AFRL-AFOSR-VA-TR-2019-0100

High-Order Methods and High Fidelity Simulation of Unsteady Turbulent Fluid Flows

Mykel Kochenderfer
LELAND STANFORD JUNIOR UNIVERSITY

03/06/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

DISTRIBUTION A: Distribution approved for public release.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
02/17/2019	Final Report	09/15/2014-12/31/2018

4. TITLE AND SUBTITLE
High-Order Methods and High-Fidelity Simulation of Unsteady Turbulent Flows

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-14-1-0186

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Mykel J. Kochenderfer
Antony Jameson
Freddie D. Witherden

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Stanford University
450 Serra Mall, Stanford, CA 94305, USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
875 North Randolph Street, Suite 325, Arlington, VA 22203, USA

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A: Approved for Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The following report contains a summary of the research carried out in the Aerospace Computing Laboratory under award number FA9550-14-1-0186 from the Air Force Office of Scientific Research (AFOSR). The objective of this research program is to further the state-of-the-art in high-order methods for unsteady compressible flows.

15. SUBJECT TERMS
High-order methods; large eddy simulation; numerical methods; computational fluid dynamics.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Mykel J. Kochenderfer
U	U	U	UU	71	**19b. TELEPHONE NUMBER** *(Include area code)* 650-497-3574

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATE COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33315-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report. e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

High-Order Methods and High-Fidelity Simulation of Unsteady Turbulent Flows (Grant FA9550-14-1-0186)

Principal Investigator: Mykel J. Kochenderfer

Department of Aeronautics & Astronautics, Stanford University, CA 94305

February 14, 2019

Abstract

The following report contains a summary of the research carried out in the Aerospace Computing Laboratory under award number FA9550-14-1-0186 from the Air Force Office of Scientific Research (AFOSR). The objective of this research program is to further the state-of-the-art in high-order methods for unsteady compressible flows.

Contents

1 Introduction

Fluid motion is a complex physical phenomena that can be found in a variety of different engineering applications ranging from the analysis of blood flow in the cardiovascular system to the supersonic flight of aircraft. The primary tool used to analyze this phenomena is computational fluid dynamics (CFD). The modern engineer relies heavily on CFD to obtain an accurate representation of fluid motion. However, current generation CFD software—as deployed in industry—is not capable of accurately predicting transient, highly separated flows over complex geometries. Examples of such flows include an aircraft wing during takeoff or the complex vortex structure generated by the blades of a rotorcraft.

The inability of current generation CFD software to adequately resolve such flows is twofold. Firstly, the numerical methods, typically second order accurate finite volume schemes, have a tendency to be overly dissipative. Hence, they require an excessive amount of resolution in order to successfully track complex flow features over time. Secondly, the methods themselves are not well suited to the requirements of modern hardware platforms which typically include co-processors and GPUs. These limitations have prevented engineers from conducting the types of large eddy simulations (LES) that have the ability to accurately resolve the aforementioned phenomena.

The purpose of this project is to address this issue through the development of high-order methods. High-order methods refer to a collection of numerical schemes whose spatial accuracy is at least third order. Specifically, our focus is on the advancement of the flux reconstruction (FR) approach. The FR approach has the ability to recover a variety of different high-order schemes include the spectral

difference (SD) method and the discontinuous Galerkin (DG) method, along with several new and novel schemes. Along with high-order accuracy—which serves to reduce numerical dissipation—the FR approach is also eminently parallelizable on both CPUs and GPUs.

This report outlines the advances that the Aerospace Computing Lab has made in the development of these methods over the past year. The project continues to improve these schemes on both a theoretical and a practical level.

2 Executive Summary

2.1 Highlights

During the course of the grant the Aerospace Computing Lab:

1. Developed of direct FR (DFR), a simplified formulation of the FR method that greatly reduces the theoretical and implementation complexity of the FR method, along with a novel extension to triangular elements.

2. Investigated an extension of tensor-product FR to triangular elements using collapsed-edge elements.

3. Developed a shock capturing scheme for high-order methods based on a novel concentration sensor.

4. Developed of a novel parallel direct cut algorithm to enable FR to be employed within the context of moving overset grids.

5. Developed a multi-colored Gauss-Seidel scheme to enable implicit time stepping on GPU platforms.

6. Via Fourier analysis gained additional insight into the behaviour of FR for both steady and unsteady advection-diffusion problems.

7. Proved the stability of the VCJH FR schemes on quadrilateral grids for both advection and advection-diffusion problems.

8. Derived a new range of energy stable FR schemes which are conservative and energy stable. These new schemes are shown to be a superset of the existing VCJH schemes.

9. Extended the FR approach to the incompressible Navier–Stokes equations via artificial compressibility combined with polynomial multigrid technology.

10. Demonstrated the scalability of high-order FR on the Titan supercomputer at Oak Ridge National Laboratories with the code, PyFR, sustaining in excess of 13 DP-PFLOP/s on a real-world problem with in excess of 200 billion degrees of freedom.

11. Conducted the first high-order large eddy calculations of static and spinning golf balls, and showed excellent agreement to prior experimental and computational studies.

12. Conducted leadership scale direct numerical simulations of the MTU T161 low pressure turbine cascade at $Re = 200\,000$.

2.2 Publications

Publications supported during the duration of the grant.

Journal papers.

- Energy stable flux reconstruction schemes for advection-diffusion problems on tetrahedra. D. M. Williams and A. Jameson. Journal of Scientific Computing, 2014.

- High-order flux reconstruction schemes with minimal dispersion and dissipation. K. Asthana and A. Jameson. Journal of Scientific Computing, 2014.

- Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. D. M. Williams, L. Shunn, and A. Jameson, Journal of Computational and Applied Mathematics, 2014.

- An extended range of stable-symmetric-conservative Flux Reconstruction correction functions. P. E. Vincent, A. M. Farrington, F. D. Witherden and A. Jameson. Comput. Methods Appl. Mech. Engrg., 2015.

- A Simplified Formulation of the Flux Reconstruction Method. J. Romero, K. Asthana and A. Jameson. Journal of Scientific Computing, 2015.

- On the Stability of the Flux Reconstruction Schemes on Quadrilateral Elements for the Linear Advection Equation. A. Sheshadri and A. Jameson. Journal of Scientific Computing, 2015.

- Non-linear stabilization of high-order Flux Reconstruction schemes via Fourier-spectral Filtering. K. Asthana, M. Lopez-Morales and A. Jameson. Journal of Computational Physics, 2015.

- Explicit filtering and exact reconstruction of the sub-filter stresses in large eddy simulation. J. Bull and A. Jameson. Journal of Computational Physics, 2015.

- A note on terminology in multigrid methods. P. Birken, J. and A. Jameson. Proceedings in Applied Mathematics and Mechanics, 2016.

- A numerical analysis of the nodal Discontinuous Galerkin scheme via Flux Reconstruction for the advection-diffusion equation. J. Watkins, K. Asthana and A. Jameson. Computers and Fluids, 2016.

- On the rate of convergence of flux reconstruction for steady-state problems. K. Asthana, J. Watkins and A. Jameson. SIAM J. Numer. Anal., 2016.

- On consistency and rate of convergence of Flux Reconstruction for time-dependent problems. K. Asthana, J. Watkins and A. Jameson. Journal of Computational Physics, 2017.

- A Direct Flux Reconstruction Scheme for Advection–Diffusion Problems on Triangular Grids. J. Romero, F. D. Witherden and A. Jameson. Journal of Scientific Computing, 2017.

- Evaluation of Fully Implicit Runge Kutta Schemes for Unsteady Flow Calculations. A. Jameson. Journal of Scientific Computing, 2017.

- An Analysis of Stability of the Flux Reconstruction Formulation on Quadrilateral Elements for the Linear Advection–Diffusion Equation. A. Sheshadri and A. Jameson. Journal of Scientific Computing, 2017.

- On the spectrum of the Steger-Warming flux-vector splitting scheme. F. D. Witherden and A. Jameson. Int. J. Numer. Meth. Fluids, 2018.

- A high-order cross-platform incompressible Navier–Stokes solver via artificial compressibility with application to a turbulent jet. N. A. Loppi, F. D. Witherden, A. Jameson and P. E. Vincent. Computer Physics Communications, 2018.

- A parallel direct cut algorithm for high-order overset methods with application to a spinning golf ball. J. Crabill, F. D. Witherden and A. Jameson. Journal of Computational Physics, 2018.

- Preconditioned Smoothers for the Full Approximation Scheme for the RANS Equations. P. Birken, J. Bull and A. Jameson. Journal of Scientific Computing, 2018.

- High-Order Computational Fluid Dynamics Simulations of a Spinning Golf Ball. J. Crabill, F. D. Witherden, and A Jameson. Accepted for publication in Sports Engineering, 2019.

Conference papers.

- Shock detection and capturing methods for high order Discontinuous-Galerkin Finite Element Methods. A. Sheshadri and A. Jameson. AIAA Paper 2014-2688, 32nd AIAA Applied Aerodynamics Conference, 16-20 June 2014, Atlanta, GA.

- Simulation of the Compressible Taylor Green Vortex using High-Order Flux Reconstruction Schemes. J. R. Bull and A. Jameson. AIAA Paper 2014-3210,

7th AIAA Theoretical Fluid Mechanics Conference, 16-20 June 2014, Atlanta, GA, Revised as AIAA Journal, Vol. 53, No. 9, September 2015.

- Mesh deformation and shock capturing techniques for high-order simulation of unsteady compressible flows on dynamic meshes. A. Sheshadri, J. Crabill, and A. Jameson. AIAA Paper 2015-1741, 53rd AIAA Aerospace Sciences Meeting, 5-9 January 2015, Kissimmee, FL.

- Stabilization of High-Order Methods for Unstructured Grids with Local Fourier Spectral Filtering: High-Re Simulations in Coarse Meshes. M. R. Lopez-Morales and A. Jameson. AIAA Paper 2015-2447, 22nd AIAA Computational Fluid Dynamics Conference, 22-26 June 2015, Dallas, TX.

- Mesh deformation and shock capturing techniques for high-order simulation of unsteady compressible flows on dynamic meshes. A. Sheshadri, J. Crabill and A. Jameson. AIAA 2015-1741, AIAA Sci Tech, 53th AIAA Aerospace Sciences Meeting, 5-9 January 2015, San Diego, CA.

- Stabilization of High-Order Methods for Unstructured Grids with Local Fourier Spectral Filtering: High-Re Simulations in Coarse Meshes. M. R. Lopez-Morales and A. Jameson. AIAA Paper 2015-2447, 22nd AIAA Computational Fluid Dynamics Conference, 22-26 June 2015, Dallas, TX.

- The Origins and Further Development of the Jameson-Schmidt-Turkel (JST) Scheme. A. Jameson. AIAA Paper 2015-2718, 33rd AIAA Applied Aerodynamics Conference, 22-26 June 2015, Dallas, TX.

- Application of Dual Time Stepping to Fully Implicit Runge Kutta Schemes for Unsteady Flow Calculations. A. Jameson. AIAA Paper 2015-2753, 22nd

AIAA Computational Fluid Dynamics Conference, 22-26 June 2015, Dallas, TX.

- A study of multigrid smoothers used in compressible CFD based on the convection diffusion equation. P. Birken, J. Bull and A. Jameson. ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering, Crete island Greece, 5-10 June, 2016.

- A High-Order Overset Method on Moving and Deforming Grids. J. Crabill, A. Jameson and J. Sitaraman. AIAA 2016-3225, AIAA Aviation, AIAA Modeling and Simulation Technologies Conference, 13-17 June 2016, Washington, DC.

- Multi-GPU, Implicit Time Stepping for High-order Methods on Unstructured Grids. J. Watkins, J. Romero and A. Jameson. AIAA 2016-3965, AIAA Aviation, AIAA Modeling and Simulation Technologies Conference, 13-17 June 2016, Washington, DC.

- Future Directions in Computational Fluid Dynamics. F. D. Witherden and A. Jameson. AIAA 2017-3791, 35th AIAA Applied Aerodynamics Conference, AIAA AVIATION Forum, 5-9 June 2017, Denver, CO.

- Deep Dynamical Modeling and Control of Unsteady Fluid Flows. J. Morton, F. D. Witherden, A. Jameson and M. J. Kochenderfer. Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018), 2-8 December, 2018, Montréal, Canada.

Dissertations.

- Manuel Lopez-Morales, *Towards industry-ready high-order flow solvers: increasing robustness and usability*. Ph.D. Thesis, Stanford University, 2016.
- Kartikey Asthana, *Analysis and design of optimal discontinuous finite element schemes*. Ph.D. Thesis, Stanford University, 2016.
- Abhishek Sheshadri, *An analysis of stability of the flux reconstruction formulation with applications to shock capturing*. Ph.D. Thesis, Stanford University, 2016.
- Josh Romero, *On the Development of the Direct Flux Reconstruction Scheme for High-Order Fluid Flow Simulations*. Ph.D. Thesis, Stanford University, 2017.
- Jerry Watkins, *Numerical analysis and implicit time stepping for high-order, fluid flow simulations on GPU architectures*. Ph.D. Thesis, Stanford University, 2017.
- Jacob Crabill, *Towards Industry-Ready High-Order Overset Methods on Modern Hardware*. Ph.D. Thesis, Stanford University, 2018.

