10)$$

where $i = 1, \dots, N_s$. This constraint can be enforced by dividing each row of \mathcal{F} with the total sum of the row. More explicitly,

$$\mathcal{T}_{ij} = \frac{\mathcal{F}_{ij}}{\sum_{k=1}^{N_s} \mathcal{F}_{ik}}, \quad (4.11)$$

where the normalized matrix \mathcal{T} is the internal filtering component in Equation (4.4).

4.2.3.3 Integration Domain

The normalization of the matrix introduces a bias. As the integration is being performed in some reference domain Ω_n , the integral $\int_{\Omega_n} G(\boldsymbol{\xi}_i - \boldsymbol{\xi}) \phi_j(\boldsymbol{\xi}) d\boldsymbol{\xi}$ will tend to have a greater value when $\boldsymbol{\xi}_i$ is well inside the reference element. This is more evident when using the multidimensional indicator function (shown in II) as the filtering kernel.

To ameliorate this bias, we have selected to perform the integration in a reference domain that is symmetric. With this precaution all integrals related to points close to a domain edge have a similar magnitude, and thus the filtering operation does not favor points close to some arbitrary edge over the others.

Figure 4.1 illustrates the potential introduction of bias in triangular elements that use a right triangle as a reference element.

In practice, the basis functions $\phi_j(\boldsymbol{\xi})$ may be defined in non-symmetrical elements. To simplify the evaluation of the integral, the definition of the norm in the filtering

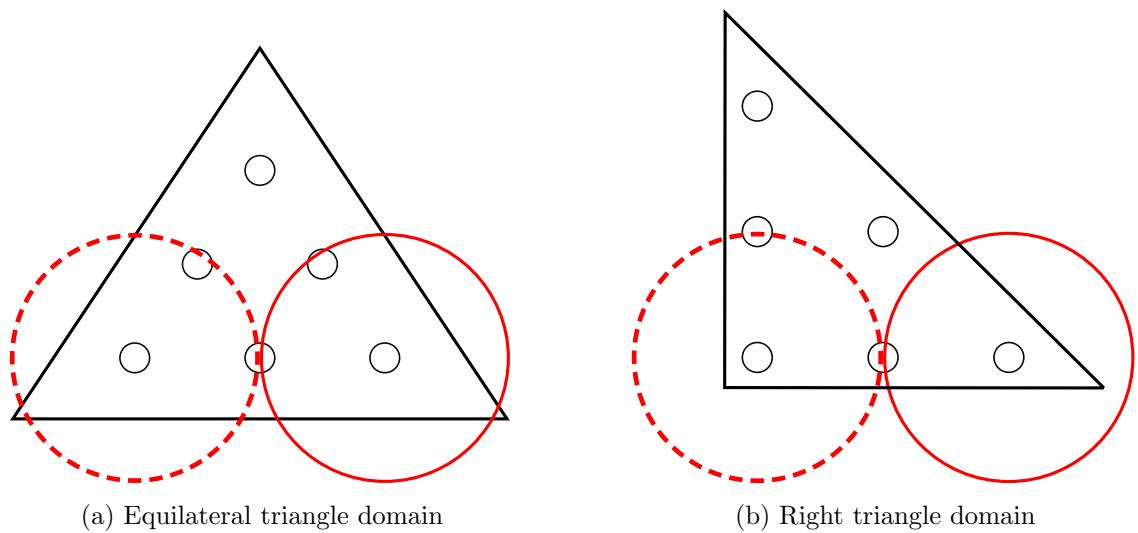


Figure 4.1: Using the same filter width in two different domains introduces different bias. The small, solid circles represent the location of solution points. The large circles represent the filter width acting on two specific internal points in two different domains. Integration is being performed in the area encompassed by the solid, straight lines.

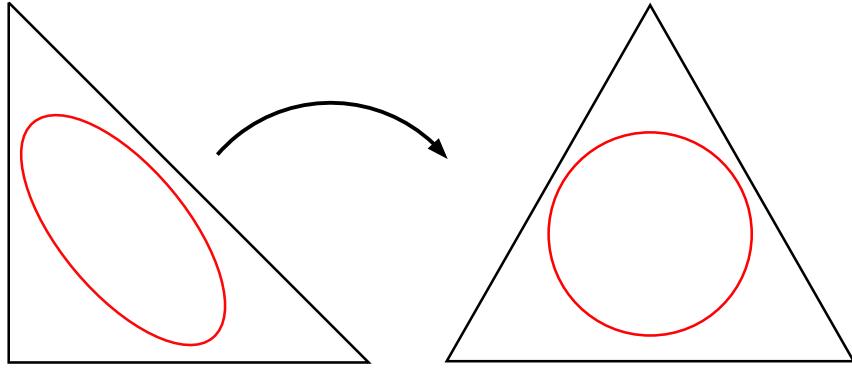


Figure 4.2: Sketch of ellipsis $\|\boldsymbol{\xi} - \boldsymbol{\xi}_0\|_{rt} = 1$ defined in the right triangle domain maps to a circle in the equilateral triangle domain

kernel $G(\boldsymbol{\xi})$ can be modified so the curve drawn by $\|\boldsymbol{\xi} - \boldsymbol{\xi}_0\| = 1$ in the non-symmetric reference element would map to a circle in a symmetric reference element.

In the case of a triangle, the basis functions are defined in a right triangle with vertices $[-1, -1], [1, -1], [-1, 1]$.

By defining the norm in this right triangle as follows

$$\|\boldsymbol{\xi}\|_{rt}^2 = \xi_1^2 + \xi_2^2 + \xi_1 \xi_2, \quad (4.12)$$

$\|\boldsymbol{\xi} - \boldsymbol{\xi}_0\|_{rt} = 1$ maps to a circle in the equilateral triangle with vertices $[-1, -1], [1, -1], [0, \sqrt{3}-1]$. Figure 4.2 sketches this mapping. $\|\cdot\|_{rt}$ is the norm defined in the right triangle.

4.2.4 Boundary Filtering Component

Formulating the influence of boundary solution values on internal solution values as a matrix is not so straightforward. The approach being suggested here is based on some reasonable desired properties, but some choices are arbitrary.

The creation of this matrix proceeds as follows: place the boundary points and the internal points in a symmetric domain, select an “influence” function with as many unknowns as dimensions, find values for these unknowns such that Condition 8 is satisfied, and create the boundary filtering component.

4.2.4.1 Ensuring Boundary Filtering Component Preserves Linearity of the Solution at the Boundary (Condition 8)

In this sub-section, we seek to form matrix \mathcal{B} in Equation 4.4. In this analysis, we assume that $\alpha = 0$ in Equation 4.4 so we can isolate \mathcal{B} 's properties from those of \mathcal{T} .

The preservation of linearity can be posed in the following way. Assume that values \vec{v}^* at the boundaries are linear in space. More explicitly, for each boundary point $i = 1, \dots, N_f$

$$v_i^* = \underline{\xi}_i^{*\top} \vec{a} + b, \quad (4.13)$$

where v_i^* is the solution at the i^{th} boundary point, $\underline{\xi}_i^*$ are the coordinates of such boundary point, \vec{a} is a vector of constant coefficients, and b is a constant scalar. In vector notation, we are assuming

$$\vec{v}^* = \underline{\xi}^{*\top} \vec{a} + \vec{b}^*, \quad (4.14)$$

where $\vec{b}^* = b \mathbb{1}_{[N_f \times 1]}$ and $\underline{\xi}^{*\top}$ is a matrix of dimensions N_f by N_d –number of dimensions.

To satisfy Condition 8, we seek a matrix \mathcal{B} such that, when \vec{v}^* satisfies Equation 4.14,

$$\vec{v} = \mathcal{B} \vec{v}^* = \underline{\xi}^{\top} \vec{a} + \vec{b}, \quad (4.15)$$

where $\vec{b} = b \mathbb{1}_{[N_s \times 1]}$ and $\underline{\xi}^{\top}$ is a matrix of dimensions N_s by N_d . The i^{th} row of $\underline{\xi}^{\top}$ is $\underline{\xi}_i^{\top}$, the coordinates of the i^{th} internal point.

Therefore, we seek a matrix \mathcal{B} such that

$$\mathcal{B} \left(\underline{\xi}^{*\top} \vec{a} + \vec{b}^* \right) = \underline{\xi}^{\top} \vec{a} + \vec{b}. \quad (4.16)$$

As a result, for general $\underline{\xi}^{*\top}$, $\underline{\xi}^{\top}$, and \vec{a} ,

$$\mathcal{B} \underline{\xi}^{*\top} = \underline{\xi}^{\top} \quad \text{and} \quad \mathcal{B} \vec{b}^* = \vec{b} \quad (4.17)$$

$$\implies \sum_{j=1}^{N_f} \mathcal{B}_{ij} \underline{\xi}_{jd}^* = \underline{\xi}_{id} \quad \text{and} \quad \sum_{j=1}^{N_f} \mathcal{B}_{ij} = 1 \quad (4.18)$$

Element type	N_s	N_d	N_f	# of Equations:	# of Unknowns:
				$N_s(N_d + 1)$	$N_s N_f$
Line	$p + 1$	1	2	$2(p + 1)$	$2(p + 1)$
Triangle	$\frac{1}{2}(p + 1) \cdot (p + 2)$	2	$3(p + 1)$	$\frac{3}{2}(p + 1) \cdot (p + 2)$	$\frac{3}{2}(p + 1)^2 \cdot (p + 2)$
Quadrilateral	$(p + 1)^2$	2	$4(p + 1)$	$3(p + 1)^2$	$4(p + 1)^3$
Tetrahedron	$\frac{1}{6}(p + 1) \cdot (p + 2)$	3	$4\frac{1}{2}(p + 1) \cdot (p + 2)$ $\cdot (p + 3)$	$\frac{2}{3}(p + 1) \cdot (p + 2)$ $\cdot (p + 3)$	$\frac{1}{3}(p + 1)^2 \cdot (p + 2)^2$ $\cdot (p + 3)$
Pyramid	$\frac{1}{6}(p + 1) \cdot (p + 2)$	3	$4\frac{1}{2}(p + 1) \cdot (p + 2)$ $\cdot (2p + 3)$	$\frac{2}{3}(p + 1) \cdot (p + 2)$ $\cdot (p + 1)^2$ $\cdot (2p + 3)$	$\frac{1}{6}(p + 1)^2 \cdot (p + 2)$ $\cdot (2p + 3) \cdot (3p + 5)$
Prism	$\frac{1}{2}(p + 1)^2 \cdot (p + 2)$	3	$2\frac{1}{2}(p + 1) \cdot (p + 2)$ $+ 3(p + 1)^2$	$2(p + 1)^2 \cdot (p + 2)$ $+ 3(p + 1)^2$	$\frac{1}{2}(p + 1)^3 \cdot (p + 2)$ $\cdot (4p + 5)$
Hexahedron	$(p + 1)^3$	3	$6(p + 1)^2$	$4(p + 1)^3$	$6(p + 1)^5$

Table 4.1: Number of internal points N_s , number of boundary points N_f , and number of equations and unknowns in matrix \mathcal{B} for each type of element, assuming the solution is being discretized with a polynomial of degree p

for all $i = 1, \dots, N_s$ and $d = 1, \dots, N_d$. Where ξ_{jd}^* is the d^{th} coordinate of the j^{th} boundary point, and ξ_{id} is the d^{th} coordinate of the i^{th} internal point.

Equation 4.18 introduces $N_s(N_d + 1)$ constraints for $N_s N_f$ unknowns in \mathcal{B} . Each row in \mathcal{B} needs to satisfy $N_d + 1$ equations. Table 4.1 shows the number of equations and unknowns for specific elements assuming that the solution within each is represented by a polynomial of degree $p \geq 0$.

All elements, except for the line element, have matrices \mathcal{B} with more unknowns than equations. This calls for a reduction of the number of unknowns in a strategic way. This is achieved by invoking Condition 5: the farther away a boundary point is from an internal point, the less influential it should be.

4.2.4.2 Ensuring Influence of Boundary Points on Internal Points is Inversely Proportional to their Distance (Condition 5)

Given this requirement and Equation (4.18), a reasonable design choice for each entry in \mathcal{B} is:

$$\mathcal{B}_{ij} = \frac{g_i (||\xi_i - \xi_j^*||)^{-1}}{\sum_{k=1}^{N_f} g_i (||\xi_i - \xi_k^*||)^{-1}}, \quad (4.19)$$

where ξ_i is the location of the i^{th} internal point, ξ_j^* is the location of the j^{th} boundary point, and $g_i(\cdot)$ is a monotonically increasing function and $g_i(0) = 0$. The norm $|| \cdot ||$ is ideally defined in a symmetric element, or a reformulation of a norm in an non-symmetric element as in Equation (4.12). We note that the requirement that $\sum_{j=1}^{N_f} \mathcal{B}_{ij} = 1$ is immediately satisfied regardless of the exact form of $g_i(\cdot)$.

To avoid dividing by zero when $||\xi_i - \xi_j^*|| = 0$, we can recast Equation (4.19) as

$$\mathcal{B}_{ij} = \frac{1}{1 + g_i (||\xi_i - \xi_j^*||) \sum_{\substack{k=1 \\ k \neq j}}^{N_f} g_i (||\xi_i - \xi_k^*||)^{-1}}. \quad (4.20)$$

This definition of \mathcal{B} is now too stringent, so it is not clear if the conditions in Equation (4.18) are satisfied for any selection of $g_i(\cdot)$. Each row in \mathcal{B} has N_d equations left to satisfy.