Book chapters.

- High-Order Flux Reconstruction Schemes. F. D. Witherden, P. E. Vincent, and A. Jameson. Handbook of Numerical Analysis, Vol. 17, Chapter 10, pp. 227-263, Editors: R. Abgrall and C-W, Shu, Elsevier B.V., 2016.
- Aerodynamics. F. D. Witherden and A. Jameson. Encyclopedia of Computational Mechanics Second Edition, Editors: E. Stein, R. de Borst, and T. J. R. Hughes. Wiley, 2017.

- The Design of Steady State Schemes for Computational Aerodynamics. F. D. Witherden, A. Jameson and D. W. Zingg. Handbook of Numerical Analysis, Vol. 18, Chapter 11, pp. 303-349, Editors: R. Abgrall and C-W Shu, Elsevier B.V., 2017.

2.3 Personnel Supported

Personnel supported during the course of the grant.

- Manuel Lopez, Graduate Student, 2015, Stanford University, Stanford, CA
- Jonathan Bull, Postdoctoral Scholar, 2015–2016, Stanford University, Stanford, CA.
- Freddie Witherden, Postdoctoral Scholar, 2016–2018, Stanford University, Stanford, CA.
- Antony Jameson, Professor (research), Summer months (2015, 2016, 2017) Stanford University, Stanford, CA.

2.4 Honors and Awards

- Jacob Crabill: Best poster award at PCCFD 2017.
- Antony Jameson: USACM John von Neumann Medal.
- Antony Jameson: AIAA/ASME/SAE/AHS Daniel Guggenheim Medal.
- Antony Jameson: AIAA Pendray Aerospace Literature Award.
- Antony Jameson: Honorary Fellow of the AIAA.

- Antony Jameson: Royal Aeronautical Society Silver Team award for the aerodynamic design of the Gulfstream G650 (with Dr. Robert Mills).
- Antony Jameson: Best paper award at the 2015 AIAA Applied Aerodynamics Conference for the paper on the origins and further development of the JST scheme.

3 Key Developments

3.1 Overset Methods

3.1.1 High order overset review

Many problems of interest to both industry and academia involve complex geometries and moving grids, such as the problem of simulating a helicopter in hover or flight. Although many methods exist to handle moving bodies within a CFD simulation, the simplest and in some respects easiest approach is the use of overset grids, where each body is meshed independently and assembled into a larger grid system. Once the inter-domain connectivity is established, solution data is interpolated between grids using matching donor and receptor nodes or cells. Each grid is then able to operate independently, essentially unaware of the other grids in the system.

Until recently, work on overset grids has focused on the use of traditional finite-volume or finite-difference solvers, sometimes mixing the two together as is done in Helios [57], where an unstructured finite-volume near-body solver is combined with an high-order adaptive Cartesian off-body solver. Unfortunately, as shown by several groups [39, 56], without a high-order near-body solver, vortical structures

and other flow features incur a rather large amount of diffusion either before or while passing through the overset boundary, causing a large increase in error and limiting the utility of a high-order off-body solver.

High order discontinuous finite element methods have a number of properties which make them attractive solutions to this problem of near-body and inter-grid overset performance. Not only are they less dissipative, meaning they can capture and preserve vortex-dominated flows with less degrees of freedom, they have also been shown to give extremely good results when used in the context of implicit large eddy simulation (ILES) or direct numerical simulation (DNS) [43, 52]. Furthermore, due to the compact nature of the polynomial representation in each element, they can be used quite naturally in an overset framework.

Galbraith [19] first used this property of the DG method to naturally extend it into the Artificial Boundary (AB) overset method. In the AB method, once the overset hole cutting has been performed (removing any elements either inside of solid boundaries or otherwise not required for the simulation), the newly created boundary faces become artificial boundaries. These artificial boundaries are treated by the solver nearly identically to standard inter-processor 'MPI interfaces' (when the Message Passing Interface (MPI) standard is used), with solution (and gradient, if needed) data interpolated to the 'right' side of each face and used in an approximate Riemann solver as usual. The polynomial representation of the solution within each element allows for simple interpolation to the flux points along each of these faces.

3.1.2 Previous overset work in the ACL

More recently, the ACL has extended Galbraith's work to moving grids [16, 17], and explored several different interpolation methods that showed some promise

in reducing a measure of the interpolation error. Over the past year, however, the ACL has returned to the original Artificial Boundary method as the most promising method, and focused on using the method with our fast in-house GPU-accelerated solver, ZEFR. Our work began with interfacing ZEFR with the open-source domain-connectivity package TIOGA [47], which had some capability to interface with high-order solvers through the use of callback functions previously developed for interfacing with a 3D DG code [6].

The first task involved modifying or creating callback functions to implement the AB method, which involves functions to create the artificial boundary faces, locate their flux points, and pass interpolated data back to the solver for solver-specific unpacking into face-based data structures, rather than the global element-based solution structure. For our GPU-accelerated solver, this required large amounts of data transfer back and forth from the GPU, along with a rather large amount of CPU-side processing. This resulted in a significant amount of overhead for even static overset cases, where the domain connectivity had to be performed only once during preprocessing. Moving cases were unfeasibly burdened by this overhead.

To reduce the overset connectivity overhead, we began to experiment with methods for porting TIOGA's functionality to GPU architectures. Although significant speedup was achieved by porting the existing algorithms, they were not well suited to the massively parallel GPU architecture. Additionally, the curved grids required for accurate boundary representation in high order methods incurred a far higher cost per element than for standard linear elements. Hence, work was begun on the new Parallel Direct Cut method, which aims to eliminate most of the bottlenecks associated with applying moving overset grids to GPUs.

The Parallel Direct Cut method is based upon the Direct Hole Cut method of

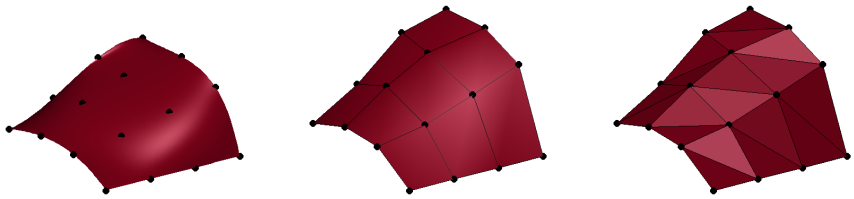
Galbraith [19], an adapted implementation of the direct cut approach introduced by Noack [40]. In a direct cut method, solid or other predefined boundaries in a grid are used as cutting surfaces for other grids. To handle curved grids, the approach Galbraith used to determine what elements were 'cut' by a surface was a Nelder-Mead minimization of the distance between an element and a face of the surface. Our own experiments have shown this method to not be robust in 3D, and furthermore to be quite expensive — acceptable for static grids, but not ideal for a moving-grid overset method.

The approach we are taking instead is to take inspiration from the original purpose of GPUs — namely, computer graphics — and convert the high-order boundary representations into approximate triangulated representations, as shown in Figure 1. All intersection or distance calculation operations can then be performed on sets of linear triangles rather than high-order Lagrange quadrilaterals, and the full 3D Lagrange representation of elements is not required for the majority of the algorithm. Because performing a very large number of triangle–triangle intersection checks is still quite expensive, a hierarchy of approximate distance calculations ranging from basic bounding-box checks to low-resolution quad/hex intersection checks are used to quickly remove elements and faces from consideration and reduce the total cost of the algorithm. For the full details of the algorithm, see [17], [15].

3.2 Moving Overset Grids: Compressible Flow Applications

3.2.1 Static Golf Ball

As a test of our method applied to external flow problems, the first 3D Navier–Stokes test case we shall consider is flow over a golf ball. We first consider a static golf ball,



(a) (b) (c)

Figure 1. Example of splitting a high-order quadrilateral into linear triangles. (a) The exact tensor-product Lagrange polynomial representation; (b) Treating the tensor-product grid as a grid of linear quadrilaterals; (c) Splitting linear quads into triangles.

for which numerous computational and experimental studies exist for comparison. Second, we consider a spinning golf ball to more fully test our method for moving grids and complex fluid dynamics.

The flow physics behind the phenomenon of drag reduction of dimpled spheres—golf balls—has been investigated since at least the 1970s [3, 36]. Early studies relied primarily on wind tunnel experiments, typically collecting force data, and occasionally also performing flow visualizations with, for example, oil streaks. It is only recently, with the advent of large-scale LES and DNS simulations of modest Reynolds numbers, that the accurate, predictive CFD simulation of a golf ball has allowed deeper insight into the effects of dimples.

The typical goal of a golf ball design is to maximize the range it can be driven in a straight line. This primarily leads to the desire to reduce its drag as much as possible, with a secondary goal of minimizing variation in side forces to maintain straight–line flight. Putting backspin on the ball will produce a lift force via the Magnus effect, which extends the flight time and distance of the ball. Drag reduction is mostly due to the dimples, which are sized to create a series of separation bubbles

that will lead to early transition in the unstable shear layer above the bubbles. The exact size, depth, and arrangement of dimples all contribute to the final aerodynamic characteristics of a golf ball under various conditions.

The rotation of a golf ball is typically defined with the non-dimensional spin parameter V/U, where V is the equatorial velocity of the ball and U is the ball's speed of flight. The typical range of rotation speeds for a realistic golf ball range from 2 000 to 4 000 rpm (33 to 66 s^{-1}), giving spin parameters in the range of 0.1 to 0.2 [1, 3, 36]. In contrast to smooth spheres, which exhibit a negative Magnus effect near the critical Re [3, 38], the forces generated by a spinning golf ball increase monotonically with the spin parameter.

3.2.2 Previous Studies

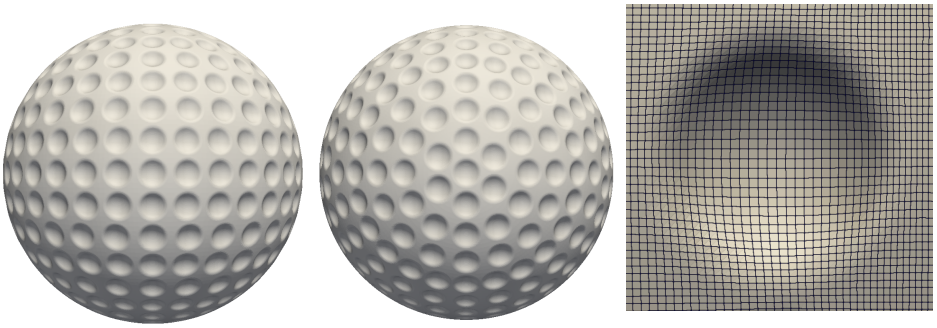
The first noteworthy experimental investigation of the aerodynamics of golf balls under a variety of flow conditions is by Bearman and Harvey in 1976 [3]. Their study used wind tunnel testing of scaled golf ball models to compare the characteristics of round vs. hexagonal dimples, with a smooth sphere used to assess the validity of their experimental setup. The hexagonally dimpled ball also had far fewer dimples than the "conventional" ball (240 vs. 330 or 336). They found that the hexagonally dimpled ball had a lower drag coefficient (C_D) and higher lift coefficient (C_L) over most of the Re and spin rate range of interest, hypothesizing that the hexagonal dimples led to more discrete vortices due to the straight edges of the dimples. The effect of the dimple edge radius was not studied. For both dimple types however, they showed that the dimples serve to reduce the critical Re at which a drag reduction occurs, and that the drag coefficient remains nearly constant for a large range of Re after this point.

Another detailed wind tunnel study was more recently performed by Choi et al. [11]. They studied both fully-dimpled and half-dimpled spheres without rotation. In comparison to Bearman and Harvey, they used only round dimples with a much smaller depth ($k/d = 4 \cdot 10^{-3}$ for Choi et al. vs. $k/d = 9 \cdot 10^{-3}$ for Bearman and Harvey, where k is the dimple depth and d is the sphere diameter) and also with a larger number of dimples (392 vs. approximately 330). Their results showed a slightly higher critical Re (~80 000 vs. ~50 000) with a slightly lower C_D (~0.21 vs. ~0.25) afterwards, with a much more noticeable rise in C_D after the initial drop. Velocity data collected with a hot-wire anemometer was used to confirm that the turbulence generated by the free shear layer over the dimples led to an increase in momentum near the surface of the golf ball after reattachment, and that the separation angle remained at a constant 110^o after the critical Re.

Further studies have been performed using a combination of RANS [50, 51], LES [1, 30, 31], DNS [4, 48], and wind tunnel experiments [1, 26]. Li et al. proposed a link between small-scale vortices created at the golf ball dimples and a reduction in side-force variations at supercritical Reynolds numbers. Also, both Ting [51] and Chowdhury et al. [26] have shown a positive correlation between dimple depth the supercritical drag coefficient, and a negative correlation between dimple depth and critical Reynolds number. Aoki et al. performed wind tunnel experiments on static and spinning golf balls and showed a positive correlation between lift force and spin rate (though smaller in magnitude than the negative lift force generated by a similar smooth sphere), and a slight correlation between drag and spin rate as well.

3.2.3 Golf Ball Geometry

In this study, the golf ball surface geometry was created as a parameterized CAD model with 19 rows of circular dimples (9 rows per hemisphere + 34 dimples around centerline), for a total of 388 dimples, as shown in Figure 2. The golf ball diameter is 42.7mm, the dimple depth is $6.41 \cdot 10^{-4}$m ($k/D = .015$), and the dimple diameter is a constant 2.99mm ($c/D = 7.0 \cdot 10^{-2}$). The dimple edges are filleted with a radius of 0.75mm. The surface was exported in the STL format and used within the multiblock structured mesh generator GridPro [18] to create a spherical grid with a boundary layer. The surface of the golf ball was divided into 24 roughly square regions, each with a resolution of 144×144 quadrilaterals, with 60 layers in the radial direction, for a total of 29 859 840 linear hexahedra, or 1 105 920 cubically curved hexahedra after agglomeration. The first cell height was chosen to be at an estimated y^+ value of 6.667 ($3.4 \cdot 10^{-5}$m), the first 18 layers were held to a constant thickness, and the remaining 42 layers were allowed to grow out to a final outer diameter of 31.82mm. The first cell height was chosen such that after agglomeration into cubically-curved hexahedra and run with 4th order tensor-product solution polynomials, the first solution point inside the element would lie at a y^+ of approximately 1. The surface mesh resolution was chosen to match the recommendations of Li et al. [30], which are based upon recommendations from Muto et al. [38]. The golf ball grid of Li used a surface resolution of less than $\frac{1}{2}\delta_B$, where $\delta_B = 3\sqrt{\frac{D\nu}{2V}}$ is the estimated laminar boundary layer thickness 90^o from the stagnation point [38] (D is the diameter, ν is the kinematic viscosity, and V is the freestream velocity). Here, with $\delta_B \approx 2 \cdot 10^{-4}m$, our surface mesh resolution at the level of the linear grid is slightly more than $\frac{1}{2}\delta_B$, with the final resolution



(a) Side View. (b) Top View. (c) Closeup of mesh within a single dimple.

Figure 2. Golf ball surface. Figure (c) shows the pre-agglomerated element sizes; the actual surface resolution of the golf ball is 4/3 times greater than that shown.

being slightly less than $\frac{1}{2}\delta_B$ once the high-order polynomials are introduced into the agglomerated hexahedra.

The mesh was output in the CGNS structured multiblock format and imported into HOPR (High-Order Pre-Processor) [24], a utility which can agglomerate the cells of a structured mesh into high-order curved hexahedra. The new high-order mesh, in an HDF5-based HOPR-specific format, was then converted into the PyFR mesh format [59], which ZEFR has the capability to read.

This pseudo-structured golf ball grid was then combined with a mostly Cartesian background grid created in Gmsh [22] to fill the desired extents of the full computational domain. The box has a width and height of 0.6832m (16 times the golf ball diameter D), and length 1.0248m ($-12D$ to $12D$). A refined region was created in the area to be occupied by the golf ball, with a refined wake region stretching out to the rear of the domain for a total of 715 750 linear hexahedra elements.

The simulation was advanced in time using the same adaptive RK54[2R+] scheme as before. Third-order solution polynomials were utilized, as 4th order

Table 1. Simulation conditions for all golf ball test cases.

Reynolds number	150 000	ρ	1.0kg/m^3
Mach	0.2	V	1.0 m/s
Prandtl	0.72	P	17.85714286 Pa
γ	1.4	R	17.85714286 J/(Kg K)
L	0.0427 m	T	1 K

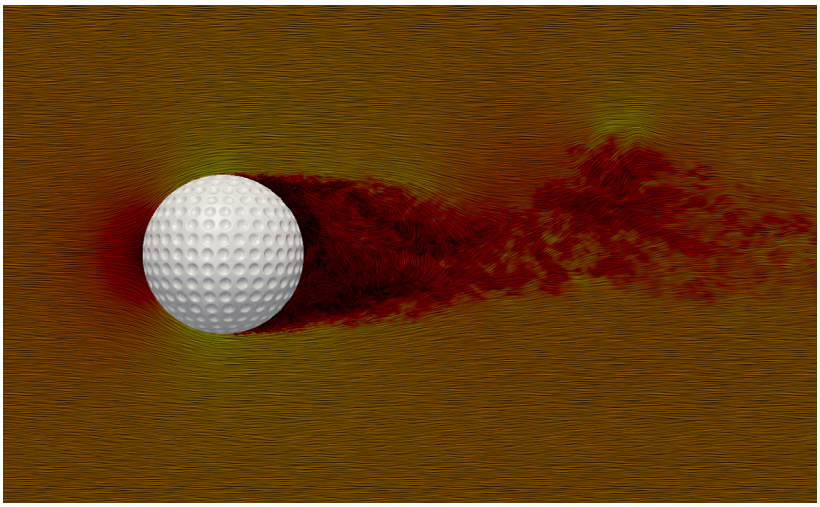
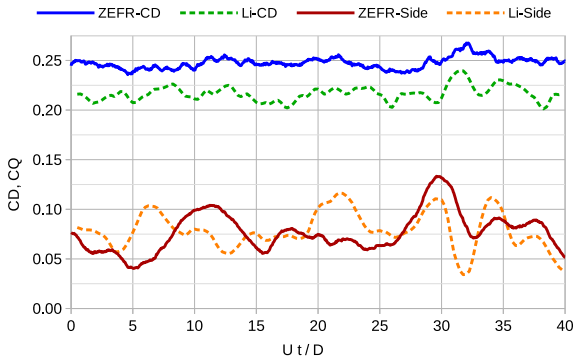


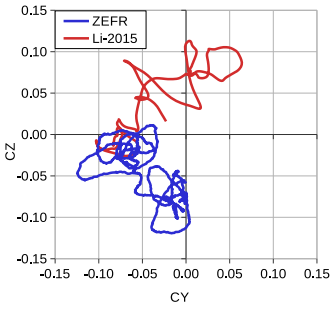
Figure 3. View of approximate streamlines and velocity magnitude field through the $y = 0$ plane (golf ball centerline).

polynomials resulted in too restrictive of a time step on this grid to generate results in a reasonable amount of time. The flow is along the x-axis, with a Reynolds number of 150 000 based upon the golf ball diameter of 0.0427m, and a Mach number of 0.2. The full physical freestream conditions used (scaled such that the freestream velocity is 1) are shown in Table 1. An instantaneous view of velocity contours and approximate streamlines in the mid plane of the ball are shown in Figure 3.

The time histories of the drag and total side forces are shown in Figure 4a,



(a) Time history of force coefficients



(b) Side forces polar

Figure 4. Force coefficients for the static golf ball simulation, compared with those produced by Li et al. under similar conditions.

and a polar plot of the two side forces C_Y and C_Z are shown in Figure 4b. As a verification that our results are reasonable, our force coefficient histories are also plotted alongside those generated by Li et al. [30] for a very similar case. The conditions for their study were $Re = 110\,000$ incompressible flow; a lower Reynolds number than that used here, but still corresponding to the supercritical regime where the drag coefficient should remain nearly constant. A second-order finite-volume LES solver was utilized for their simulation, using implicit time-stepping. Their golf ball had 392 dimples with a dimensionless diameter $c/D = 9.0 \cdot 10^{-2}$ and depth $k/D = 0.005$. An unstructured prism / tetrahedron grid was used with a total of approximately $1.45 \cdot 10^6$ elements in the domain, with overall extents $-13D \leq x \leq 13D$ and $-5.6D \leq y, z \leq 5.6D$.

Since the dimples primarily change the drag, it would be expected that the side forces should be quite similar between the two cases. Indeed, that is the case as shown in Figure 4; the average and standard deviation of the side force histories are nearly identical. The side-force polar plot shows this as well; the two studies

show similar trajectories, simply offset by a rotation about the axis of the flow. The present study was run for much longer (100 passes vs. 40), leading to a more visibly bimodal polar plot, but the trends remain the same. The drag histories are also in agreement; the offset between the two is to be expected, as the dimple depth used here is far greater than the dimple depth used by Li et al. Results from a variety of studies have shown a direct correlation between dimple depth and supercritical drag coefficient, along with an inverse correlation to the critical Reynolds number. Of course, many other factors have an effect on the aerodynamic properties of golf balls, including the number, shape, and placement of the dimples, but here the effects of varying dimple depth are quite apparent.

3.2.4 Spinning Golf Ball

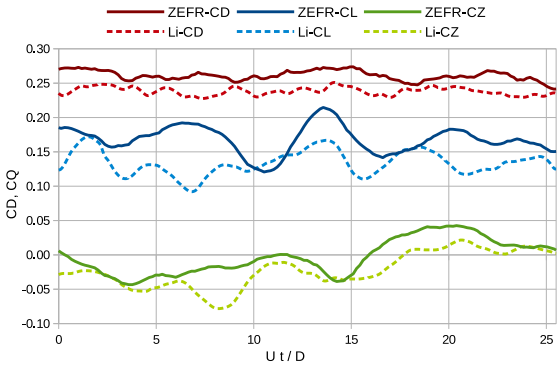
In order to fully test the effectiveness of our moving-grid overset capability, we next move on to the case of a spinning golf ball. We keep the golf ball fixed at the origin, but apply a constant rotation rate around the z-axis; to fall in line with other studies, we choose a non-dimensional spin rate $\Gamma = \omega r / U_{inf} = 0.15$. All other physical flow parameters are left the same. We simplify our handling of moving grids through the use of rotation matrices to map between the updated and original positions of the golf ball grid. To handle arbitrary rigid-body motion, however, the full 6 DOF equations of rigid-body motion are integrated in time to keep track of the current translation and rotation of the inner golf ball grid (although the translation is not modified here, in the future the calculated surface forces on the golf ball could be integrated in time for a full 6 DOF simulation).

Our average C_D values are compared against the results from a number of other studies, both experimental and computational, in Figure 8. As expected from

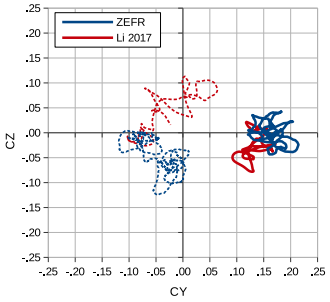
previous literature, the spin induces a slightly higher drag coefficient than the static case but imparts a more regular variation in side forces upon the golf ball; the averages for all force coefficients (with standard deviations) are summarized in Table 2 and the time history is shown in Figure 5. While the out-of-plane side force (C_Z) hovers near zero, the lift (C_L or C_Y) hovers around a value of 0.16, with relatively large low-frequency oscillations. However, looking at a polar plot of the side forces, the oscillations are far more constrained than in the static case, where the symmetric nature of the flow allows the wake to oscillate randomly with no preferred direction. In addition to providing a sizable lift force, the spin has the effect of imposing some structure and a more preferred direction to the oscillations of the wake.

As was done with the case of the static golf ball, we may compare against the prior results of Li et al. [31], who have also performed detailed LES calculations of golf balls at a similar Reynolds number (110 000 vs. 150 000) and spin rate (.1 vs. .15). Figure 5 shows the time history of lift, drag, and side forces between the present results and those of Li et al., along with a polar plot of the lift and side forces. The polar plots from the same cases without spinning are also included for comparison. As in the static case, the present drag value is larger, as should be expected; our present value is about 5% larger than that of the static golf ball. The average lift value is also larger, as is also expected due to the higher spin rate used in the present study.

We can also more quantitatively compare our results to those of Li et al. by using the power spectrum of the golf ball forces, shown in Figure 6. While the two side force spectra in Figure 6c show the same large peak at .04 Hz and nearly identical behavior over the remaining frequency spectrum, the lift and drag coefficients show



(a) Time history.



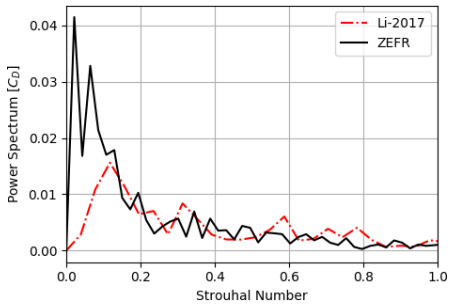
(b) Side forces polar.

Figure 5. Comparison of force time histories for our spinning golf ball vs. the results from Li et al. 2017. The dashed curves in (b) re-plot the same polar data from the static golf ball cases.

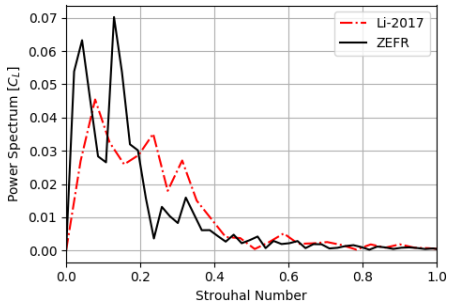
slight differences. In particular the drag spectra show a difference in peak, with the present results showing more low-frequency components. Similarly, the present lift coefficient spectra also show a slight shift to lower frequencies for the two primary frequencies which appear, although the two smaller peaks are in the same locations as those of Li et al.