By introducing N_d unknowns in the definition of $g_i(\cdot)$ we can expect to satisfy the remaining N_d equations per row. A choice made here is to let

$$g_i(x) = \sum_{d=1}^{N_d} a_{id} x^d, \quad (4.21)$$

where x is a scalar, and a_{id} are the unknowns to be found in each row i .

To find the values of a_{id} , a regular non-linear minimization function can be invoked. In this paper, we used the downhill simplex method by Nelder and Mead [59]. The implementation in C++ was adapted from Jia [41]. The function being minimized

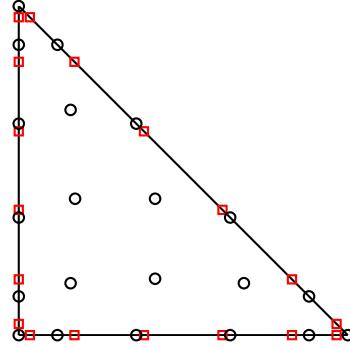


Figure 4.3: Internal and boundary points in the reference triangular domain when solution is represented with polynomial of degree 5 ($p = 5$). Black circles represent the internal points. Red squares represent the boundary points.

for each row i of \mathcal{B} is

$$J_i(a_{i1}, \dots, a_{1N_d}) = \sum_{d=1}^{N_d} \left(\sum_{j=1}^{N_f} \mathcal{B}_{ij} \xi_{jd}^* - \xi_{id} \right)^2, \quad (4.22)$$

where J_i is the cost function to be minimized related to row i . In occasions, the minimum of J does not reach machine zero. However, as it will be shown in the case of triangles, \mathcal{B} still behaves as desired.

4.3 Visualization of the LFS Filters in Triangular Elements

In this section we present visualizations of the effect of the LFS filters on a polynomial solution of degree 5 in a reference triangle.

In the plots that follow, the internal and boundary points are located as shown in Figure 4.3.

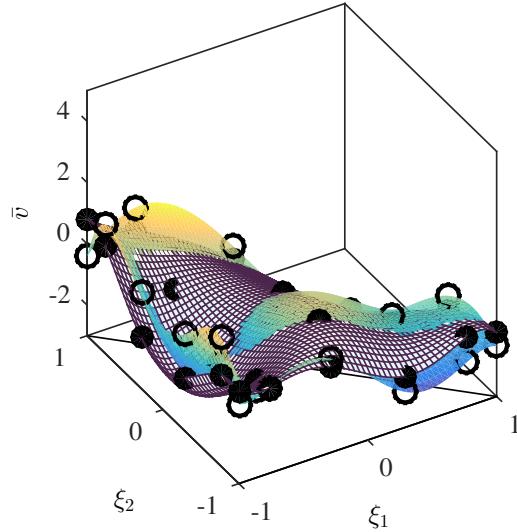


Figure 4.4: Solution values after filtering with internal component exclusively, $\vec{v} = \mathcal{T}\vec{v}$, where $v_i = \sin(k\xi_{i1}) + \cos(k\xi_{i2})$, where $k = 500$. Hollow black circles show the unfiltered solution at the interior points, transparent colored surface is the polynomial representation of the unfiltered solution, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the filtered solution.

4.3.1 Internal Filtering Components

To illustrate the effect of the internal filtering component, let us filter a solution using matrix \mathcal{T} exclusively. Figure 4.4 shows the result of filtering using a width of $h = 10$ in the 2-D sharp-spectral filtering kernel (III) with the modified norm (4.12).

It can be seen that the polynomial representation of the filtered values is smoother than the polynomial representation of the unfiltered values while maintaining the general shape and curvature.

4.3.2 Boundary Filtering Components

To illustrate the effect of the boundary filtering component, let us filter a solution using matrix \mathcal{B} exclusively. Figure 4.5 shows the result of filtering. Because only \mathcal{B} is acting on the solution, the unfiltered values of the solution at the internal points

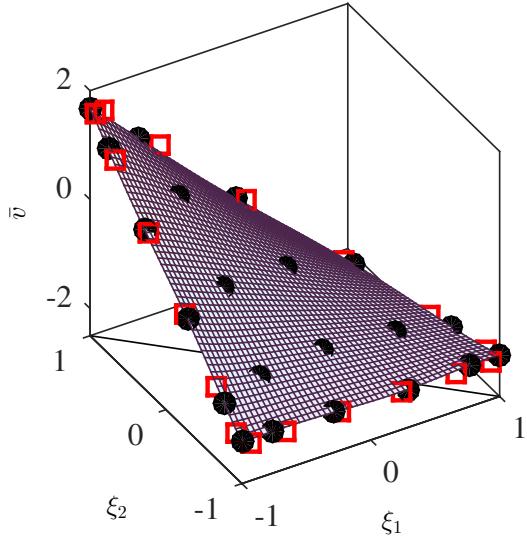


Figure 4.5: Solution values after filtering with boundary component exclusively, $\vec{v} = \mathcal{B}\vec{v}^*$, where $\vec{v}^* = a\vec{\xi}_1^* + b\vec{\xi}_2^* + c$ for some constants a, b, c . Red squares show the values of the solution at the boundaries, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the solution.

are not plotted.

We have made all the values of the solution at the boundaries co-planar to illustrate how effectively Condition 8 is satisfied. The operation of filtering with \mathcal{B} does bring the filtered solution closer to a plane if the boundary values are co-planar.

Figure 4.6 illustrates how the boundary filtering component would behave in the case the boundary values are not coplanar. It is interesting to note that close to the corner $(\xi_1, \xi_2) = (-1, 1)$, two boundary points with different values are close to an internal point. The value of the filtered solution close to those boundary points assumes a value that is close to the average of the two.

The polynomial interpolation of the filtered interior values shows that the closer an interior point is to a boundary point, the more it will be influenced by such boundary point. This causes the filtered values close to the boundaries to get closer to the boundary values. This suggests that the boundary filtering component does in fact help in bringing the solution closer to satisfying the boundary conditions while

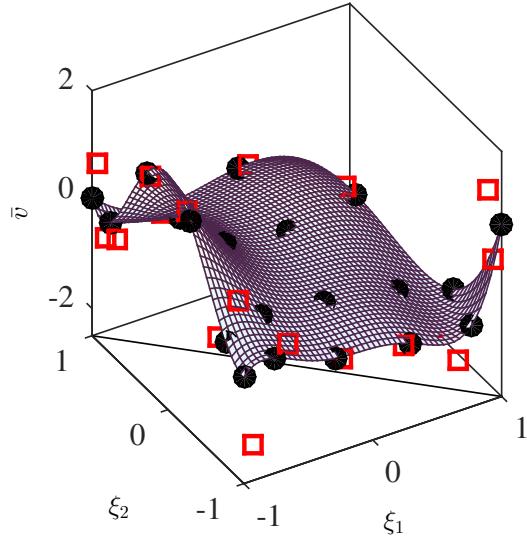


Figure 4.6: Solution values after filtering with boundary component exclusively, $\vec{v} = \mathcal{B}\vec{v}^*$, where $v_i^* = \sin(k\xi_{i1}^*) + \cos(k\xi_{i2}^*)$, $k = 500$. Red squares show the values of the solution at the boundaries, filled black circles show the filtered values of the solution at the interior points, and the meshed surface shows the polynomial representation of the solution.

diminishing oscillations within the element.

4.3.3 Filtered Solutions

No filtering component is used solely by itself. The factor α in Equation 4.4 determines how much weight to give to each component. Figure 4.7 illustrates the effect of the boundary values on a fully filtered solution, when $\alpha = 0.8$.

The internal filtering component reduces oscillations, while the boundary filtering component brings the interior values closer to the boundary values. By placing the boundary values on different planes, this effect becomes more evident.

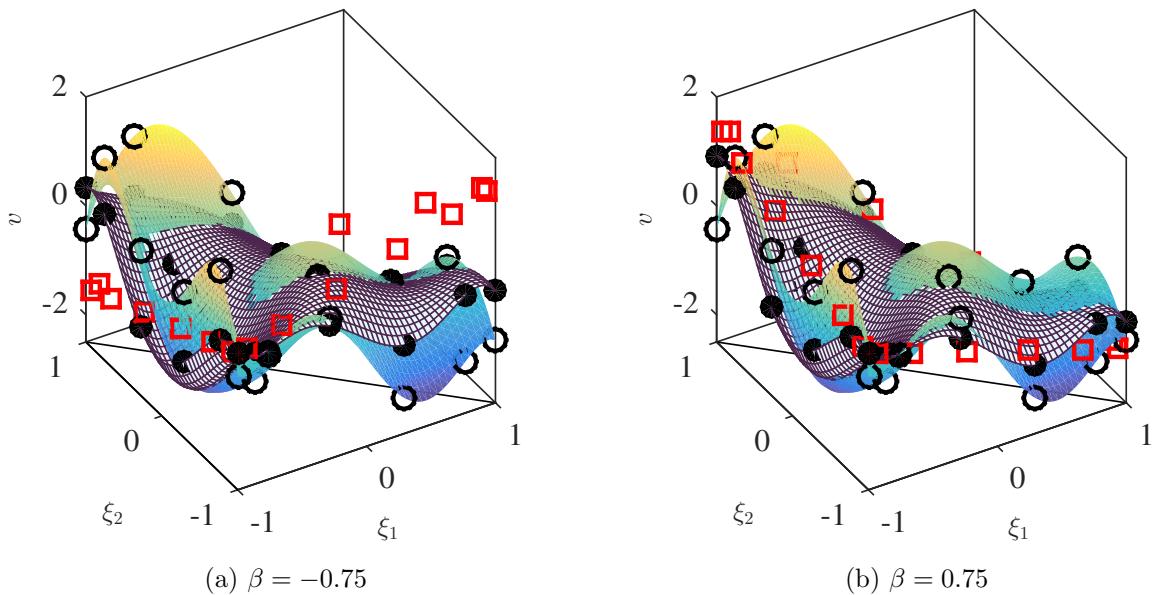


Figure 4.7: Solution values after filtering with both internal and boundary components, $\vec{v} = \alpha \mathcal{T}\vec{v} + (1 - \alpha)\mathcal{B}\vec{v}^*$, where $\alpha = 0.8$, $v_i = \sin(k\xi_{i1}) + \cos(k\xi_{i2})$, $k = 500$, $\vec{v}^* = \beta(-\vec{\xi}_1^* + \vec{\xi}_2^*)$. Hollow black circles show the unfiltered solution at the interior points, transparent colored surface is the polynomial representation of the unfiltered solution, filled black circles show the filtered values of the solution at the interior points, hollow red squares show the value of the solution at the boundary points, and the meshed surface shows the polynomial representation of the filtered solution.

4.4 Results

To test the filters' ability to increase general robustness of a high-order solver, we have implemented their formulation in HiFiLES for triangular elements and performed simulations where the solver would become unstable otherwise. In addition, we analyze the impact the filters have on a well-resolved 2-D simulation.

We wanted to test the increase of robustness in extreme cases of grid coarseness, high Reynolds number, very low Ma, and moderate Ma. It is important to keep in mind that in these cases we are not seeking very accurate results, but rather robustness under all conditions. In order to popularize high-order methods, we need to make them as robust as their low-order counterparts while retaining their benefit of higher accuracy with less computational and setup effort.

The goal is to have a cheap stabilization strategy that preserves boundary conditions for cases in which the mesh is not necessarily perfectly appropriate for resolving the flow physics over the entire domain. This scenario arises frequently in industrial applications, where the mesh would be properly refined at regions of interest and coarse in regions that the engineer/scientist has decided are not as important for the problem at hand.

All the simulations that follow were performed using HiFiLES[51]. 2-D NS equations are being solved, with varying values of Ma, Re, time-step (Δt), and filtering frequency. The common parameters are:

1. Four-stage, five-step, low-storage Runge-Kutta time-stepping method (RK45) [14] was used in the GPUs, and forward Euler was used when running a simulation in CPUs
2. Polynomial solution representation (p) of order 4. Rusanov Flux as a Riemann solver, and a Local Discontinuous Galerkin (LDG) [23] viscous flux.
3. Filters with width $h = 10$ and weighting parameter $\alpha = 0.8$.
4. Starting from uniform flow

Case	Ma	Δt	n_F	\bar{C}_D	St	Flow time (s)	Time steps	Wall time (hours)	Computing Resources
1	0.2	$5e - 5$	1000	0.9256	0.1600	1.2069	1,675,700	12.65	1 4-core i7 CPU
2	0.077	$5e - 5$	1000	0.9314	0.1627	15.44	8,252,500	11.78	1 GPU
3	0.87	$5e - 5$	100	1.8383	0*	1.3833	8,355,256	12.63	1 GPU
4	0.0077	$1.25e - 5$	1000	1.18	0.20	53.44	11,428,000	59.51	2 GPUs

Table 4.2: Summary of simulation results. All cases were run at $Re = 1e6$ with polynomial representation of order 4. Cases 1-3 were run using the mesh shown in Figure 4.8. Case 4 was run using the mesh shown in Figure 4.13. Cases with 0* Strouhal number reached an artificial steady-state.