The present values of lift and drag coefficients agree well with those of Bearman and Harvery. Using the data shown in Figure 7, the estimated CD for a conventional golf ball at a nondimensional spin rate of .15 would be about .28, or 8% higher than that of a static golf ball, with a lift coefficient of about .18. Our results align more with their results at a spin rate of .13, with a lift coefficient of .16 and a spinning-to-static C_D ratio of 1.05. The results of Li et al, meanwhile, predict a much higher rise in C_D with respect to spin rate, with a change of 11% at a spin rate of .1.

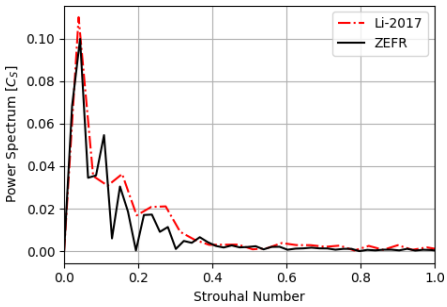
The present results represent a milestone in the use of high-order methods for



(a) Drag Coefficient



(b) Lift Coefficient



(c) Side Coefficient

Figure 6. Comparisons of power spectrum density for our spinning golf ball vs. the results from Li et al. 2017

Table 2. Summary of average force coefficients for the static and spinning golf balls; C_Q refers to the combined magnitude of the lift and side forces C_Y and C_Z. Present results compared to the similar study from Li et al.

	Static	Spinning (Γ = .15)	Li 2015 (Static)	Li 2017 (Γ = .1)
C_D	0.2469 ± 0.005	0.256 ± 0.010	0.217 ± 0.008	0.238 ± 0.0057
C_Q	0.076 ± 0.020	0.165 ± 0.021	0.079 ± 0.019	0.190 ± 0.025
C_Y	−0.047 ± 0.032	0.164 ± 0.021	−0.029 ± 0.045	0.134 ± 0.018
C_Z	−0.044 ± 0.032	0.002 ± 0.022	0.046 ± 0.040	−0.022 ± 0.026

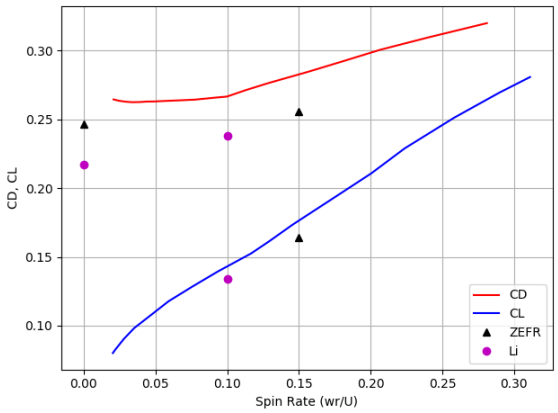


Figure 7. C_D and C_L vs. nondimensional spin rate. Data for conventional golf balls over a wide range of supercritical Reynolds numbers, from Figure 9 of Bearman and Harvey [3]

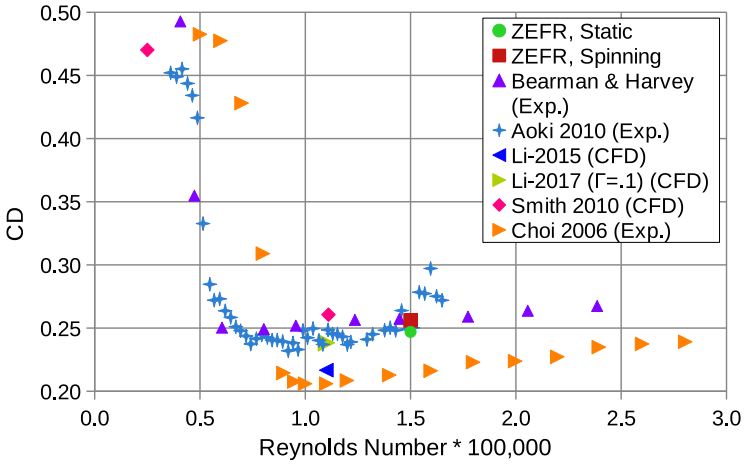


Figure 8. Comparison of C_D values from several experimental and computational studies.

the accurate simulation of difficult, large-scale fluid physics problems including moving objects. These are believed to be the first high order simulations of static and spinning golf balls, and represent an advance in the state of the art in both scale-resolving CFD and overset grid calculations. Many interesting applications are now within reach of high-fidelity simulation using the overset methods described here, including high-lift systems, turbomachinery, and a variety of multicopters and small-scale unmanned aerial vehicles (UAVs).

3.3 Computational Performance of the Parallel Direct Cut Method

It is important to evaluate the computational efficiency of our overset connectivity method, as high accuracy is useless if the method is too expensive to produce a result in a reasonable amount of time. Therefore, in this section, detailed profiling results and performance metrics will be presented on real test cases — specifically, the preceding test cases of the Taylor–Green vortex and the static/spinning golf ball.

An important performance optimization was implemented with respect to the interpolation of solution gradients. ZEFR uses the Local Discontinuous Galerkin (LDG) [12] viscous flux, which in general requires solution gradient data from both sides of every interface. The LDG flux relies on several parameters; in particular, there is a parameter β which can be varied to produce a more upwinded, downwinded, or central viscous flux. For $\beta = \pm 0.5$, the viscous flux requires the solution from only one side of the interface, and the gradient only from the opposite side of the interface. For MPI faces, as discussed by Romero [44], inter-process boundaries can be preprocessed such that one side of the boundary only sends gradient data, and the other side only receives gradient data. We handle overset boundaries similarly: at any artificial boundary face, the value of the LDG parameter

β is set such that the gradient from the 'external' side of the interface is not needed - i.e., the gradient does not need to be interpolated and communicated between grids. Since in 3D simulations the gradient would require 3x as much computation and communication as the solution already requires, this constitutes a significant savings in terms of wall time per time step.

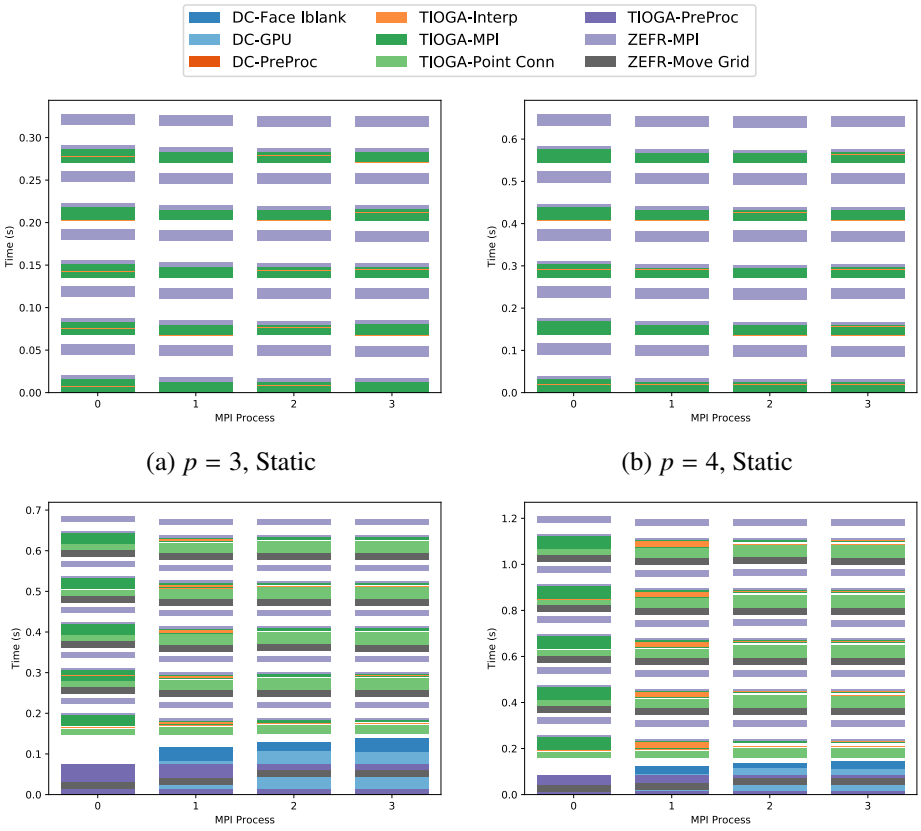
All cases in this section have been run on the Stanford Research Computing Facility's XStream supercomputer, a Cray CS-Storm GPU compute cluster. XStream has a 1.0 petaflops total computing capability comprised of dense nodes with 8 x NVIDIA Tesla K80 GPUs (16 logical GPUs), 2 x 10-core Intel Xeon E5-2680 v2 CPUs, and 1 x FDR Infiniband card. CUDA-aware MPI was enabled to allow the MPI distribution to operate directly on the memory resident on each GPU, and handle the CPU/GPU memory transfers automatically in the background, or even use direct GPU memory access with RDMA or PCI-e.

3.3.1 Performance of the Taylor–Green Test Case

To evaluate the performance of our new algorithm, the previous test cases have been profiled in detail and compared to equivalent non-overset cases. Both static and moving overset grids are compared. For the Taylor–Green case, the grids are partitioned such that the inner grid is run on one rank, and the background grid is run on three ranks. In order to ensure that the GPUs are being utilized efficiently, each rank contains ~46 000 elements; we size the background grid to have 52^3 elements and the inner grid to have 36^3 elements. Each rank has approximately 14 980 000 DOF for $p = 3$ and 29 260 000 DOF for $p = 4$.

Figure 9 shows the timeline of overset-related work performed during one complete time step, consisting of five Runge–Kutta stages, for the static and moving

overset Taylor–Green test cases for two polynomial orders. Since only overset- and MPI-related functions are plotted, the white space in each timeline corresponds to work solely being done in the solver — i.e., the underlying FR operations. Note that non-blocking MPI sends/receives are used throughout ZEFR and TIOGA, so that much of the MPI communication time is overlapped with useful work in the solver. For both $p = 3$ and $p = 4$, the unblanking procedure—moving the grid and performing the hole cutting twice to locate cells which must be added/removed from the grid—takes up the first ~0.15 seconds of the time step, with the remaining time devoted to the residual calculation and Runge–Kutta stage updates. This highlights the efficiency of high-order methods; if coarser grids with higher polynomial orders can be used, the amount of time spent on geometry-related operations relative to the numerical scheme is reduced. The time required to complete the unblank procedure is fairly evenly split between the MPI communication of search points and the construction of the hole maps ('TIOGA-PreProc'), the MPI communication required to ensure consistency in assigning a blanking status (normal/hole/artificial boundary) to all MPI interfaces ('DC-Face Iblank'), and the actual Parallel Direct Cut procedure ('DC-GPU'). Once within the main residual computation, the main sources of overhead are the moving of the grid and corresponding geometry-related updates with ZEFR ('ZEFR-Move Grid'), and the point connectivity process required to find updated donor elements for all fringe points ('TIOGA-Point Conn'), which includes communication of search points, searching the ADT, and computing interpolation weights. On paper the TIOGA data interpolation and MPI communication ('TIOGA-MPI') appear to take a sizable portion of the wall time as well. However, since these operations are overlapped with other useful work within the solver, the actual impact on the runtime is somewhat less than it appears.



(a) $p = 3$, Static

(b) $p = 4$, Static

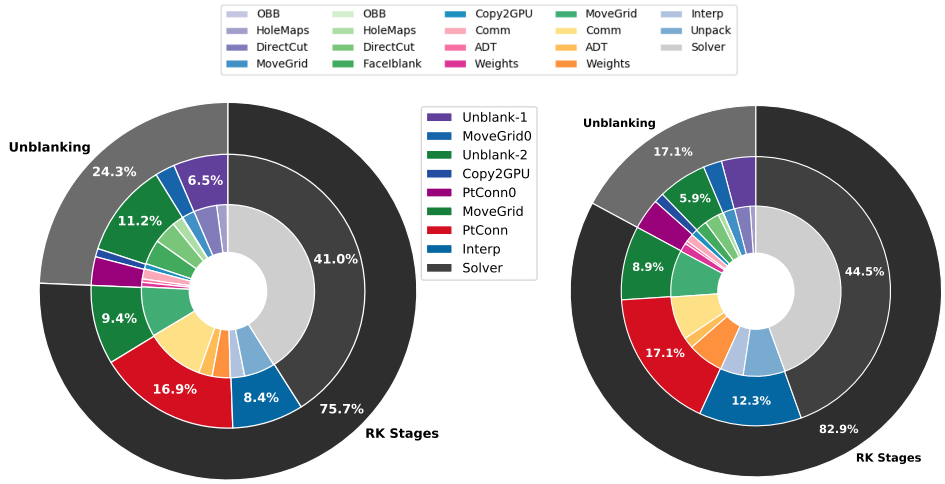
(c) $p = 3$, Moving

(d) $p = 4$, Moving

Figure 9. Timeline of overset grid work during one full five-stage explicit time step over all ranks of the Taylor Green test case. Top: Static overset grids; Bottom: Moving overset grids. The prefix 'DC' refers to work specific to our Direct Cut method; 'TG' refers to standard procedures used within TIOGA; and 'ZEFR' refers to work performed within the solver.

This same data from the moving-grid cases is also plotted proportionally in Figure 10. Here, the timing data from the four MPI ranks have been averaged to get representative values for the overall simulation. The outer ring of each donut plot shows the proportion of time spent on the unblanking procedure vs. the Runge–Kutta stages. The middle ring breaks up the overset work into its high-level components (with the legend in the middle), and the innermost ring breaks down each component further (with the legend at the top). Legend items appear in CCW order in the plots, starting at the top of each ring. 'Unblank-1' and 'Unblank-2' refer to the two stages of the unblank procedure, where the hole cutting is performed for the time step's beginning and ending grid positions. This includes updating each MPI rank's OBB and hole map ('OBB' and 'HoleMaps'), in addition to performing the Parallel Direct Cut procedure ('DirectCut') and determining the final blanking status of all faces ('FaceIblank'). 'PtConn' refers to the point connectivity process, where fringe points are sent to possible donor ranks ('Comm'), donors are found with an ADT search ('ADT'), and interpolation weights are calculated from the donor search results ('Weights'). The final operations displayed are 'Copy2GPU', which is the host-to-device copy of the new iblank values for elements and faces, and 'Interp', which involves both the actual interpolation kernel as well as the MPI buffer packing, sending/receiving, and unpacking time ('Unpack').

The advantages of using higher polynomial orders are quite clear here. Since both cases use the same grid, the unblanking time is nearly identical between the two. However, the unblanking procedure comprises 7% less of the total time for the $p = 4$ case due to the additional work done by the solver. Within the RK stages, the update of the grid positions takes up a substantial amount of time. Since this consists mostly of several matrix-matrix multiplications to update the



(a) $p = 3$, Moving

(b) $p = 4$, Moving

Figure 10. Breakdown of time spent on overset-related tasks for moving grids on GPUs (single node, 4-GPU case). Middle layer: High-level overset tasks; Inner layer: Lower-level subtasks for each slice of the middle layer.

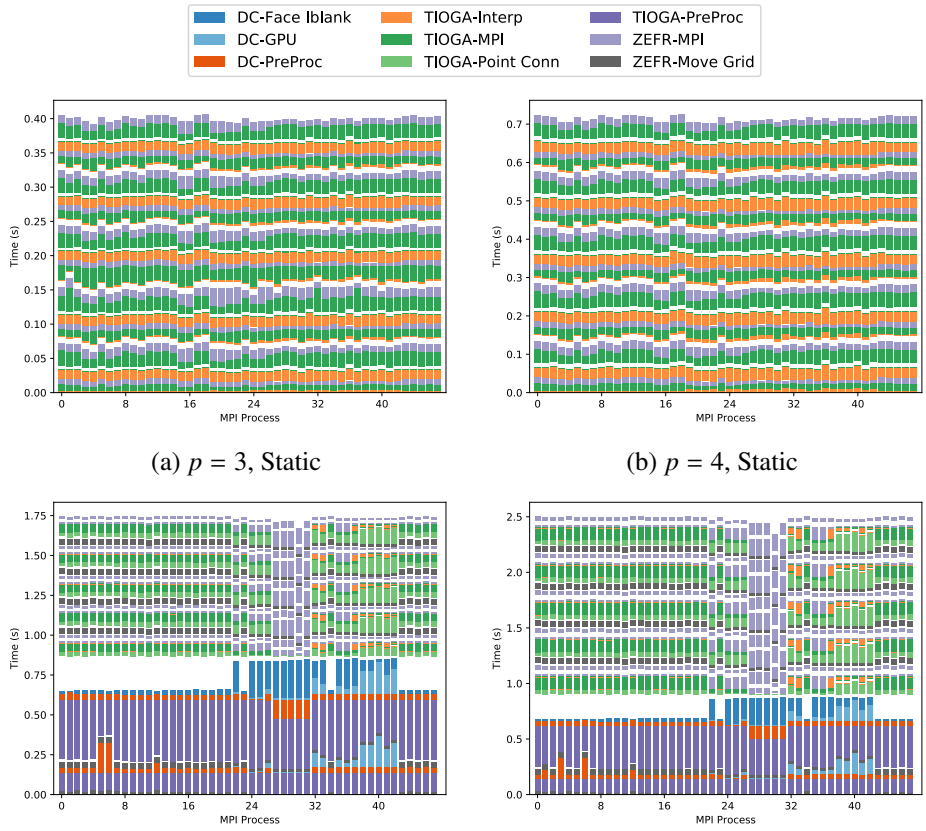
rigid-body dynamics of the grids, there is not much that can be done to reduce this time; it is completely independent of the domain connectivity method. The single most expensive operation (in terms of wall time) during the RK stages is the communication of fringe node positions between each rank and its potential donor ranks. This is because the point connectivity operation within TIOGA is not overlapped with any other useful work; the inter-rank communication latency is not hidden. It may be possible in the future to separate the point connectivity process into two stages, with useful work in between, but at present it would require invasive changes to both ZEFR and TIOGA to implement.

3.3.2 Performance of the Golf Ball Test Case

The large scale of the golf ball test case provides a very useful platform on which to assess the performance of ZEFR and the Parallel Direct Cut method at scale. The golf ball grid uses cubically curved hexahedra, while the background grid is a structured box using linear hexahedra. A total of 715 750 elements are used in the background grid, and 1 105 920 in the golf ball grid. The background and body grids are partitioned into 19 and 29 ranks respectively, giving an average of 37 951 elements per GPU. The first case is run across 3 nodes of Stanford University's XStream cluster, each of which has 8 K80 boards (each with two logical GPUs), 2 Intel Xeon E5-2680 v2 CPUs, and 1 Infiniband card.

Detailed profiling results for both the static and moving cases for two polynomial orders are shown in Figure 11. Similar to the profiling performed for the Taylor–Green test case, for the moving cases, the unblank procedure at the start of the time step is clearly distinguished from the Runge–Kutta residual calculation stages. CPU-side preprocessing activities again take up most of the time of the unblank procedure ('TIOGA-PreProc', 'DC-PreProc'); however in this case the hole cutting does take a noticeable amount of time. It can also be seen that the ADT search and related operations within the point connectivity update procedure ('TIOGA-Point Conn') take up a substantial amount of time during each Runge–Kuta stage.

This is shown more clearly in Figure 12. In contrast to the relatively small and simple Taylor–Green test case, the golf ball domain connectivity takes up a considerably larger proportion of the total time, with the underlying numerical solver comprising just under a quarter of the total time for $p = 4$, and less for $p = 3$. As was the case for the Taylor–Green simulation, the point connectivity procedure



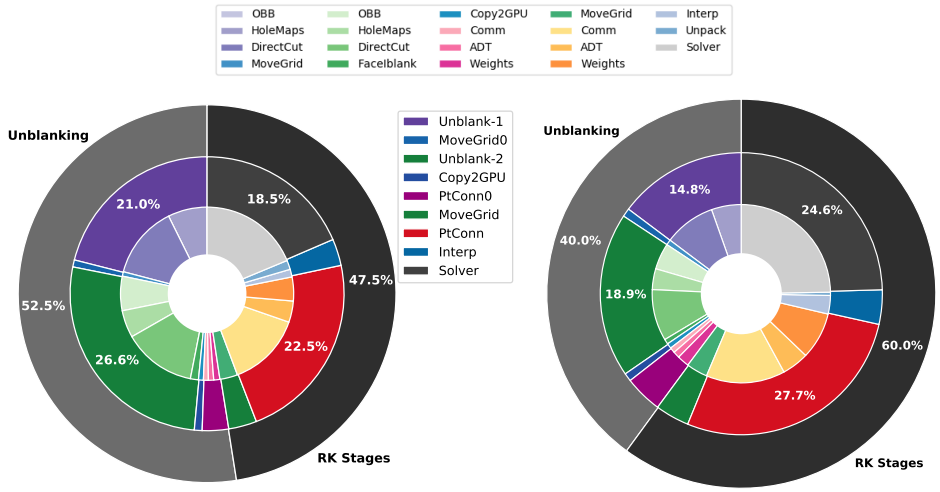
(a) $p = 3$, Static

(b) $p = 4$, Static

(c) $p = 3$, Moving

(d) $p = 4$, Moving

Figure 11. Timeline of overset grid work during one five-stage explicit time step over all ranks of the golf ball test case. Top: Static overset grids; Bottom: Moving overset grids. The prefix 'DC' refers to work specific to our Direct Cut method; 'TG' refers to standard procedures used within TIOGA; and 'ZEFR' refers to work performed within the solver.



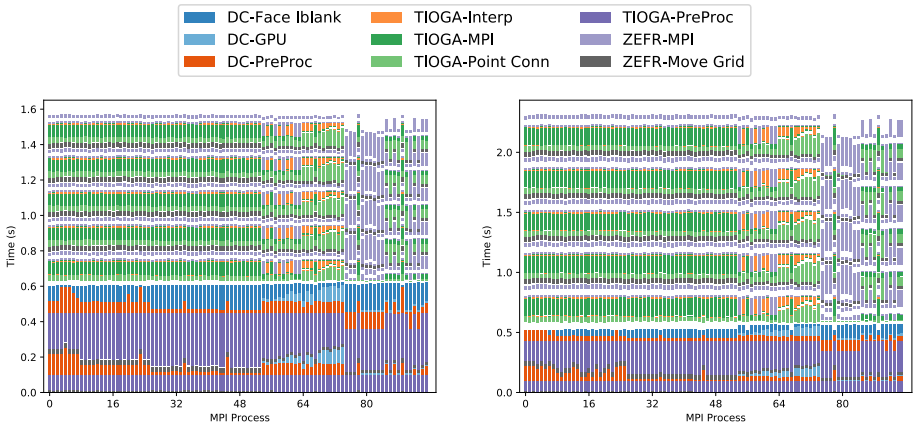
(a) $p = 3$, Moving

(b) $p = 4$, Moving

Figure 12. Breakdown of time spent on overset-related tasks for moving grids on GPUs (3-node, 48-GPU golf ball case). Middle layer: High-level overset tasks; Inner layer: Lower-level subtasks for each slice of the middle layer.

— particularly the MPI communication of the fringe points — is the single largest contributor to the total wall time.

We can further test our method by scaling the case across double the number of computing nodes. Figure 13 shows the timelines of operations for the same moving-grid golf ball test case, but now across 96 GPUs (6 nodes on XStream) for $p = 4$ and $p = 5$. Running the golf ball at $p = 5$ was not possible on 48 GPUs due to requiring more than 576 GB of GPU memory, but here we can see that it runs quite well on 96 GPUs, with an efficiency comparable to that of the $p = 4$ case on 48 GPUs. The $p = 4$ case also scales reasonably well; the next section will discuss quantitative performance metrics.



(a) $p = 4$, Moving

(b) $p = 5$, Moving

Figure 13. Timeline of overset operations for the golf ball case across 96 GPUs.

3.3.3 Summary of Performance Results

As a solver-agnostic, quantitative metric to compare performance between codes, Table 3 shows the average time per DOF required to compute the residual ($\nabla \cdot \mathbf{F}$) and all necessary connectivity and interpolation overhead once. Since we have been using a 5-stage 4th order Runge–Kutta time integration method, this is defined as $1/5$ of the time per DOF required to advance one complete time step. We also compare against the same metric provided by Witherden et al. [60] for the PyFR code under equivalent conditions and on similar hardware (NVIDIA Tesla K40c). Although PyFR and ZEFR are performing essentially identical numerical methods, the ZEFR single-grid performance metrics are slightly better due to the introduction of GiMMiK [62], a library which greatly improves the performance of sparse matrix multiplications such as the ones which occur due to polynomial operations inside of tensor-product elements. These numbers have also been reported for the baseline (non-overset) ZEFR solver within the PhD thesis of Romero [44].

Considering first the Taylor–Green case, we can see that for moving overset grids with linear hexahedra, the overhead involved is a factor of ~2.2 – 2.6 over a single static grid, while the overhead for performing overset interpolation without grid motion and connectivity updates at every time step is only 35%. The 'worst-case' numbers are provided by the moving-overset calculations for the golf ball test case. The case was first run over 3 nodes of the large computing cluster XStream (totaling 48 GPUs), and the body grid was represented with cubically curved hexahedra, increasing the amount of work required in the Parallel Direct Cut algorithm as well as the point connectivity update. In this case, the cost for moving overset grid calculations relative to single-grid calculations becomes a factor of ~3.8 for $p = 4$, or ~5.2 for $p = 3$. The increased apparent cost for $p = 3$ vs. $p = 4$ is because both cases are using the same grid, and hence the overset and geometry-related processing required for each time step is about the same between the two cases, but the $p = 4$ case has about double the number of degrees of freedom, so the total cost is more evenly distributed. The overhead for static overset grids, meanwhile, is at a relatively low 14% for $p = 4$, and 21% for $p = 3$.