- 5. Characteristic boundary conditions at the inflow and outflow. No-slip, isothermal wall boundary conditions at the cylinder's surface.
- 6. All quantities non-dimensionalized with free-stream temperature and cylinder wall temperature of 300, reference length of 1.
- 7. Flow properties: $\gamma = 1.4$, Prandtl number $Pr = 0.72$, gas constant $R = 286.9 \frac{J}{KgK}$, viscosity determined by Sutherland's law with reference temperature of $291.15K$ and reference viscosity of $\mu = 1.827e - 5$

Results of interest are shown in Table 4.2. Accuracy of the results is not expected. Nevertheless, as a reference, for the flow around a cylinder at $Re = 1e6$, $\bar{C}_D \approx 0.6$ in [1], $\bar{C}_D \approx 0.4$ in [73] and $St \approx 0.4$ in [73]. It is important to note that at a high Re , flow over a cylinder can result in a range of \bar{C}_D and St values, as the results become highly sensitive to surface roughness and the level of free-stream turbulence [98]. The experimental values of \bar{C}_D vary from 0.17 to 0.40, and those of St from 0.18 to 0.50.

Because of the results obtained in Case 4, it is good to keep in mind that for flow around a cylinder at $Re \approx 2e2$, $\bar{C}_D \approx 1.18$ in [73] and $St \approx 0.2$ in [73].

4.4.1 Stabilization Strategy

In the simulations presented here, the solution inside every element in the entire domain is being filtered using Equation 4.4 every n_F time-steps, where n_F is an integer to be determined. No sensor is being used to detect problems in the flow.

The frequency of filter application is being chosen in the following heuristic way:

1. Start the simulation with a specific time step and no filtering. Record at what time-step the simulation ends prematurely (produces Nan values) and note the value of the residuals at the last valid time-step. This step usually takes no more than 1 minute.
2. To ensure the simulation is ending prematurely because of grid resolution problems or presence of sharp gradients, and not because of an unstable time step, halve the time step and run the simulation again.
3. Wait for the simulation to exit prematurely. If the residual at this last exit is close in value to the previous exiting residual, the time step in Step 2 was stable. Set the new time step to the time step in 2. Otherwise, record the exiting residual and go back to Step 2.
4. Now that a stable time step has been found, apply the filter to the simulation every n_F time steps, where n_F is about 90% of the number of imte-steps it took the simulation to become unstable when unfiltered.

It would certainly be desirable to filter the solution at elements where a problem is detected. Nevertheless, this heuristic approach has so far enabled the stabilization of every case tried and de-couples the effectiveness of the filters from possible shortcomings of aliasing/shock sensors.

4.4.2 Coarse mesh used in simulations

In these tests, we have used the very coarse triangular mesh with 714 elements shown in Figure 4.8. The boundary layer is purposefully not resolved properly, as we would like to induce aliasing errors in the unfiltered calculation.

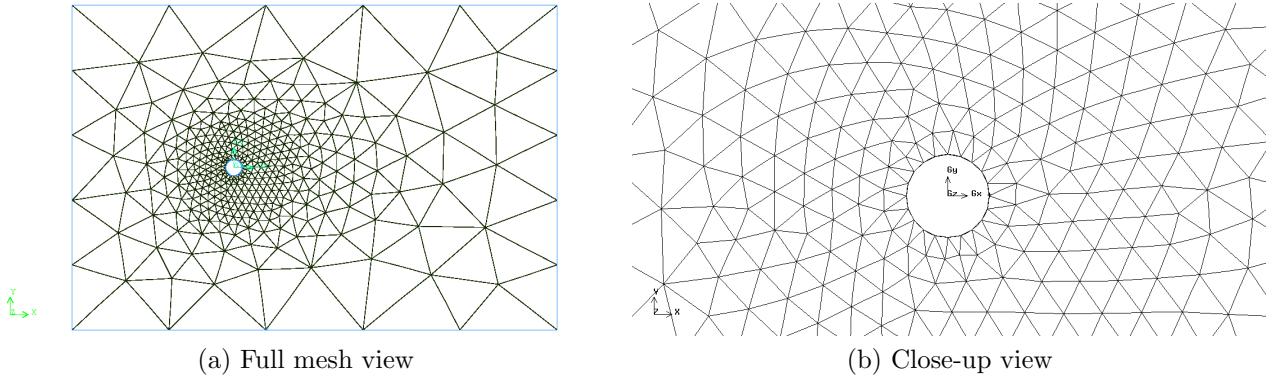


Figure 4.8: Unstructured, coarse mesh of a circular cylinder with 714 triangular elements. Elements adjacent to the cylinder have quadratic edges.

4.4.3 Flow Around a Circular Cylinder, $\text{Re} = 10^6$, $\text{Ma} = 0.2$

This was the first simulation performed after implementing the filters in HiFiLES, so the GPU implementation was not available then. The time-stepping scheme used here is simply forward Euler.

Figure 4.9 shows “pretty pictures” resulting from the simulation. A video of this simulation is linked here.

It is interesting to note that there is a very dissipative form of vortex shedding occurring. The wake region is long, as in lower Re-number cases.

The simulation remained stable throughout and no human intervention was performed while it was occurring, from the start in uniform flow to the moment it was stopped.

From this experiment, it is unclear what portion of the numerical dissipation arises from the coarse discretization and what portion is due to the filtering operation.

This case demonstrates that the stabilization strategy can work well in cases where the mesh is improperly refined: they stabilize the solution and preserve the boundary conditions.

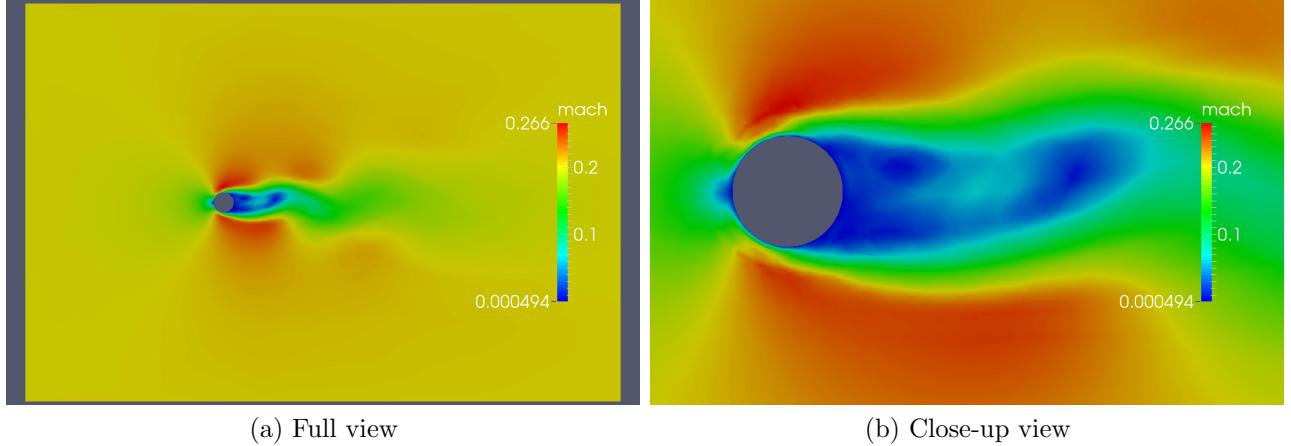


Figure 4.9: Flow past a cylinder. $\text{Re} = 1e6$, $\text{Ma} = 0.2$, $p = 4$

4.4.4 High Reynolds Number, Flow Around a Circular Cylinder, $\text{Re} = 10^6$, $\text{Ma} = 0.077$

This was the first simulation performed using GPUs. The lower Ma case was of interest, as HiFiLES had not been able to run full simulations of flows with $\text{Ma} < 0.2$.

Figure 4.10 shows the colorful results for this case. Once again, the boundary conditions are satisfied and the simulation is stabilized without further intervention. The same time-step size was used as in the previous case in order to leave as many parameters as possible unchanged.

A video of this simulation is linked here in real-time, and here at $0.1\times$. A feature of these simulations that can only be appreciated by watching the linked videos is that the filters have a visible effect on the regions where aliasing and instabilities are expected: the rear part of the cylinder and the boundary layer. However, even though the filters are being applied everywhere, smooth, well-resolved regions of the flow look unchanged. To what extent the smooth regions remain unchanged has not been quantified.

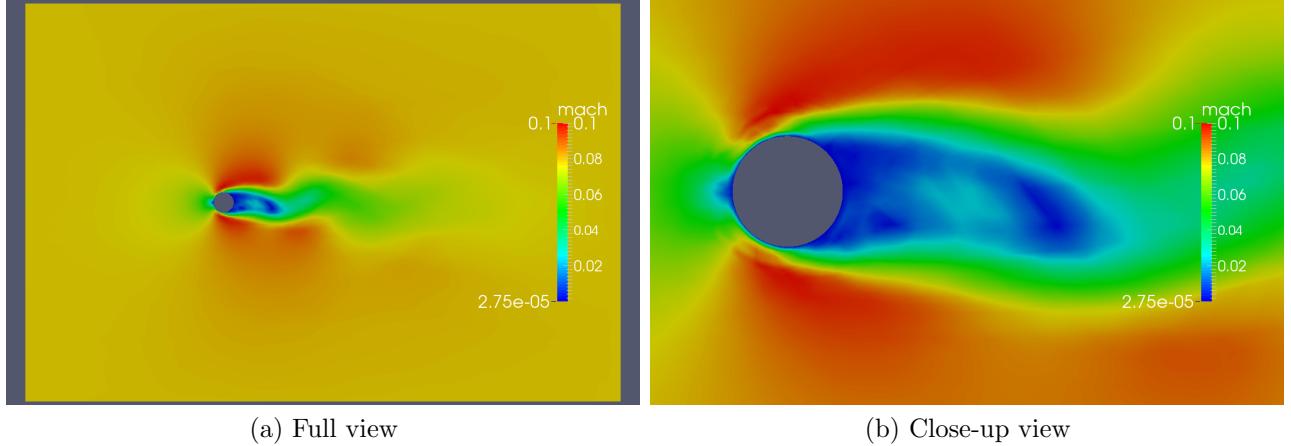


Figure 4.10: Flow past a cylinder. $\text{Re} = 1e6$, $\text{Ma} = 0.077$, $p = 4$

4.4.5 High Reynolds Number, Flow Around a Circular Cylinder, $\text{Re} = 10^6$, $\text{Ma} = 0.87$

This case encompasses all potential sources of instabilities in a high-order solver: poor resolution, aliasing, and sharp gradients. Figure 4.11 shows plots of the solution. The simulation shows a clear un-physical asymmetry due to the coarseness of the mesh. Nevertheless, the no-slip boundary conditions are being satisfied and the shock is present.

Because of the coarseness of the mesh and the high-gradients present in the solution, quite a lot of filtering had to occur. This forced the flow to a “steady state” and shown in the Residual and C_D plots in Figure 4.12.

Values of C_D and residual history are shown in Figure 4.12. The value of drag “converges” after time-step $1.846e6$. The residual in the energy conservation equation also “converges” to a zig-zag pattern after this iteration. Figure 4.12b shows the energy residual in the last few thousand time-steps. The sharp decrease in residual magnitude reveals the time-steps at which the filter is being applied.