When the same case was run with double the number of MPI ranks (6 nodes, 96 GPUs), strong scaling efficiency of 79% was achieved for $p = 4$. If the amount of work is increased by switching to 5th-order polynomials however (a 1.73x increase in the number of DOF vs $p = 4$), then a weak scaling efficiency of 92% is achieved, and is even more efficient than running $p = 3$ on 48 ranks. This again highlights an advantage of high order methods: high polynomial orders combined with relatively coarse grids (compared to a 2nd-order FV grid) can offer greatly improved efficiency in terms of mesh generation and mesh-related operations during the simulation.

Table 3. Time per Runge–Kutta stage per DOF in nanoseconds for various cases. ZEFR used NVIDIA K80 GPUs and Intel Xeon E5-2680 v2 CPUs, while PyFR used NVIDIA K40 GPUs and Intel XeonE5-2697 v2 CPUs.

		Time per DOF / $10^{-9}s$				
		NVIDIA GPUs			Intel CPUs	
Code	Case	$p = 3$	$p = 4$	$p = 5$	$p = 3$	$p = 4$
ZEFR	TGV (Base)	4.28	4.37	–	340	370
(4 Ranks)	Static Overset	5.78	5.86	–	360	380
	Moving Overset	11.1	9.69	–	450	500
(29 Ranks)	Golf Ball (Base)	5.54	5.67	–	–	–
(48 Ranks)	Static Overset	6.74	6.46	–	–	–
	Moving Overset	28.7	21.3	–	–	–
(96 Ranks)	Static Overset	6.91	7.09	8.12	–	–
	Moving Overset	36.2	26.9	23.2	–	–
PyFR	Cylinder	4.88	6.17	–	332	383

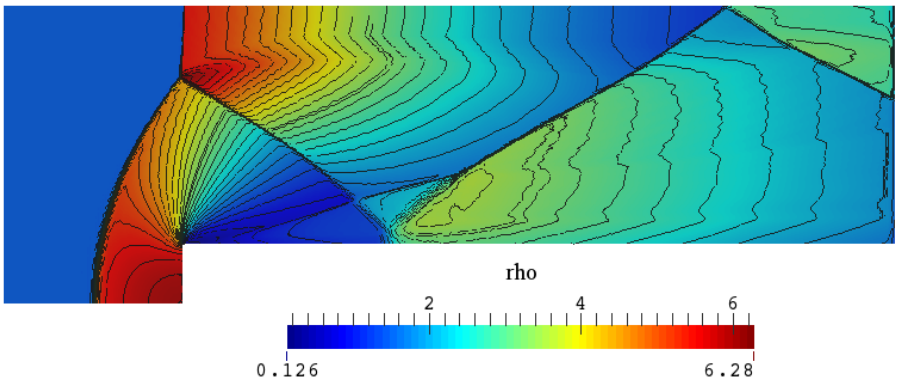
3.4 Efficient shock sensing and capturing

While high order methods have shown great promise of improved accuracy and efficiency over their low order counterparts, they have not yet been widely adopted by the industry primarily due to their lack of robustness and the lack of good shock capturing techniques for these schemes. Discontinuous solutions or shocks are one of the major sources of nonlinear instabilities and high order methods are more prone to such instabilities due to their lower dissipation. Discontinuous solutions carry multiple threats to a numerical scheme. Apart from destabilizing the numerical scheme, they can also leave behind persistent oscillations which can cause non-physical solutions or lead to loss of accuracy around the shocks, or even farther away from them if not handled appropriately. While a wide variety of methods have been developed in CFD for the treatment of shocks, very few are truly suitable for

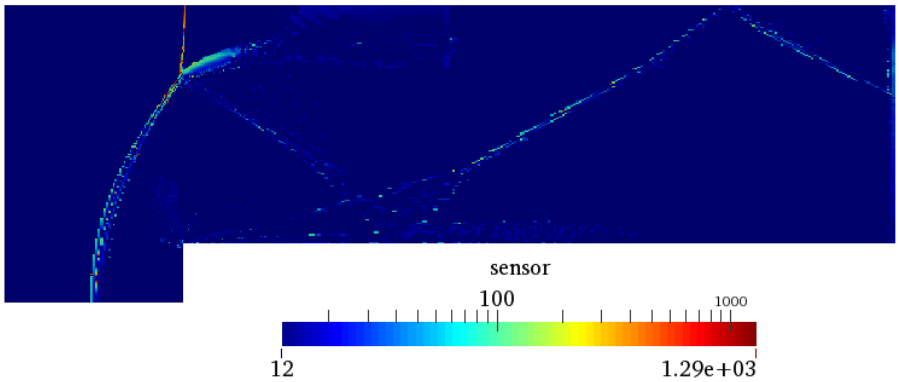
high order unstructured methods. With this as a motivation, a shock detection and capturing method that can be used by any finite element type method and for any nonlinear PDE has been proposed.

The higher resolution available within a cell or element in high order methods allow for a sub-cell resolution of shocks. To achieve this in a computationally efficient fashion, filtering has been proposed as the shock capturing tool. Filtering is a non-intrusive technique where the original PDE remains unaffected and the solution is filtered as a post-processing step after every (or every few) time-steps. This provides a major advantage in terms of computational efficiency over artificial viscosity methods, especially in the context of explicit time-stepping methods. It is also very suitable for implementation on GPUs since all operations can be cast in the form of matrix-matrix multiplications.

The disadvantage of filtering over artificial viscosity methods is the lack of an efficient approach for varying the amount of dissipation across the domain in a smooth fashion. In order to tackle this effectively, a robust shock detection mechanism which can clearly distinguish between shocks and other gradient rich regions like vortices and boundary layers becomes necessary. In this regard, a novel shock detection technique inspired by the method of concentration used in image detection [20, 21] has been developed. While the concentration property of Fourier expansions is used in image edge detection, our method utilizes a similar concentration property of Jacobi polynomials to detect regions with shocks. In comparison to image edge detection, the number of polynomial modes available in the context of CFD is often much lower. A clear guideline for selecting parameters so as to effectively handle this has been laid out and the method has been shown to work very well even at polynomial degrees as low as 2.



(a) Density at flow time t = 4s with 30 contour lines.



(b) Sensor at flow time t = 4s.

Figure 14. Results obtained from the simulation of the flow in a Mach 3 wind tunnel with a step at $t = 4s$.

The positivity preserving limiter of Xiang and Shu [64, 65, 66] which was later extended by Lv and Ihme [34] along with Strong Stability Preserving (SSP) time-stepping schemes have been used to provide robustness against formation of negative or non-physical solutions along with the shock capturing tools. Several numerical experiments were performed to test the robustness of the shock capturing method under a variety of flow conditions and computational meshes, its performance in the context of high order methods, and its ability to perform well for viscous problems in the presence of boundary layers and vortices.

Figure 14 shows the results from a simulation of the flow in a Mach 3 wind tunnel with a step, a test case popularized by Woodward and Colella [61]. The sharp corner is a singularity of the flow and is known to generate gross errors if not treated appropriately. While the standard approach is to implement artificial boundary conditions, the sharp corner is handled directly using our framework composed of sensing, filtering and positivity preservation and the results agree very well with the reference [61]. This test case shows the robustness of this framework in adverse flow conditions and its applicability to a wide variety of compressible flow problems.

Some of the major observations and conclusions from the numerical experiments were as follows:

- The method works very well with both structured and unstructured meshes in 2D and 3D.
- The method was found to be robust even under highly adverse flow conditions and was capable of recovering accurate solutions without the need for mesh adaptation.

- Simulations with the same number of degrees of freedom performed better when higher order polynomials were used even under the presence of multiple shocks. This highlights that although it might be harder to stabilize a high order scheme compared to its low order counterparts, and the order of accuracy is reduced near the shock, if the shock capturing is handled effectively and in a sub-cell fashion, the benefit of higher accuracy away from the shocks provided by high order methods can be retained.

- The shock detector is capable of clearly distinguishing between shocks and regions with vortices or boundary layers and the shock capturing method works well when multiple shocks are present along with such viscous flow structures.

- The shock capturing framework can also be readily utilized for convergence acceleration in steady state problems as well as for stabilizing against aliasing instabilities.

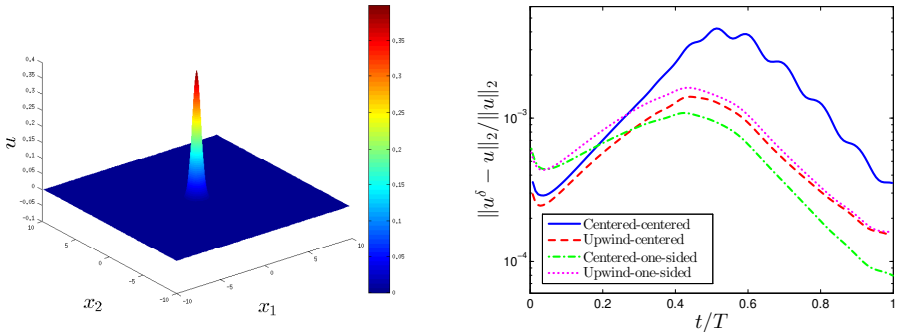
3.5 Fourier analysis of FR for advection-diffusion problems

Numerical analysis of DG and FR methods has primarily relied on functional analysis and spectral decomposition. The primary tool for analyzing dissipation and dispersion properties has been Fourier (von Neumann) analysis. Fourier analysis of the FR formulation was performed for the linear advection-diffusion equation to investigate the stability, dissipation and dispersion associated with the DG scheme. This led to several significant discoveries.

We were able to show that the maximum stable time step for advection-diffusion is stricter than that for pure-advection or pure-diffusion individually. A connection

between maximum stable time steps for pure-advection and pure-diffusion was found through a harmonic sum leading to a suitable CFL estimate for the linear advection-diffusion and Navier-Stokes equations on unstructured, tensor product elements. The estimate has been shown to be conservative on tests with Cartesian grids but not always on unstructured grids.

Our analysis also led to a verification that centered schemes produce less error for well resolved solutions while schemes with a one-sided bias produce less error for solutions that are under-resolved. Figure 15 shows an approximate Gaussian propagated through a 2D domain where elements increase in size, making the wave more difficult to resolve. The centered schemes produce less error for well resolved waves while the one-sided schemes produce less error for waves that are poorly resolved. These trends become particularly important for turbulent flows because a large spectrum of waves need to be captured accurately in order to obtain valid results. These findings show that the combination of the schemes being used can affect the solution.



(a) Initial condition

(b) Relative Error vs. Time Periods

Figure 15. Initial condition and relative error of the propagation of an approximate Gaussian. The advection-diffusion equation is solved using the FR method.

3.6 Stability Analysis of FR on tensor product elements

The linear stability of the FR approach utilizing the Vincent Castonguay Jameson Huynh (VCJH) correction functions has been previously studied in the one dimensional context by Vincent et al. [53] and the scheme has been shown to be stable whenever the VCJH parameter c is non-negative. In fact, the VCJH correction functions were designed to obtain stable numerical schemes. This idea was later extended to triangles by Castonguay et al. [7] and to tetrahedra by Williams et al. [54, 55] who developed new correction functions and criteria required for energy stable schemes on simplex elements.

Parallel to these developments, the FR approach was formulated on quadrilateral and hexahedral elements as a simple tensor-product like extension of the one-dimensional approach utilizing the 1D VCJH correction functions. But the question of whether such an extension is stable remained open until recently, due to certain major difficulties in extending the 1D stability analysis to tensor product elements. In order to overcome these difficulties, a norm different from the one used in 1D and for simplex elements was formulated and it was shown that this partial Sobolev norm is non-increasing as long as the VCJH parameter $c \geq 0$. This conclusion was obtained for both the linear advection and advection-diffusion equations on Cartesian meshes. Since the solution is represented using a polynomial basis, norm equivalence can then be invoked to show that the L^2 energy of the solution cannot grow in an unbounded fashion, thereby proving stability of the numerical scheme.

In addition to proving stability, the newly formulated norm for tensor product elements also displays an explicit dependence on the VCJH parameter c, which is unlike the 1D or simplex element cases where stability analysis only predicted a

bound for c, above which stability of these schemes is guaranteed. This explicit dependence provides an intuition for the results obtained through numerical experiments. In particular, the higher dissipation and correspondingly higher stability of the schemes as c increases can be anticipated directly from the results of the stability analysis. Also, in 1D, the stability analysis breaks as c is decreased below a certain value c_-, but it is not clear whether the schemes would blow up as c is decreased below this value. The results obtained in by us show that when c becomes negative, there is a competing effect between stable and unstable contributions, and that the schemes become less stable as c is decreased.

Although the analysis is focused on quadrilateral elements, the extension to hexahedral elements, i.e., 3D Cartesian meshes is believed to be straightforward when utilizing the new norm formulated for tensor product elements. This answers the last major open question regarding the linear stability of the FR formulation.