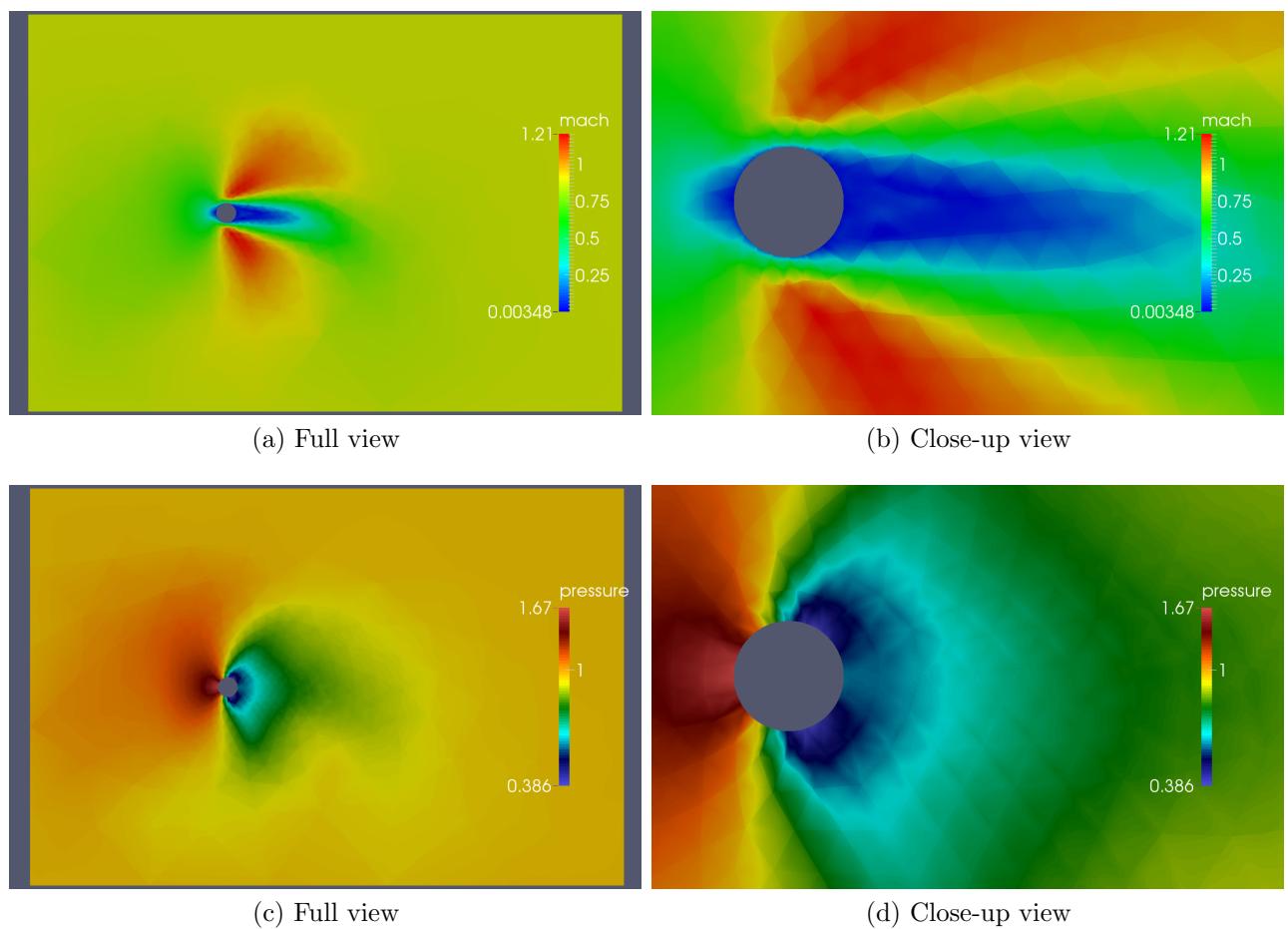
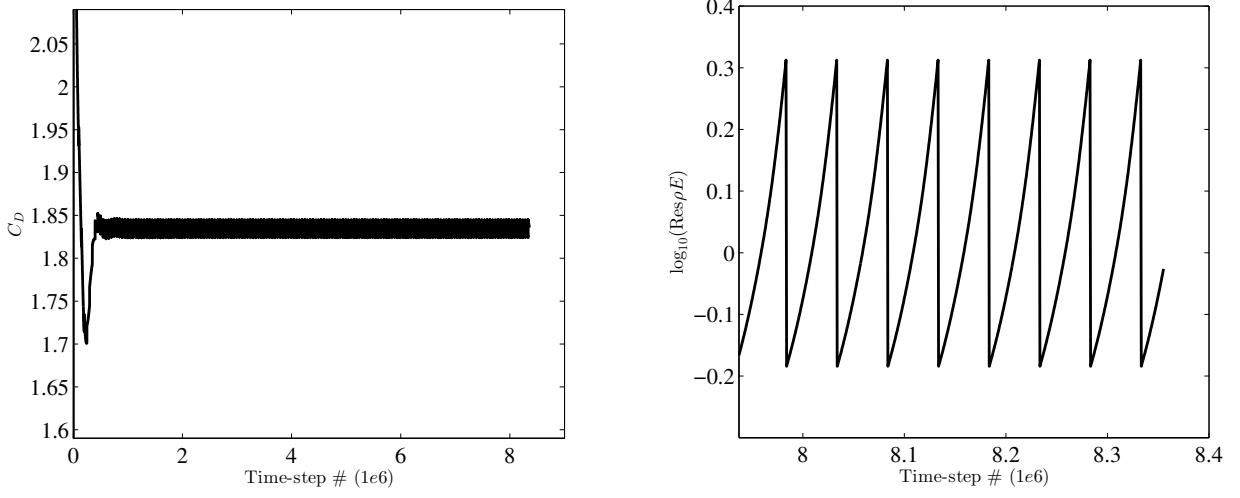


Figure 4.11: Flow past a cylinder. $\text{Re} = 1e6$, $\text{Ma} = 0.87$, $p = 4$



(a) Drag coefficient history over entire simulation run.
 “Steady state” is reached at time-step 1.846e6. (b) Energy residual history over the last few thousand time-steps

Figure 4.12: History of C_D and energy residual of simulation in Figure 4.11

4.4.6 High Reynolds Number, Flow Around a Circular Cylinder, $\text{Re} = 10^6$, $\text{Ma} = 0.0077$, less-coarse mesh

This case was run with the more refined mesh seen in Figure 4.13. This mesh is still coarse for standard turbulent computations. It is interesting to note that the predicted $C_D = 1.18$ and $\text{St} = 0.20$ match experimental results for $\text{Re} = 2e2$. This phenomenon could imply that the grid of the stabilized simulation determines the effective Reynolds number being simulated.

A real-time video of this simulation is linked here for Mach contours, and here for vorticity strength contours. The filters stabilized this almost-incompressible simulation without a problem.

4.4.7 Effects of Filtering in a Well-resolved Simulation

This Section shows that the use of LFS has effects similar to using artificial viscosity. The previous results show that LFS filters can stabilize even the most extreme of circumstances. What about the effect of using LFS filters in more reasonable

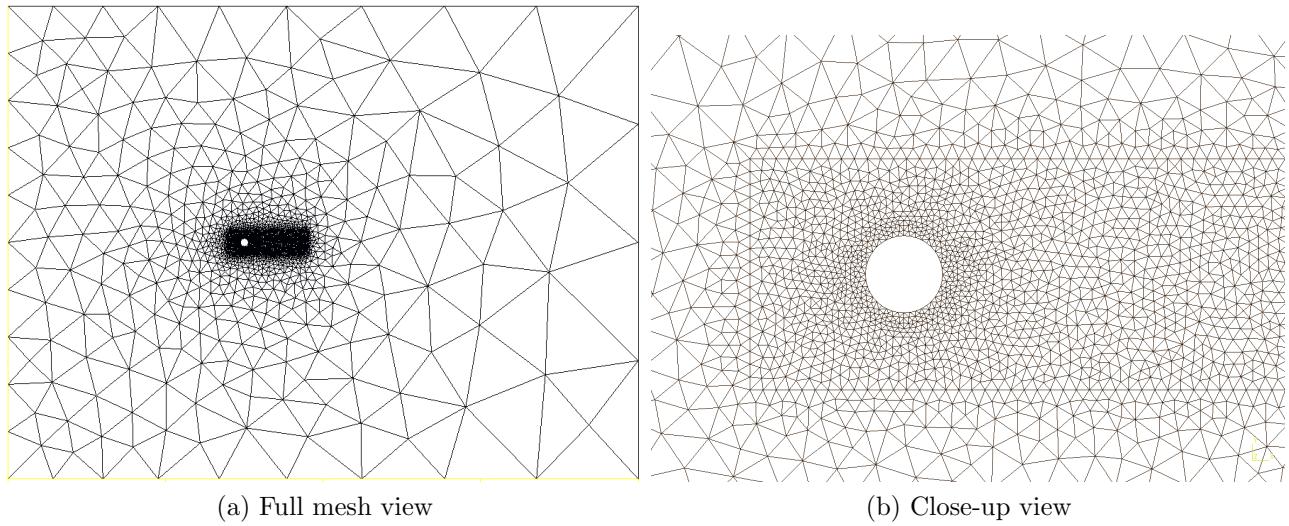


Figure 4.13: Unstructured, coarse mesh of a circular cylinder with 5,616 triangular elements. Elements adjacent to the cylinder have quadratic edges.

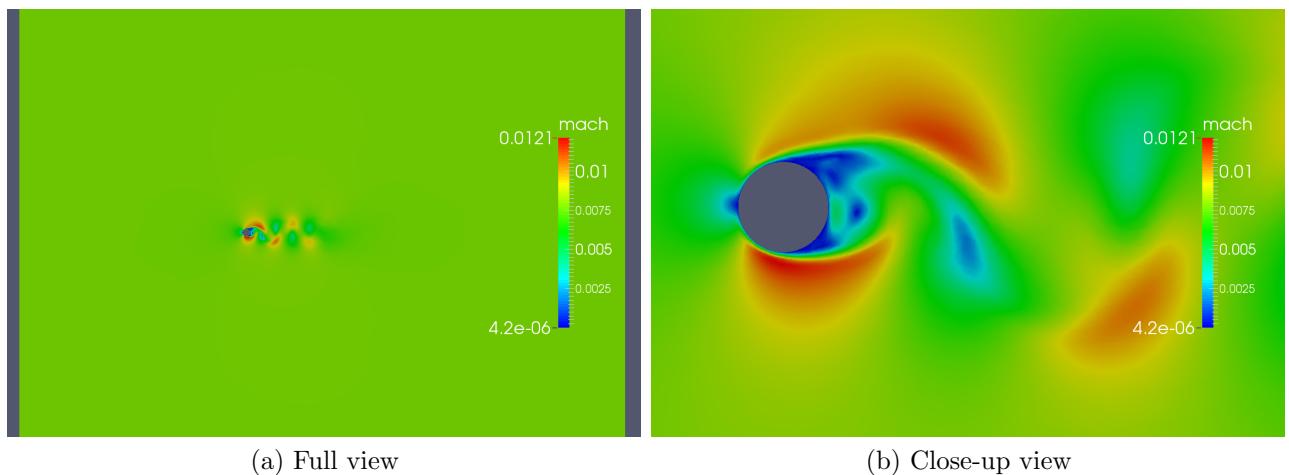


Figure 4.14: Flow past a cylinder. $Re = 1e6$, $Ma = 0.0077$, $p = 4$

conditions?

The case of 2-D flow over a circular cylinder at $\text{Re} = 3,900$, $\text{Ma} = 0.1$ has been selected given the good amount of computational results for it. The thesis by Beaudan [6] provides a very thorough discussion of the physics of the 3-D problem and results of simulations with structured high-order methods.

At $\text{Re} = 3,900$, a proper simulation of flow around a circular cylinder must be 3-D [10]. In fact, previous 2-D simulations of this case have shown the flow to be highly asymmetric [10]. The aim of the simulations presented here is not to predict experimental results, but rather to provide an understanding of how the flow will be affected by the use of LFS filters. We will see that the LFS filters act as artificial viscosity: flow that would otherwise exhibit chaotic behavior becomes periodic. A similar transition from chaotic to periodic behavior due to added dissipation (via turbulence modeling) has been observed in Unsteady RANS (URANS) simulations [19].

4.4.7.1 Setup

We first generate a grid for a 2nd order method that resolves the $y+$ scales around the cylinder. We then generate grids that maintain close to the same number of DoF for a 5th and 6th spatial order of accuracy scheme. We perform the simulation of $\text{Re} = 3.9e3$ past a circular cylinder using the 5th and 6th order schemes in the different grids. The aims of this setup are to:

1. Have a well-resolved, baseline case
2. Visualize the effects of changing the spatial order of accuracy while maintaining the number of DoF almost constant
3. Visualize the effects of using an under-resolved grid
4. Visualize the effects of using LFS filters in different grids with different spatial orders of accuracy

A baseline case with a 6th order accurate in space scheme was attempted while maintaining the same number of degrees of freedom as the case with the 2nd order

scheme. However, the non-linearities within the relatively large elements elements prevented the simulation from completing without intervention. However, a simulation with filtering in the same grid with the same order of accuracy did run to completion.

In addition, to compare the effects of filtering versus running a slightly unresolved simulation, an unfiltered case of a slightly unresolved grid was run with a 5th order scheme.

Finally, a well resolved, yet filtered, simulation with the 5th order scheme was performed.

All cases ran with the same non-dimensional time-step of $\Delta t = 1e - 5$ for a non-dimensional time of $t = 64.7$. Given that $Ma = 0.1$, temperature of air simulated was $300K$, $\gamma = 1.4$, and the reference length was 1 meter, the physical time-step was $2.88e - 6$ seconds and the physical simulation time was 1.86 seconds.

Filtering is performed with a value of $h = 10$ in Equation (4.8) and $\alpha = 0.8$ in Equation (4.4). All conservation fields are filtered every 500 time-steps, so every $5e - 2$ non-dimensional time units, or every $1.44e - 3$ seconds. The filtering procedure was kept the same to isolate the effects of filtering. In practice, a sensor should be used in order to only filter the elements where instabilities could arise.

4.4.7.2 Results and Discussion

Table 4.3 summarizes all the cases run and provides hyperlinks to videos of the resulting flow simulations. The value of St provided reflects the peak St of the lift coefficient power spectrum.

All cases display the following phases: a pair of vortices strengthen behind the cylinder, the vortices elongate and the drag coefficient decreases, asymmetries in the solution trigger vortex shedding, the vortex growth and shedding process transitions into a quasiperiodic (when not filtered) or periodic (when filtered) behavior.

Effect of changing the spatial order of accuracy while keeping the number of DoF constant. Cases A ($P = 1$) and B ($P = 4$) keep close to the same number of DoF. Their lift coefficient power spectra show that the main vortex shedding frequencies are similar. Their drag coefficient figures reveal that Case B, as expected, experiences

Case	P	Mesh	Filtered	Grid	# DoF	C_L and C_D vs. t	St	Video
						Figure		
A	1	4.15	No	4.15	1,579,620	4.19	0.19	link
B	4	4.16	No	4.16	1,398,780	4.20	0.17	link
C	4	4.17	No	4.17	703,260	4.21	0.17	link
D	4	4.16	Yes	4.16	1,398,780	4.22	0.25	link
E	5	4.18	No	4.18	1,362,396	Unstable	N/A	N/A
F	5	4.18	Yes	4.18	1,362,396	4.23	0.21	link
G	5	4.16	Yes	4.16	1,958,292	4.24	0.25	link

Table 4.3: Simulation results that illustrate effect of filtering, changing meshes, and varying the spatial order of accuracy. All cases were run at $\text{Ma} = 0.1$, $\text{Re} = 3.9\text{e}3$.