3.7 Extended range of energy stable FR schemes

Various properties of FR schemes, including their dispersion and dissipation characteristics and their associated CFL limits are determined in full or in part by the form of the correction function. In 2011 Vincent, Castonguay and Jameson employed an energy method in order to identify a family of provably stable 1D correction functions. Parameterised by a single scalar constant these correction functions provided a substantial degree of insight into the nature of FR.

Over the past year we, in collaboration with Peter Vincent at Imperial College London, have developed a procedure for identifying an extended range of energy-stable correction functions. Unlike the original energy stable FR schemes, which are always paramerized by a single constant, the extended range of schemes are

paramerized by one or more constants depending on the order of accuracy. For all polynomial orders at which we have applied the procedure we have identified the original energy stable FR schemes as a subset of the new extended range. Interestingly, it appears that—unlike the original family of schemes—it is not possible to cast these new extended range of schemes in the form of a filtered DG scheme.

Work to apply these new schemes to real-world problems is still ongoing, both at Stanford University and Imperial College London. The objective of these investigations is to identify parametrizations of these new schemes with particularly favourable properties for high-order turbulent flow calculations.

3.8 Extension of direct FR to triangles

The direct FR approach (DFR) is a simplified formulation of the FR approach that admits a simpler presentation and permits a more efficient implementation. Heretofore, without resorting to a collapsed-edge formulation it has not been possible to extend DFR to triangular elements. Over the past year Romero, Witherden, and Jameson developed a natural generalization of the DFR approach to triangular elements. The resulting scheme inherits many of the desirable properties of the original DFR scheme and is substantially simpler to implement than existing extensions of FR to triangular elements. A key aspect of these schemes is a novel treatment of the auxiliary equation which arises when discretizing problems with a diffusive term.

This new scheme defines a set of N_s solution points $\{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_{N_s}\}$ in the interior of the standard triangular element where $N_S = \frac{1}{2}(P+1)(P+2)$ is the number of points required to interpolate a polynomial of degree P within the triangle. Next, a set of N_F flux points $\{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_{N_S}, \boldsymbol{r}_{N_S+1}, ..., \boldsymbol{r}_{N_F}\}$ which includes the previously

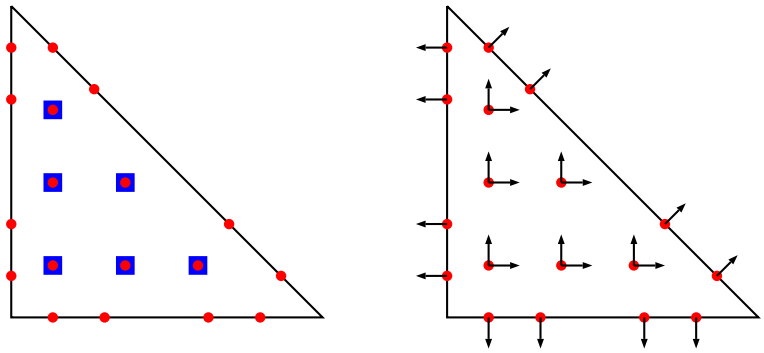


Figure 16. Degrees of freedom on reference triangle for $P = 2$. The solution points are represented by blue squares, flux points are represented by red circles, and unit vectors for Raviart–Thomas flux interpolation are represented by black arrows.

defined solution points and a number of additional points on the element boundaries, is defined. By construction the flux points in the element interior and solution points are collocated. In a departure from the standard FR formulation on triangles [8, 54] and the SD-RT scheme [2, 35] which place $P + 1$ points on each edge of the triangular element, the DFR scheme on triangles places $P + 2$ points on each edge of the triangular element, which results in $N_F = N_S + N_{FB}$ where $N_{FB} = 3(P + 2)$, the number of flux points on the element boundary. A depiction of the solution and flux point locations on a reference triangle can be found in fig. 16.

Further, we have also devised a novel means of handling the auxiliary equation that arises when discretizing advection-diffusion problems. Whereas the standard FR formulation on triangles computes the gradients of the corrected the solution in *transformed* space, our DFR formulation computes these gradients directly in *physical* space. This approach elegantly sidesteps many of the issues that have prevented related methods, such as the aforementioned SD-RT scheme, from being extended to advection-diffusion problems.

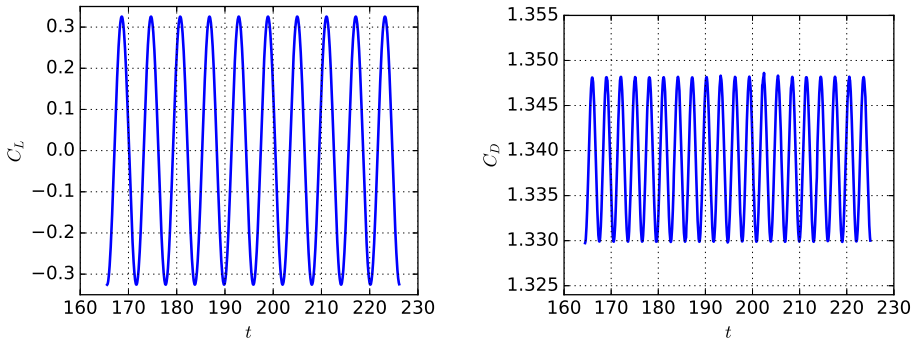
To assess the performance of the DFR scheme in simulating unsteady viscous flow problems of engineering interest, we consider the solution to the unsteady Navier–Stokes equations over a circular cylinder of infinite length at a Reynolds number $R_e = 100$ at a fixed constant viscosity μ. At low Reynolds numbers, this problem can be modeled within a two-dimensional domain, perpendicular to the cylinder axis. This problem is simulated within a square domain $[-30, 70] \times [-50, 50]$ with a circular cylinder of diameter $D = 1$, centered at coordinate $(0, 0)$. The domain is partitioned into 4,030 triangular elements with quadratic edges used to represent the cylinder wall. The outer boundaries of the domain are treated using Riemann-invariant characteristic boundary conditions [27] and an adiabatic, no-slip boundary condition is applied at the cylinder wall boundary. To minimize compressibility effects, the freestream velocity was set consistent with a Mach number $M = 0.1$.

For each case, the flow is marched forward in time using the low-storage RK45[2R+] time integration scheme of Kennedy et. al. [28] until a periodic laminar vortex shedding pattern is fully developed. At this point, the average and peak lift coefficients, C_L, and average and peak drag coefficients, C_D, are computed over 10 shedding cycles, along with the Strouhal number, St. Plots of the time history of lift and drag coefficient over this period can be seen in fig. 17 with an associated contour plot of vorticity in fig. 18.

A comparison of the results achieved using $P = 4$ with those reported by others in the literature can be seen in table 4. In comparing the results across polynomial order and point configuration, the lift, drag an Strouhal numbers are equivalent, indicating that the computational grid is adequately refined for this problem. Comparison of the computed values with the reported results of from

Table 4. Cylinder results and comparison at $Re = 100$

Study	Method	C_L	C_D	St
Current	DFR ($P = 4$)	± 0.326	1.339 ± 0.009	0.165
Cox et. al [14]	Incompressible FR ($P = 3$)	± 0.325	1.339 ± 0.009	0.164
Chan et. al [10]	Spectral Difference ($P = 3$)	± 0.325	1.338 ± 0.009	0.164
Park et. al [42]	Fractional Step	± 0.332	1.33 ± 0.009	0.165
Sharman et. al [46]	SIMPLE	± 0.325	1.33 ± 0.009	0.164



(a) Lift

(b) Drag

Figure 17. Time history of lift and drag coefficients for cylinder at $Re = 100$, $P = 4$, Williams-Shunn points

several other studies show excellent agreement. This result provides support for the efficacy of the DFR scheme on triangles for simulating unsteady viscous flow phenomenon.

3.9 Implicit time stepping via multi-colored parallel Gauss-Seidel method

The push towards high-order, unsteady flow simulations over complex geometries has sparked a need for faster convergence for large scale problems. Accelerated explicit methods and the polynomial multigrid method have been used to accelerate convergence rates but are sometimes not enough to overcome the stiffness found

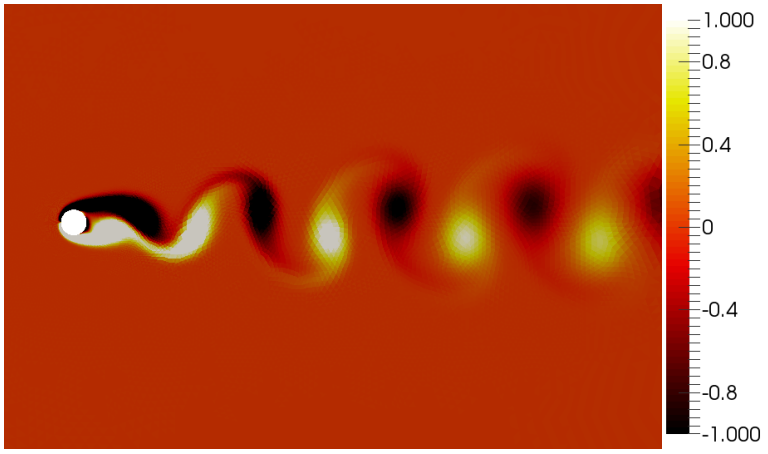


Figure 18. Vorticity contours of cylinder at $Re = 100$, range scaled to $[-1, 1]$ for emphasis

in aerodynamic applications where the cell volume varies by several orders of magnitude between the body and the far field [9, 32, 58]. For these class of problems, implicit methods offer an alternate means to converge steady state solutions or drive the solution to physical time steps in dual time stepping methods by means of larger pseudo time steps. In particular, lower-upper symmetric Gauss-Seidel (LU-SGS) has shown promising results in unstructured compressible flow solvers utilizing finite volume methods, SD methods, the CPR method and more recently, the compact high-order method [13, 25, 32, 49, 63].

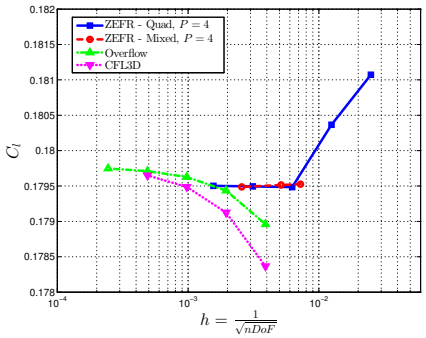
Graphical Processing Units (GPUs) are also becoming more popular among those in the scientific computing community and can demonstrate a substantial performance gain for programs using high-order methods [9, 25, 29, 33, 58]. The DFR method is well suited for GPUs because the vast majority of operations are element local and the increase in amount of work per degree of freedom couples well with the high computational potential of GPUs. Castonguay et al. has shown

the potential of these accelerators to produce results for unsteady simulations using explicit time stepping [9]. Typically, implicit time stepping has been a more difficult problem to address because of the increase in memory requirements and the serial aspects of the algorithm but there have been advances which show that there are methods of overcoming these problems on a single GPU [25].

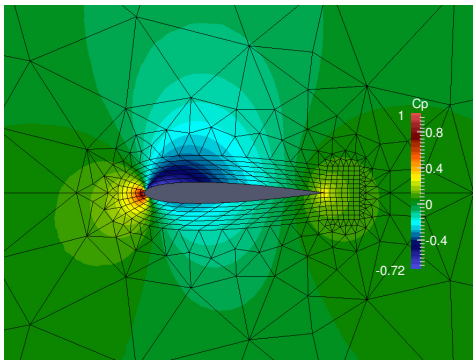
Recent efforts have led to the development and implementation of a multi-GPU, implicit, high-order compressible flow solver for unstructured grids. The solver utilizes the direct Flux Reconstruction (DFR) method and a multi-coloured Gauss-Seidel (MCGS) method to converge the steady state Euler equations in a multi-GPU environment. The MCGS scheme is able to obtain a fast, grid converged lift coefficient of 0.1795 for the NACA 0012 airfoil at a 1.25 degree angle of attack, Mach 0.5. Figure 19 shows that results are obtained with fewer degrees of freedom when compared to Overflow and CFL3D. We also note the substantially improved rate of convergence when compared with explicit RK4 time stepping. The high arithmetic intensity and the ease of parallelization makes MCGS an ideal choice for multiple GPUs. The memory size of the left-hand side matrices in the implicit method limits the scheme's use for high polynomial orders on a single GPU but this bottleneck in memory usage can be mitigated by using multiple GPUs. The scheme is able to maintain near perfect weak scaling showing that it can be effectively distributed over multiple GPUs to solve large problems without a significant degradation in performance.