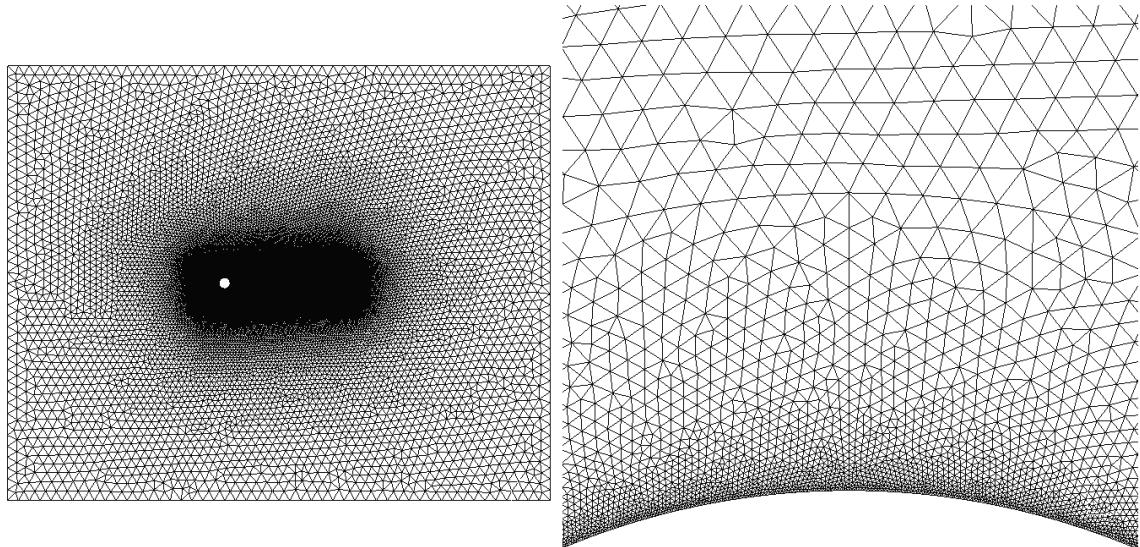


Figure 4.15: Mesh used in case A in Table 4.3. Contains 131,635 triangular elements with second order edges.

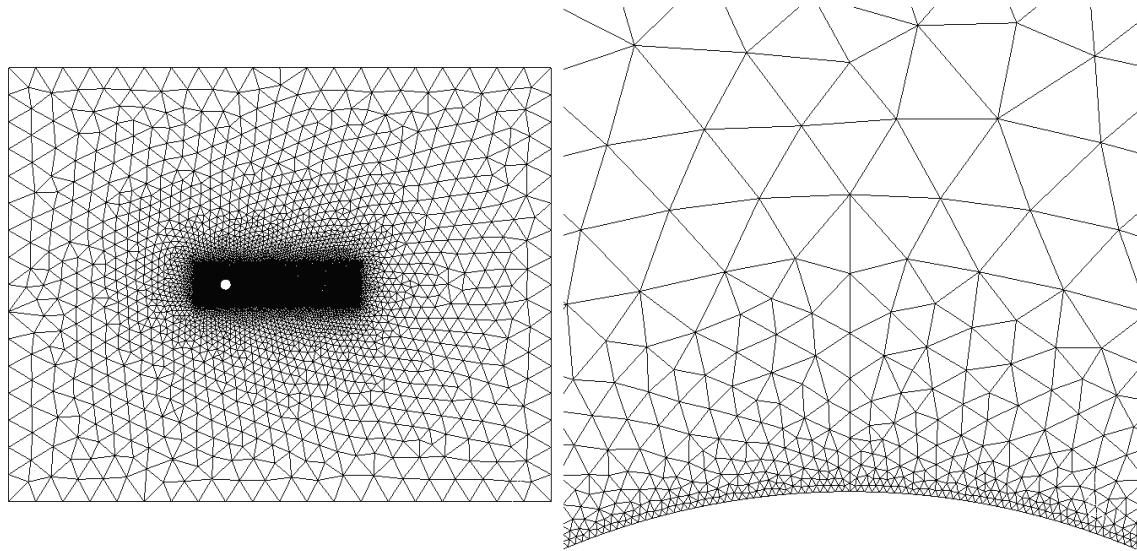


Figure 4.16: Mesh used in cases B, D, and G in Table 4.3. Contains 23,313 triangular elements with second order edges.

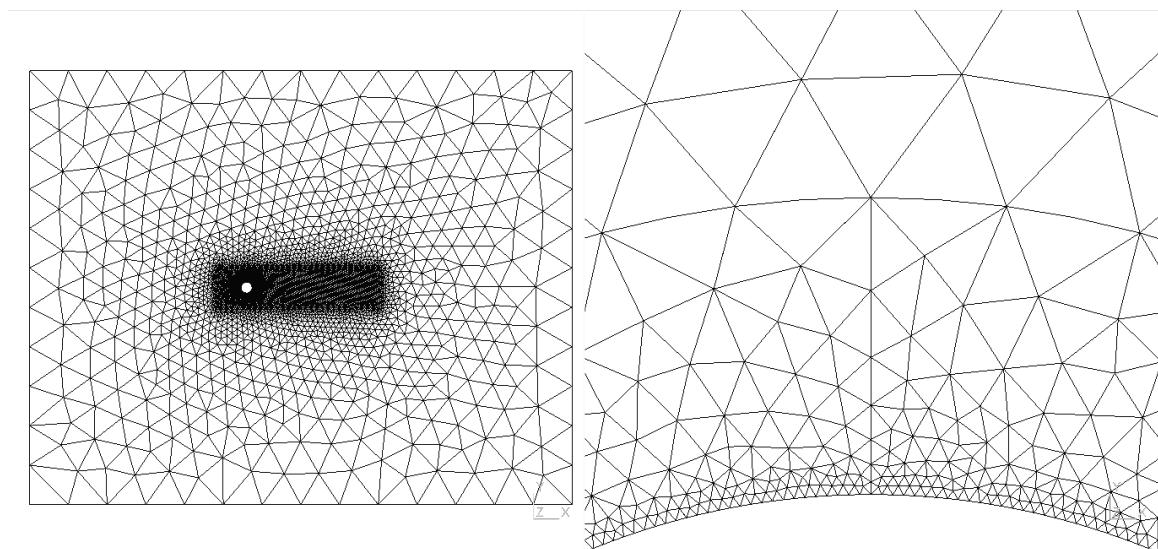


Figure 4.17: Mesh used in case C in Table 4.3. Contains 11,721 triangular elements with second order edges.

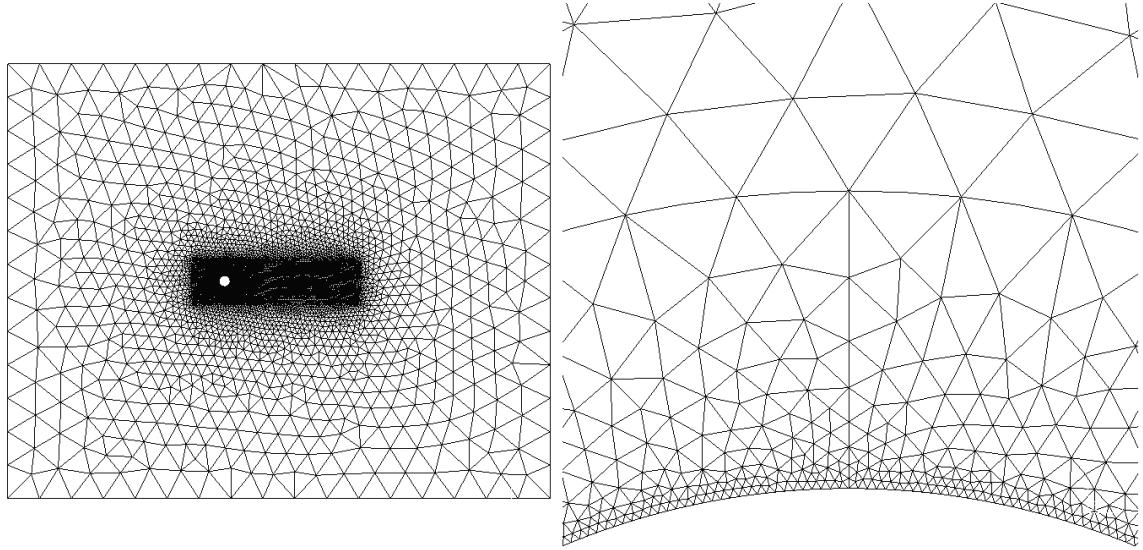


Figure 4.18: Mesh used in cases E and F in Table 4.3. Contains 16,219 triangular elements with second order edges.

less numerical dissipation; the initial vortices detach later than in Case A. In addition, the drag coefficient plot in Case B shows the presence of fairly small structures. Case A seems to diffuse such structures. This effect can be seen in the corresponding videos as well.

Effect of changing the number of DoF while maintaining the spatial order of accuracy constant. Cases B and C differ only in the mesh, Case C has about half as many DoF as B. Both cases have a peak in the lift coefficient power spectrum at $St = 0.17$. As can be seen in the videos and the drag coefficient plots, smaller structures are present in Case B, however, Case C still captures smaller structures than A ($P = 1$) while using half as many DoF and introduces less dissipation as demonstrated by Case C's delayed start of vortex shedding. The strength of the peak at $St = 0.17$ has decreased in Case C. This points to the fact that higher dissipation increases the strength of shearing forces on the vortices, thus prompting earlier detachment and smaller lift coefficient oscillation amplitude.

Effect of using LFS filters. Cases B and D differ only in the filtering. Case D is filtered. The most salient effect of filtering is that the filtered solution starts shedding vortices earlier, sheds vortices periodically (as opposed to quasiperiodically), and very

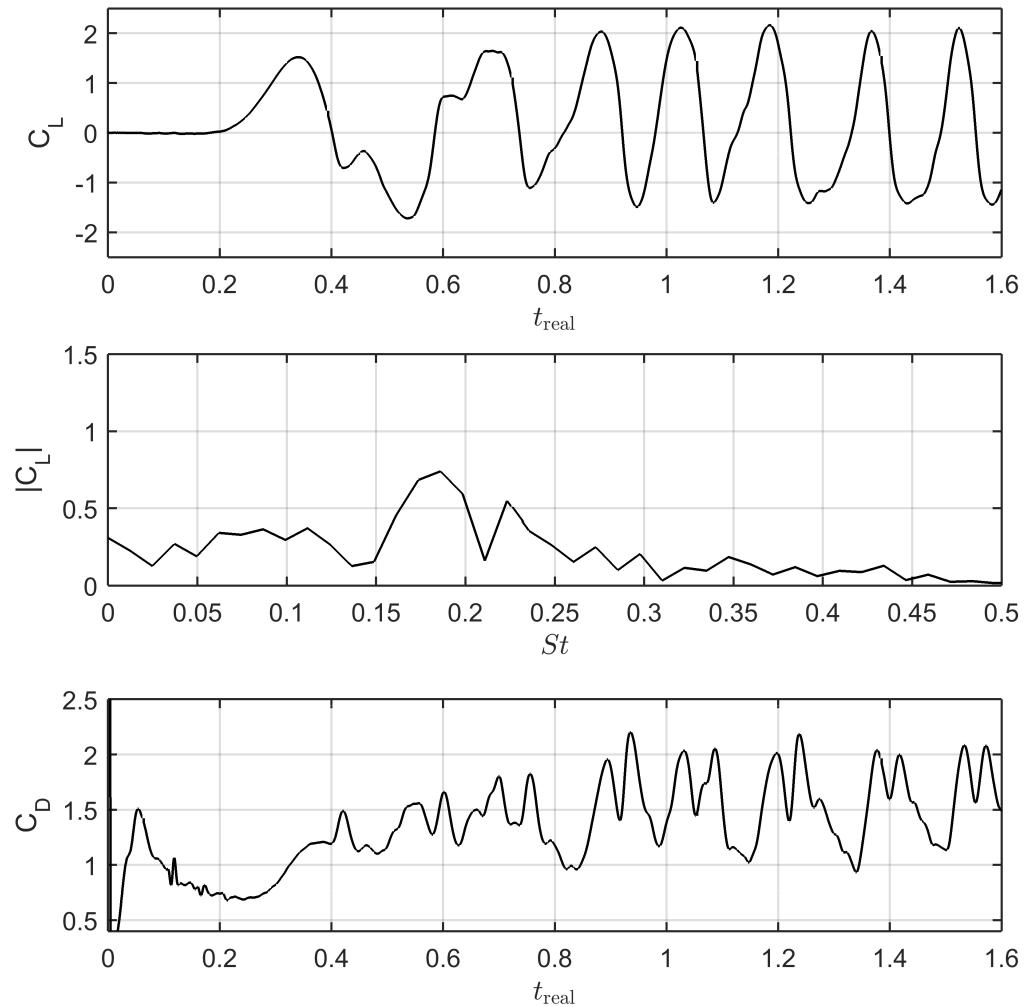


Figure 4.19: Case A in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.15 with $P = 1$.

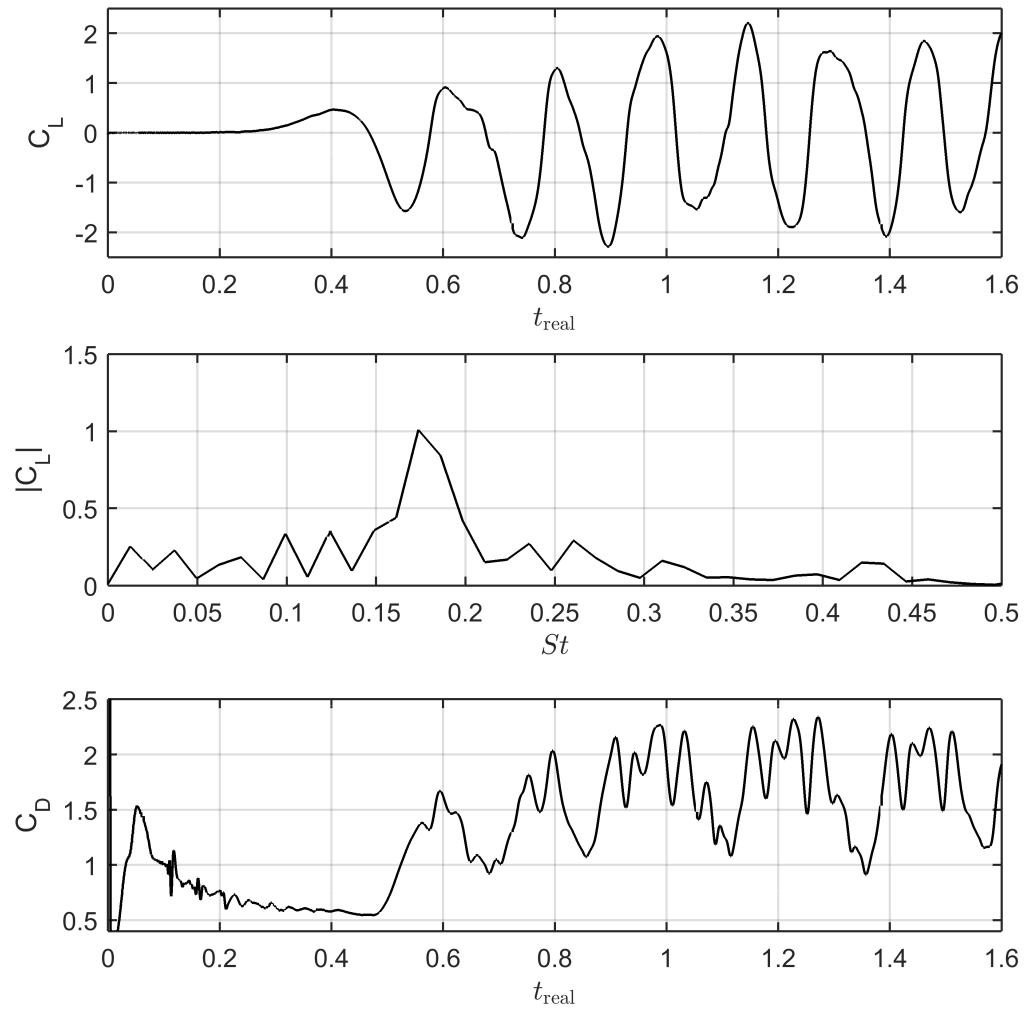


Figure 4.20: Case B in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.16 with $P = 4$.

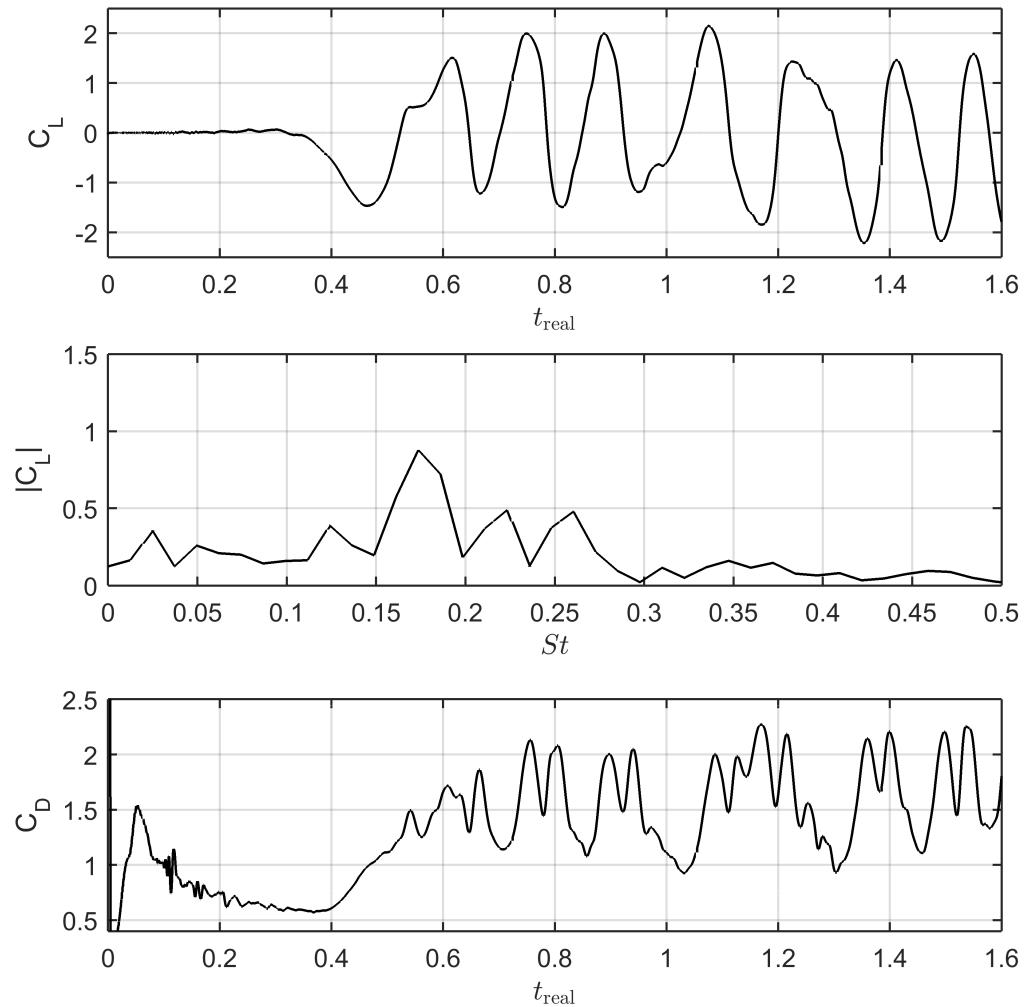


Figure 4.21: Case C in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is not filtered and uses mesh 4.17 with $P = 4$.

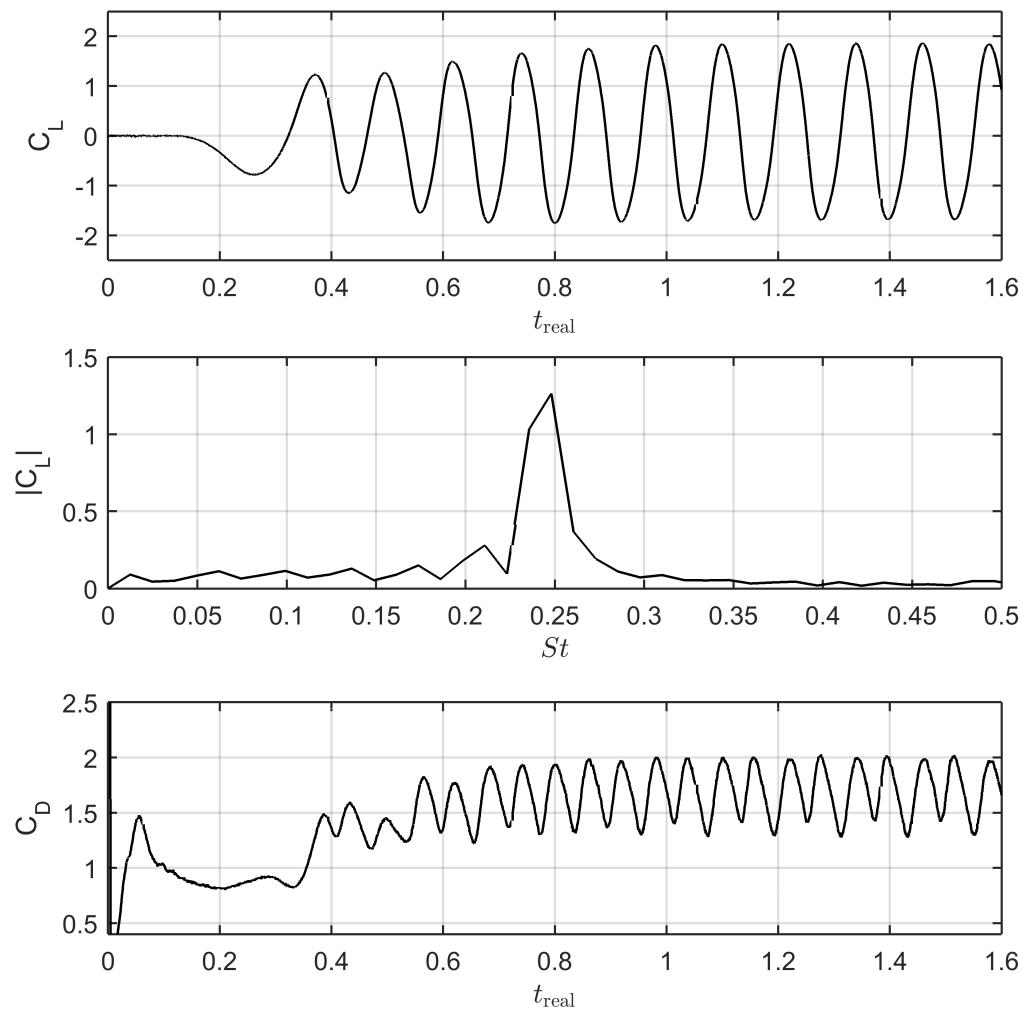


Figure 4.22: Case D in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.16 with $P = 4$.

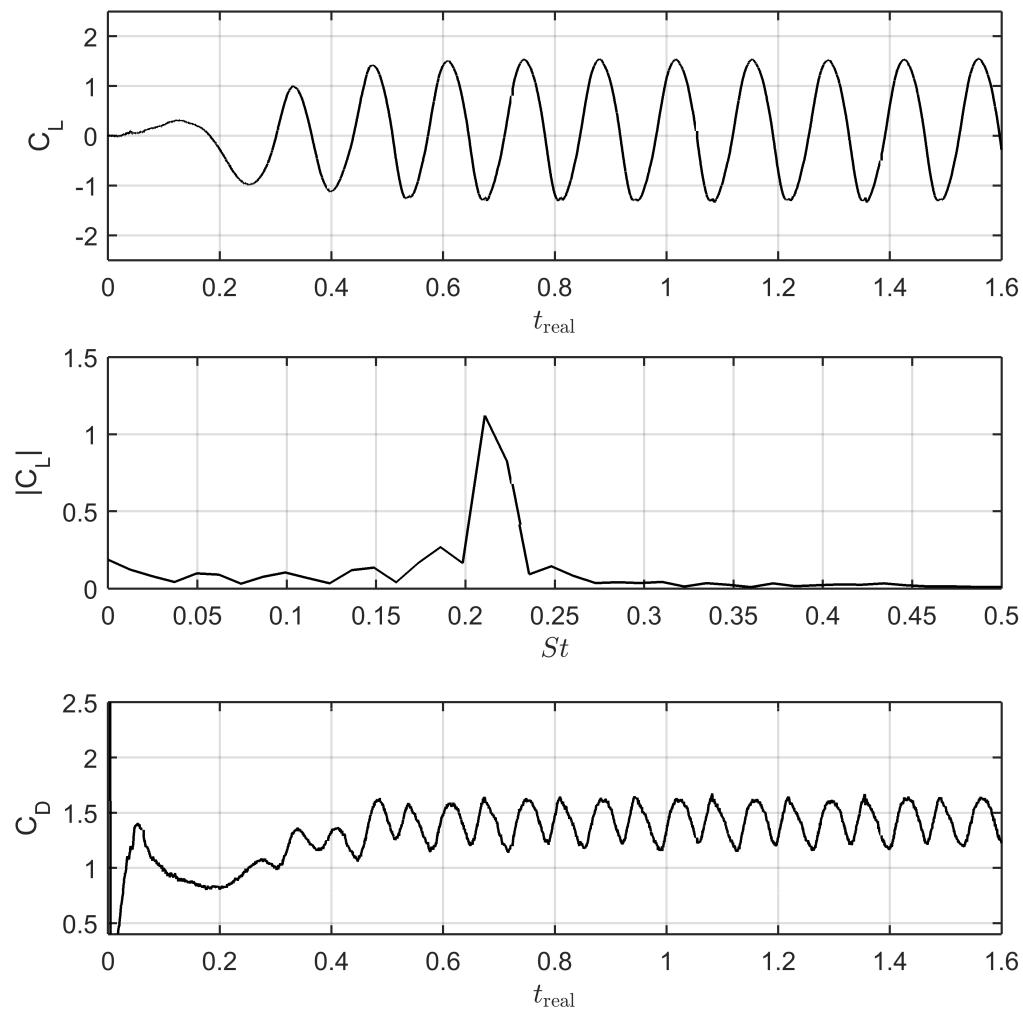


Figure 4.23: Case F in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.18 with $P = 5$.

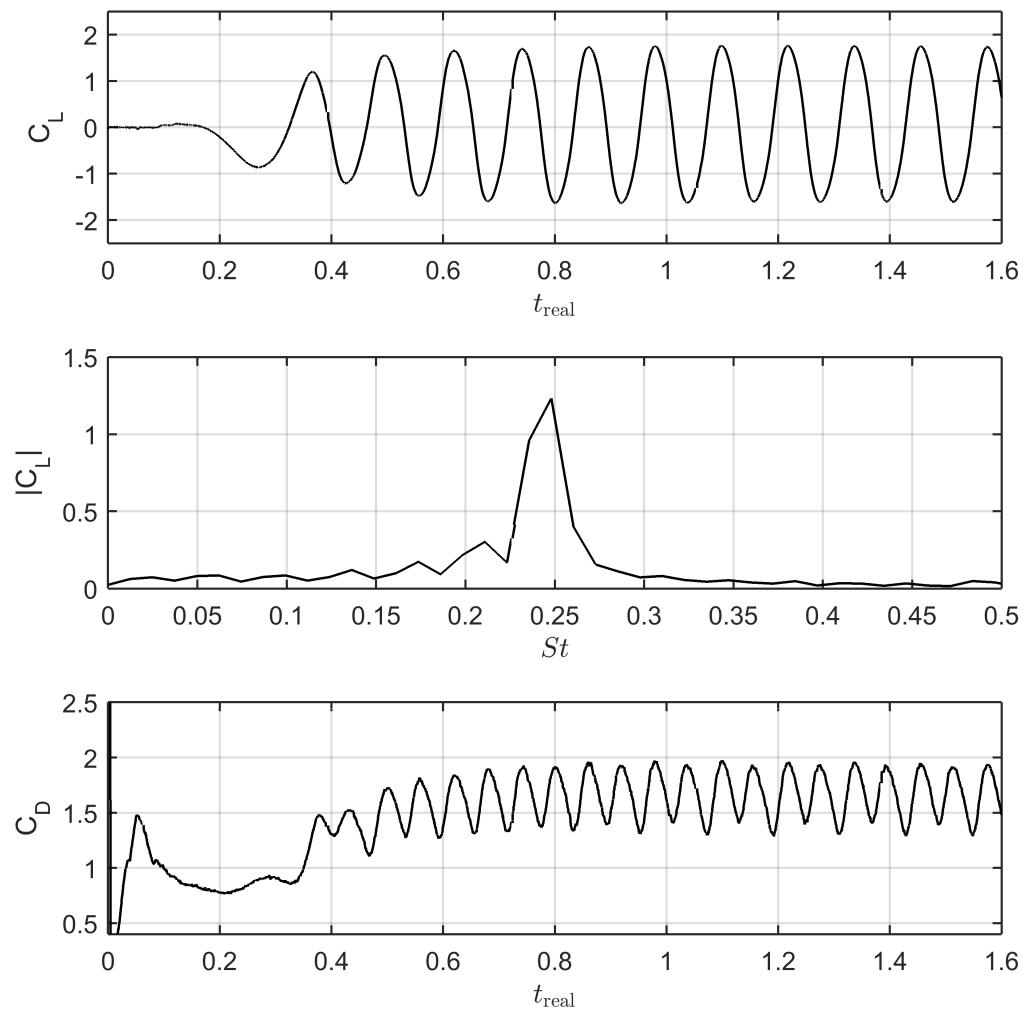


Figure 4.24: Case G in Table 4.3 lift coefficient, St power spectrum, and drag coefficient. Case is filtered and uses mesh 4.16 with $P = 5$.

fine structures are almost non-existent. Indeed the flow looks like the case of $\text{Re}=100$ in Figure 2.13, yet it predicts a higher mean drag coefficient. The power spectrum of the lift coefficient has very little energy at high values of St . This is consistent with the desire of filtering specific frequencies from the simulation. Unfortunately the author could not find studies performed regarding the effect of artificial viscosity on the properties of chaotic flows. Nevertheless, the behavior observed in the filtered solution is consistent with what would be expected of a more viscous flow.

Effect of spatial order of accuracy on filtered simulations. Cases D and G are both filtered and differ only in the spatial order of accuracy. The two simulations are nearly identical. This result is very encouraging: the filtering formulation maintains the spectral properties independent of the spatial order of accuracy. Recall that the filtering matrices are different for the different schemes. This means that the LFS filters' spectral properties can be relied on when developing or using SGS models.

Effect of the mesh on filtered simulations. Cases F and G are both filtered and differ only in the mesh used. The peak St number for the coarser grid, Case F, is lower and its predicted average drag coefficient is lower. This result reflects the general high dependence of simulation results on the grid quality. The grid used in Case F could not produce a stable unfiltered simulation result even with smaller time-steps. It is possible the shift in the drag coefficient is a reflection of the mesh's improper resolution. This shows that the LFS filters could in some cases provide stability at the expense of accurate physics and confirms that LFS filters de-couple stability from proper resolution.

4.4.7.3 Conclusion

In the simulations summarized in Table 4.3, it was possible to isolate the effects of filtering on a $\text{Re}=3.9e3$, $\text{Ma}=0.1$ 2-D flow past a circular cylinder. The unfiltered solutions in different grids obtained with different spatial orders of accuracy displayed minor differences: all unfiltered flows showed quasiperiodic flow with a peak lift coefficient power at a frequency of $\text{St} \approx 0.18$. All filtered solutions exhibited an apparent regime change: the flow became periodic and the peak lift coefficient power occurred at a higher frequency.

It could be surmised that the LFS filters showed the following strengths:

1. The scales filtered by LFS filters are relative to the element size and virtually independent of the order of the basis polynomials, this could be leveraged in the development of SGS models. This was demonstrated by the extremely similar results of Cases D and G, which were performed in the same grid with different orders of accuracy.
2. LFS filters de-couple stability from proper simulation resolution. This could be very helpful when performing LES of turbulent flows. If proper resolution were required for stability, such cases would need to have a resolution close to that of DNS.

The following was identified as a potential drawback:

1. Application of LFS filters in the entirety of the simulation can cause an artificial regime change consistent with what would be expected of increasing viscosity in the flow. This calls for a selective application of LFS filters.
2. Because the spectral properties of LFS filters scale with the element size, results of flows filtered throughout are grid dependent. Once again, the use of sensors could ameliorate this grid dependence.

4.5 Conclusion

We have suggested a formulation of LFS filters for the stabilization of NS solvers for unstructured grids that use a Finite Element Method-based approach to achieve high order spatial discretizations. This includes DG, SD, Spectral Element, and FR methods. The filtering operation can be performed at individual elements and maintains a local stencil by using the element's solution and boundary values. This makes their implementation highly parallelizable.

The filters have been developed with the desired properties shown in Section 4.2.1. In essence, the filters have a spectral interpretation and satisfy boundary conditions

asymptotically. The computational cost of applying a filtering operation to a single element is two small matrix multiplications. This low cost plus the compact stencil makes the LFS filters a good alternative to using artificial dissipation. The main advantage of LFS filters over artificial dissipation is that no modification needs to be made to the partial differential equations being solved.

We have shown by implementing the LFS filters in HiFiLES that little to no tuning is necessary to achieve stability in cases where instability is expected: coarse grids, high-Re flows, high-Ma flows, and low-Ma flows. In all cases, the filters preserved the boundary conditions, did not introduce visible flow anomalies, and allowed the flow to develop its natural features. The summary of results can be seen in Table 4.2.

Because the filter has a physical interpretation, SGS modeling could be done with the classical physical arguments. A similar type of filter has been used by Lodato[50] in the SD scheme to do SGS modeling rather than to stabilize the solution.

The unexpected finding that the filters could bring simulations in very coarse meshes to a pseudo-steady state opens up the possibility of using the LFS filters as part of a pre-conditioning strategy or to start flow simulations from conditions more developed than uniform flow.

All algorithms are publicly available in the HiFiLES repository under the branch “LFS-filters”. The filters have been fully implemented for GPU/CPU computations for triangular grids only.

4.6 Future Work

Implementation in 3D elements is straightforward and shall be the immediate course of action. 3D high-Re simulations had not been possible in HiFiLES and it is expected stabilization with LFS filters will enable them.

So far in all simulations the filters are acting on all elements in the domain with a pre-determined frequency. A more surgical approach to filtering is needed. As shown by Lv et al. [54], localized and selective direct solution manipulation (limiting by multiplying the solution by a scalar while preserving the average within an element)

can preserve the overall order of accuracy. The implementation of an aliasing/shock sensor is in order. Because the filters appear to have little effect on regions where the solution is well resolved, it is acceptable if the sensor is too conservative. The sensor proposed by Persson et al. [69] for general elements and by Sheshadri et al. [75] for tensor-product elements are prime candidates.

The filters are modifying all conservation variables within an element, and very likely key flow quantities like entropy and pressure are being disturbed. This disturbance needs to be quantified.

To prove the usefulness of the LFS filters in a challenging simulation environment, it will be essential to assess their performance in a grid refined properly for the case at hand.

Chapter 5

Conclusion

Spurring the adoption of high-order methods in industry is a process. Not only is it necessary to show the advantages of these numerical schemes in regards to parallelizability, potential for scalability, and accuracy per degree of freedom, but also demonstrate that they can be as *robust* and *usable* as the current industrial tools.

Through the release, maintenance, and support of HiFiLES, an open-source high-order code, the ACL aims to increase *usability*. Feedback from fellow scientists and engineers can only help improve the learning curve needed to use high-order methods for practical purposes. This dissertation has shown validation and verification cases performed in HiFiLES to demonstrate its versatility and provide a guide for researchers interested in performing similar computations.

The development of the CMFR schemes provides a path to tuning dissipation and dispersion properties of a numerical scheme as a simulation progresses. Through solution of challenging 1-D Euler Equations scenarios in 3.6.3, the robustness of C1FR schemes seems to translate to non-linear systems of equations without additional modifications to the flux computations, or the performance of limiting, or filtering.

The LFS filters provide a low-cost stabilization method for all Finite Element Method-based high-order solvers to use in the cases where under-resolution and high gradients can lead to instabilities. These two efforts aim to increase the *robustness* in high-order methods.

5.0.1 Future Work

Usability also relates to an intuitive user interface. I must acknowledge the current workflow in HiFiLES is not very intuitive: the user creates a mesh in some program, then selects parameters in an input file, then runs HiFiLES in the command line. Only advanced users can, as of now, truly take advantage of the power of HiFiLES. Part of future, post-graduation, efforts must include the implementation of a Graphical User Interface. An excellent example of a starting point is provided by Gmsh [31].

The CMFR schemes have variable dispersion-dissipation properties with arbitrary tuning parameters. It is still unclear how much effect each new parameter has on such properties. A Von Neumann study along the lines of that performed by Vincent et al. [84] would help quantify this.

The LFS filters show great promise for their use in industrial simulations with unstructured grids. Interestingly, as shown by Asthana et al. [3], the stronger the filter, the lower the order of the scheme; yet accuracy in high-order, filtered simulations remains higher than in low-order simulations. LFS filters could provide a way to perform grid refinement via polynomial refinement in very large elements without worrying about instabilities. Understanding this phenomenon through grid refinement studies and analysis would be an interesting future research path.

Bibliography

- [1] E Achenbach. Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to $re = 5 \times 10^6$. *Journal of Fluid Mechanics*, 34(04):625–639, 1968.
- [2] Y Allaneau and A Jameson. Connections between the filtered discontinuous galerkin method and the flux reconstruction approach to high order discretizations. *Computer Methods in Applied Mechanics and Engineering*, 200(49):3628–3636, 2011.
- [3] K Asthana, M.R. López-Morales, and A Jameson. Non-linear stabilization of high-order flux reconstruction schemes via fourier-spectral filtering. *Journal of Computational Physics*, 2014.
- [4] Kartikey Asthana and Antony Jameson. High-order flux reconstruction schemes with minimal dispersion and dissipation. *Journal of Scientific Computing*, pages 1–32, 2014.
- [5] J. Bardina, J.H. Ferziger, and W.C. Reynolds. Improved subgrid-scale models for large-eddy simulation. In *13th AIAA Fluid and Plasma Dynamics Conference*, Snowmass, Colo., July 14-16, 1980, July 1980.
- [6] Patrick Beaudan and Parviz Moin. Numerical experiments on the flow past a circular cylinder at sub-critical reynolds number. Technical report, DTIC Document, 1994.

- [7] A.D. Beck and G. Gassner. On the accuracy of high-order discretizations for underresolved turbulence simulations. *Theoretical and Computational Fluid Dynamics*, 27(3-4):221–237, 2012.
- [8] H.M. Blackburn and S. Schmidt. Spectral element filtering techniques for large eddy simulation with dynamic estimation. *Journal of Computational Physics*, 186:610–629, 2003.
- [9] M. Breuer and W. Rodi. Large-eddy simulation of turbulent flow through a straight square duct and a 180 bend. In *Direct and Large-Eddy Simulation I*, pages 273–285. Springer, 1994.
- [10] Michael Breuer. Large eddy simulation of the subcritical flow past a circular cylinder: numerical and modeling aspects. *International Journal for Numerical Methods in Fluids*, 28(9):1281–1302, 1998.
- [11] J.R. Bull and A. Jameson. High-order flux reconstruction schemes for LES on tetrahedral meshes. In W. Haasse, editor, *Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Vol.* Springer, May 2014. unpublished.
- [12] J.R. Bull and A. Jameson. Simulation of the compressible Taylor Green vortex using high-order flux reconstruction schemes. In *AIAA Aviation 2014, 16–20 June, Atlanta, Georgia*, June 2014. unpublished.
- [13] Nicholas K Burgess and Dimitri J Mavriplis. Robust computation of turbulent flows using a discontinuous galerkin method. *AIAA paper*, 457, 2012.
- [14] Mark H Carpenter and Christopher A Kennedy. Fourth-order 2n-storage runge-kutta schemes. *Nasa tm*, 109112:871–885, 1994.
- [15] P Castonguay, DM Williams, PE Vincent, and A Jameson. Energy stable flux reconstruction schemes for advection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 267:400–417, 2013.

- [16] P. Castonguay, D.M. Williams, P.E. Vincent, M. R. López-Morales, and A. Jameson. On the development of a high-order, multi-gpu enabled, compressible viscous flow solver for mixed unstructured grids. In *AIAA Computational Fluid Dynamics Conference, AIAA-2011-3229*, 2011.
- [17] Patrice Castonguay, Chunlei Liang, and Antony Jameson. Simulation of transitional flow over airfoils using the spectral difference method. *AIAA paper*, 4626:2010, 2010.
- [18] Patrice Castonguay, Peter E Vincent, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes for triangular elements. *Journal of Scientific Computing*, 51(1):224–256, 2012.
- [19] Pietro Catalano, Meng Wang, Gianluca Iaccarino, and Parviz Moin. Numerical simulation of the flow around a circular cylinder at high reynolds numbers. *International Journal of Heat and Fluid Flow*, 24(4):463–469, 2003.
- [20] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The runge-kutta local projection discontinuous galerkin finite element method for conservation laws. iv. the multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.
- [21] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iii: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.
- [22] Bernardo Cockburn and Chi-Wang Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework. *Mathematics of Computation*, 52(186):411–435, 1989.
- [23] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

- [24] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261, 2001.
- [25] Bernardo Cockburn, Chi-Wang Shu, and SY Lin. *The Runge-Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws*. Institute for Mathematics and its Applications (USA), 1989.
- [26] J.R. Debonis. Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods. *AIAA Paper 2013-0382*, 2013.
- [27] Bernd Einfeldt. On godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis*, 25(2):294–318, 1988.
- [28] Bernd Einfeldt, Claus-Dieter Munz, Philip L Roe, and Björn Sjögreen. On godunov-type methods near low densities. *Journal of computational physics*, 92(2):273–295, 1991.
- [29] George Elmer Forsythe, Cleve B Moler, and Michael A Malcolm. *Computer methods for mathematical computations*. Prentice-Hall, 1977.
- [30] Marshall Galbraith and Miguel Visbal. Implicit large eddy simulation of low reynolds number flow past the sd7003 airfoil. *AIAA paper*, 225:2008, 2008.
- [31] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [32] Nigel Gregory and CL O’reilly. *Low-Speed aerodynamic characteristics of NACA 0012 aerofoil section, including the effects of upper-surface roughness simulating hoar frost*. HM Stationery Office, 1973.
- [33] Oksana Guba, Mark Taylor, and Amik St-Cyr. Optimization-based limiters for the spectral element method. *Journal of Computational Physics*, 267:176–195, 2014.

- [34] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer Science & Business Media, 2007.
- [35] Justin Hudson. A review on the numerical solution of the 1d euler equations. 2006.
- [36] HT Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. *AIAA paper*, 4079:2007, 2007.
- [37] Hung T Huynh. A reconstruction approach to high-order schemes including discontinuous galerkin for diffusion. *AIAA paper*, 403:2009, 2009.
- [38] Antony Jameson. A proof of the stability of the spectral difference method for all orders of accuracy. *Journal of Scientific Computing*, 45(1-3):348–358, 2010.
- [39] Antony Jameson, Wolfgang Schmidt, Eli Turkel, et al. Numerical solutions of the euler equations by finite volume methods using runge-kutta time-stepping schemes. *AIAA paper*, 1259:1981, 1981.
- [40] Antony Jameson, Peter E Vincent, and Patrice Castonguay. On the non-linear stability of flux reconstruction schemes. *Journal of Scientific Computing*, 50(2):434–445, 2012.
- [41] Botao Jia. Simplex optimization algorithm and implemetation in c++ programming. <http://www.codeguru.com/cpp/article.php/c17505/Simplex-Optimization-Algorithm-and-Implemetation-in-C-Programming.htm>. Accessed: 2015-05-11.
- [42] Andrey Nikolaevich Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(01):82–85, 1962.
- [43] D. A. Kopriva and J. H. Kolias. A conservative staggered-grid chebyshev multidomain method for compressible flows. *Journal of Computational Physics*, 125:244–261, 1996.

- [44] Dmitri Kuzmin and Stefan Turek. High-resolution fem-tvd schemes based on a fully multidimensional flux limiter. *Journal of Computational Physics*, 198(1):131–158, 2004.
- [45] L.D. Landau and E.M. Lifshitz. *Fluid Mechanics (2nd Edition)*. Pergamon Press, 1993.
- [46] Meng-Sing Liou. A sequel to ausm: Ausm+. *Journal of computational Physics*, 129(2):364–382, 1996.
- [47] Yen Liu, Marcel Vinokur, and ZJ Wang. Discontinuous spectral difference method for conservation laws on unstructured grids. In *Computational Fluid Dynamics 2004*, pages 449–454. Springer, 2006.
- [48] G. Lodato, P. Castonguay, and A. Jameson. Discrete filter operators for large-eddy simulation using high-order spectral difference methods. *Int. J. Numer. Meth. Fl.*, in press, 2012.
- [49] G. Lodato, L. Vervisch, and P. Domingo. A compressible wall-adapting similarity mixed model for Large-Eddy Simulation of the impinging round jet. *Physics of Fluids*, 21(3), 2009.
- [50] Guido Lodato, Patrice Castonguay, and Antony Jameson. Structural wall-modeled les using a high-order spectral difference scheme for unstructured meshes. *Flow, Turbulence and Combustion*, 92(1-2):579–606, 2014.
- [51] M López-Morales, Jonathan Bull, Jacob Crabbill, Thomas D Economou, David Manosalvas, Joshua Romero, Abhishek Sheshadri, JE Watkins, David Williams, Francisco Palacios, et al. Verification and validation of hifiles: a high-order les unstructured solver on multi-gpu platforms. In *32nd AIAA applied aerodynamics conference, Atlanta, Georgia, USA*, pages 16–20, 2014.
- [52] Manuel R López-Morales, Kartikey Asthana, and Antony Jameson. Flux reconstruction schemes with corrected fluxes continuous in m derivatives. *in preparation*, 2015.

- [53] Manuel R López-Morales and Antony Jameson. Stabilization of high-order methods for unstructured grids with local fourier spectral filtering: high-re simulations in coarse meshes. In *22nd AIAA Computational Fluid Dynamics Conference*, page 2447, 2015.
- [54] Yu Lv and Matthias Ihme. Entropy-bounded discontinuous galerkin scheme for euler equations. *Journal of Computational Physics*, 295:715–739, 2015.
- [55] D.A. Lyn, S. Einav, W. Rodi, and J.H. Park. A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder. *J. Fluid Mech.*, 304(1):285–319, 1995.
- [56] D.A. Lyn and W. Rodi. The flapping shear layer formed by flow separation from the forward corner of a square cylinder. *J. Fluid Mech.*, 267:353–376, 1994.
- [57] Hanieh Mirzaee, Jennifer K Ryan, and Robert M Kirby. Efficient implementation of smoothness-increasing accuracy-conserving (siac) filters for discontinuous galerkin solutions. *Journal of Scientific Computing*, 52(1):85–112, 2012.
- [58] D Moro, NC Nguyen, and J Peraire. Navier-stokes solution using hybridizable discontinuous galerkin methods. *AIAA Paper*, 3407:2011, 2011.
- [59] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [60] Ngoc Cuong Nguyen, Per-Olof Persson, and Jaime Peraire. Rans solutions using high order discontinuous galerkin methods. *AIAA Paper*, 914:2007, 2007.
- [61] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62(3):183–200, September 1999.
- [62] Michael V Ol, Brian R McAuliffe, Ernest S Hanff, Ulrich Scholz, and Christian Kähler. Comparison of laminar separation bubble measurements on a low reynolds number airfoil in three facilities. *AIAA paper*, 5149(1):2005, 2005.

- [63] Todd A Oliver. A high-order, adaptive, discontinuous galerkin finite element method for the reynolds-averaged navier-stokes equations. Technical report, DTIC Document, 2008.
- [64] F. Palacios, T. D. Economou, A. C. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, A. S. Padron, B. Tracey, A. Variyar, and J. J. Alonso. Stanford University Unstructured (SU²): Open-source analysis and design technology for turbulent flow. In *52nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2014-0243, National Harbor, Maryland, January 2014.
- [65] Francisco Palacios, Michael R Colombo, Aniket C Aranake, Alejandro Campos, Sean R Copeland, Thomas D Economou, Amrita K Lonkar, Trent W Lukaczyk, Thomas WR Taylor, and Juan J Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. *AIAA Paper*, 287:2013, 2013.
- [66] Jeongyoung Park, Kiyoung Kwon, and Haecheon Choi. Numerical solutions of flow past a circular cylinder at reynolds numbers up to 160. *KSME International Journal*, 12(6):1200–1205, 1998.
- [67] Per-Olof Persson. High-order les simulations using implicit-explicit runge-kutta schemes. In *Proceedings of the 49th AIAA Aerospace Sciences Meeting and Exhibit, AIAA*, volume 684, 2011.
- [68] Per-Olof Persson. Shock capturing for high-order discontinuous galerkin simulation of transient flow problems. In *21st AIAA Computational Fluid Dynamics Conference*, pages 1–9, 2013.
- [69] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous galerkin methods. *AIAA paper*, 112:2006, 2006.
- [70] Rolf E Radespiel, Jan Windte, and Ulrich Scholz. Numerical and experimental flow analysis of moving airfoils with laminar separation bubbles. *AIAA journal*, 45(6):1346–1356, 2007.

- [71] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*, 1973.
- [72] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.
- [73] Anatol Roshko. Experiments on the flow past a circular cylinder at very high reynolds number. *Journal of Fluid Mechanics*, 10(03):345–356, 1961.
- [74] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.
- [75] Abhishek Sheshadri and Antony Jameson. Shock detection and capturing methods for high order discontinuous-galerkin finite element methods. *32nd AIAA Applied Aerodynamics Conference*, 2014.
- [76] J. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, 1963.
- [77] Riemann Solvers. Numerical methods for fluid dynamics. *A Practical Introduction*, 1:605, 1997.
- [78] P Spalart and S Allmaras. A one-equation turbulence model for aerodynamic flows. *30th AIAA Aerospace Sciences Meeting & Exhibit*, 1992.
- [79] Michael Steffen, Sean Curtis, Robert M Kirby, and Jennifer K Ryan. Investigation of smoothness-increasing accuracy-conserving filters for improving streamline integration through discontinuous fields. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3):680–692, 2008.
- [80] A Uranga, P-O Persson, M Drela, and J Peraire. Implicit large eddy simulation of transition to turbulence at low reynolds numbers using a discontinuous galerkin method. *International Journal for Numerical Methods in Engineering*, 87(1-5):232–261, 2011.

- [81] O. V. Vasilyev, Thomas S. Lund, and Parviz Moin. A general class of commutative filters for les in complex geometries. *Journal of Computational Physics*, 146:82–104, 1998.
- [82] O. V. Vasilyev, A. L. Marsden, and P. Moin. Construction of commutative filters for LES on unstructured meshes. *Journal of Computational Physics*, 175:584–603, 2002.
- [83] PE Vincent and A Jameson. Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists. *Mathematical Modelling of Natural Phenomena*, 6(03):97–140, 2011.
- [84] Peter E Vincent, Patrice Castonguay, and Antony Jameson. Insights from von neumann analysis of high-order flux reconstruction schemes. *Journal of Computational Physics*, 230(22):8134–8154, 2011.
- [85] Peter E Vincent, Patrice Castonguay, and Antony Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, 2011.
- [86] Miguel R Visbal, Raymond E Gordnier, and Marshall C Galbraith. High-fidelity simulations of moving and flexible airfoils at low reynolds numbers. *Experiments in Fluids*, 46(5):903–922, 2009.
- [87] Miguel R Visbal, PE Morgan, and DP Rizzetta. An implicit les approach based on high-order compact differencing and filtering schemes. *AIAA paper*, 4098:2003, 2003.
- [88] David Walfisch, Jennifer K Ryan, Robert M Kirby, and Robert Haimes. One-sided smoothness-increasing accuracy-conserving filtering for enhanced streamline integration through discontinuous fields. *Journal of Scientific Computing*, 38(2):164–184, 2009.
- [89] Yue Wang and Shuanghu Wang. Arbitrary high order discontinuous galerkin schemes based on the grp method for compressible euler equations. *Journal of Computational Physics*, 298:113–124, 2015.

- [90] Zhi Jian Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.
- [91] ZJ Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, HT Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.
- [92] ZJ Wang, Yen Liu, Georg May, and Antony Jameson. Spectral difference method for unstructured grids ii: extension to the euler equations. *Journal of Scientific Computing*, 32(1):45–71, 2007.
- [93] Michael D White and Miguel R Visbal. Investigation of shock/turbulent boundary interaction unsteadiness with various obstructions. In *46th AIAA Plasma-dynamics and Lasers Conference*, page 3250, 2015.
- [94] David C Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.
- [95] D. M. Williams. *Energy Stable High-Order Methods for Simulating Unsteady, Viscous, Compressible Flows on Unstructured Grids*. PhD thesis, Stanford University, 2013.
- [96] DM Williams, P Castonguay, PE Vincent, and A Jameson. Energy stable flux reconstruction schemes for advection-diffusion problems on triangles. *Journal of Computational Physics*, 2013.
- [97] DM Williams and A Jameson. Energy stable flux reconstruction schemes for advection–diffusion problems on tetrahedra. *Journal of Scientific Computing*, pages 1–39, 2013.
- [98] MM Zdravkovich. Flow around circular cylinders; vol. i fundamentals. *Journal of Fluid Mechanics*, 350(1):377–378, 1997.