3.10 DNS of the T161 low pressure turbine cascade

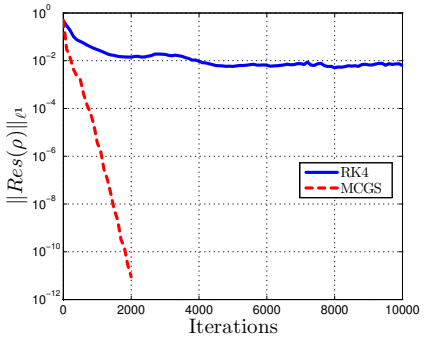
During the course of the award we also conducted scale resolving simulations of MTU T161 low pressure turbine (LPT) cascade. These simulations, performed in



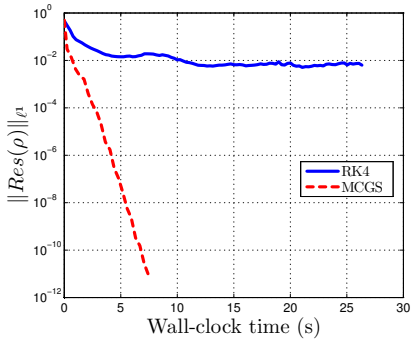
(a) Lift coefficient vs. $\frac{1}{\sqrt{nDoF}}$.



(b) Mesh and pressure contours.



(c) Density residual vs. iteration count.



(d) Density residual vs. wall clock time.

Figure 19. Inviscid flow over the NACA 0012 airfoil, implicit MCGS.

collaboration with MTU Aero Engines and the Vincent Lab at Imperial College London, were undertaken on the Titan supercomputer as part of a DOE INCITE award. The T161 is a highly loaded configuration and is representative of general trends in turbine design. Moreover, it is also particularly challenging as it involves complex unsteady three-dimensional flow behaviour including a large separation bubble on the suction side of each blade and end-wall boundary layer effects. Our specific focus with these simulations is to use Direct Numerical Simulation (DNS) to resolve the flow over a full-scale T161 configuration at $Re = 200\,000$ and $M = 0.6$. Such a simulation provides unprecedented fidelity into the flow physics associated with modern LPT cascades.

The mesh for the simulation is shown in fig. 20 and consists of 90,760,192 second order curved hexahederal elements. With fourth order solution polynomials this corresponds to $11.3 \times 5 = 56.5$ billion degrees of freedom. For computational reasons laminar inflow conditions were applied. Time average statistics for 200 quantities were collected over a period of 11 blade-passes. The computed distribution of $\Delta y+$ along the suction side of the blade can be seen in Figure 21, where Δy^+ corresponds to the non-dimensional wall distance. Within the context of FR this is taken to be the distance from the boundary to the first solution point. Looking at the plot we observe that at all locations $\Delta y^+ < 1$ thus indicating DNS levels of resolution in the boundary layer.

To validate the accuracy of our numerical simulations we compare our results against the experimental data of MTU Aero Engines. The distribution of isentropic Mach M_I number over the blade at the mid-span location is shown in Figure 22. This demonstrates that our DNS results achieve excellent agreement with the experimental data at all measurement locations on both the pressure and suction

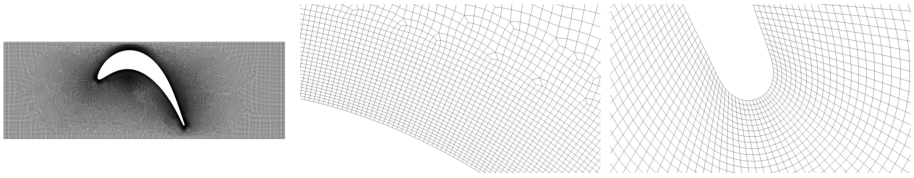


Figure 20. Two dimensional cross-section of the hexahedral mesh used for the T161 simulations.

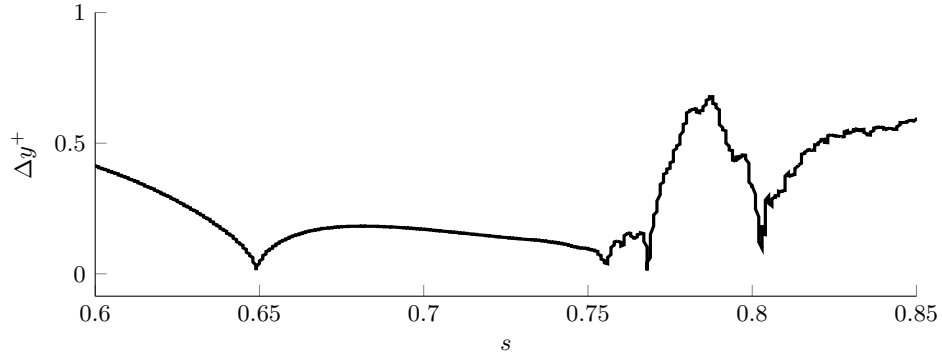


Figure 21. Distribution of Δy^+ along the suction side of the blade.

sides of the blade. Furthermore, a comparison between experimental oil streak lines and computational shear line integral convolution (LIC) lines from the DNS results are shown in Figure 23. On the suction side we observe that the DNS results accurately predict the turbulent end-wall effects and their development downstream on the suction side of the blade. Also, the location and curved shape of the mid-span separation line, which is demarcated by a horizontal edge in the oil streak lines ahead of the mid-chord, is also predicted accurately. Furthermore, the behaviour on the pressure side of the blade is also qualitatively consistent with the experimental data. Instantaneous snapshots of Q criterion isosurfaces, which define vortical structures, over the suction- side of the blade, including a zoom of the end wall region can be seen in Figure 24. It can bee seen that turbulent transition in the mid-span

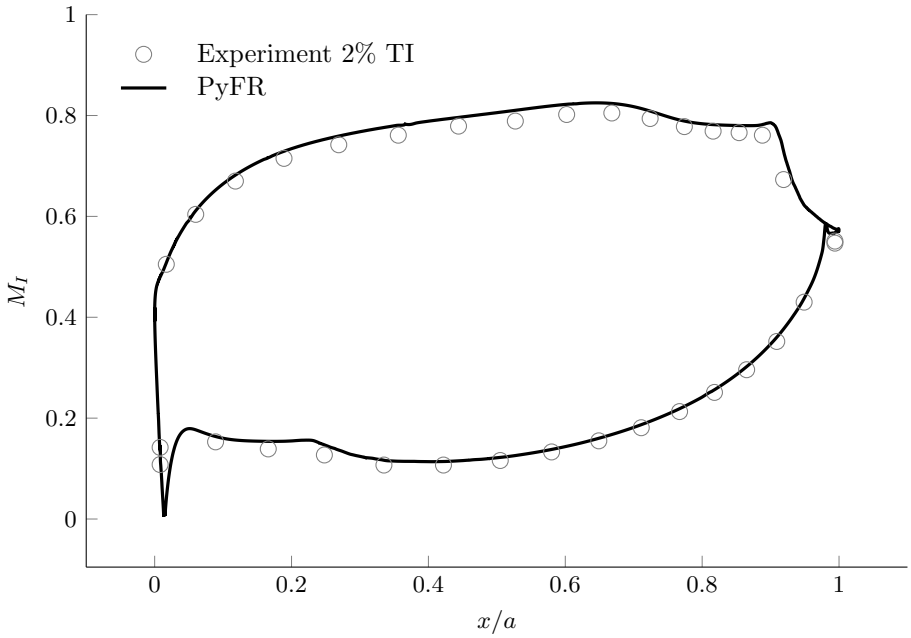


Figure 22. Distribution of M_I over the pressure and suction sides of the LPT blade surface at mid-span.

region occurs near the aft portion of the blade. However, for the end-wall region transition occurs near the leading edge of the blade and propagates inwards towards the mid-span via end-wall effects. This demonstrates that turbulent transition near the ends of the blade is dominated by end-wall effects, whereas turbulent transition in the mid-span region is relatively uniform across the span.

3.11 Data-driven modeling and flow control

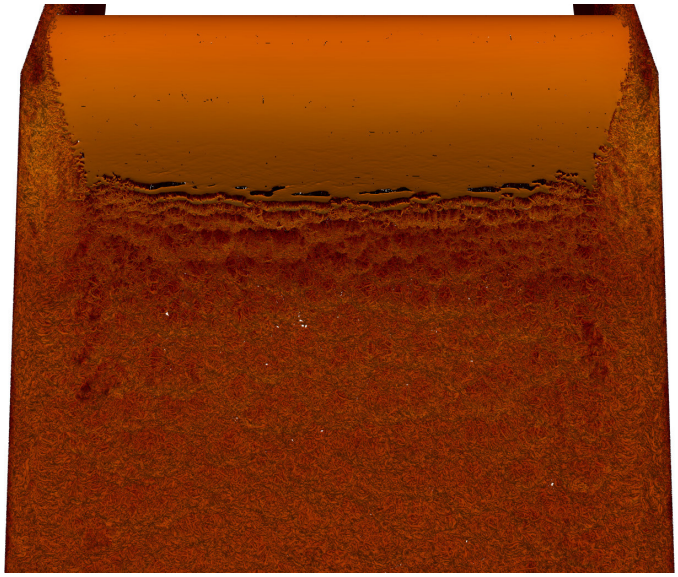
The design of flow control systems remains a challenge due to the nonlinear nature of the equations that govern fluid flow. However, recent advances in CFD have enabled the simulation of complex fluid flows with high accuracy, opening the possibility of using learning-based approaches to facilitate controller design. To

(a) PyFR (b) Experiment 2% TI

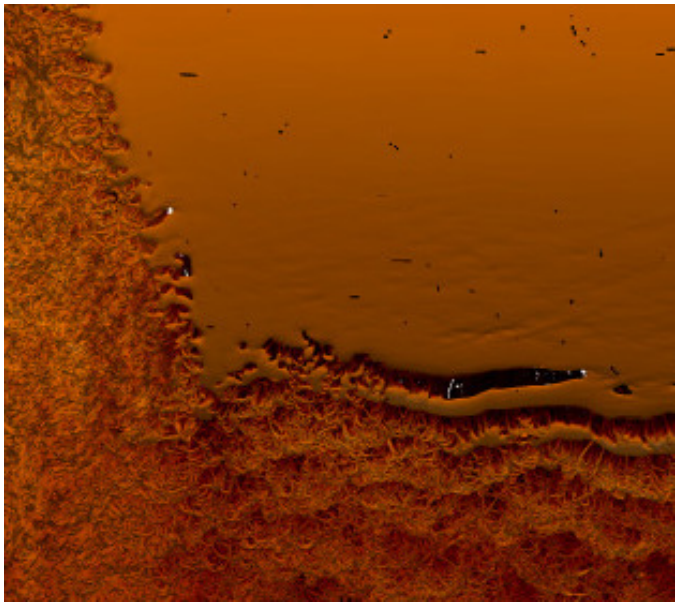
Figure 23. Image of computational shear LIC lines (a) and experimental oil-streak lines (b) on the pressure and suction sides of the LPT blade surface.

demonstrate the possibility of learning the forced dynamics of fluid flow directly from data, we consider the test case of two-dimensional airflow over a rotating cylinder at a Reynolds number of 50. Using a training procedure grounded in Koopman theory, we have shown that it is possible to train neural network models to accurately model the forced and unforced dynamics of the airflow over the cylinder for significant time horizons [37].

Subsequently, we were able to incorporate the trained models into a model predictive control (MPC) framework to suppress vortex shedding over the cylinder. Figure 25 shows the x-momentum of the airflow over time, and demonstrates the effectiveness of the MPC algorithm in suppressing vortex shedding. Furthermore, by examining the selected control inputs, we discovered that the controller obtained through MPC is functionally similar to a proportional controller performing feedback based on y-velocity measurements at a point in the wake of the cylinder. This aligns with previous experimental studies that have shown that vortex suppression

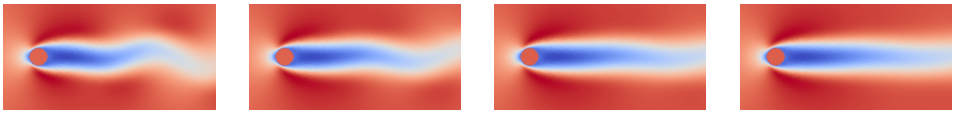


(a) All



(b) Zoom

Figure 24. Instantaneous isosurfaces of Q criterion, colored by velocity magnitude, on the suction side of the blade.



(a) $t = 0$ (b) $t = 100$ (c) $t = 400$ (d) $t = 700$

Figure 25. Snapshots of x-momentum over time as the MPC algorithm attempts to suppress vortex shedding.

can be achieved based on proportional control schemes [5, 23, 41, 45]. Thus, we were able to identify an effective, straightforward, and interpretable control law for vortex suppression, and feel that such data-driven modeling and control approaches may be instrumental in identifying flow control laws for other systems.

Future work will focus on incorporating uncertainty into the learned dynamical models, which may enable the control of increasingly complex fluid flows.

References

[1] K. Aoki, K. Muto, and H. Okanaga. Aerodynamic characteristics and flow pattern of a golf ball with rotation. *Procedia Engineering*, 2010.

[2] A. Balan, G. May, and J. Schöberl. A stable high-order spectral difference method for hyperbolic conservation laws on triangular elements. *Journal of Computational Physics*, 231(5):2359–2375, 2012.

[3] P. W. Bearman and J. K. Harvey. Golf ball aerodynamics. *The Aeronautical Quarterly*, 27:112–122, 1976.

[4] N. Beratlis, K. Squires, and E. Belaras. Numerical investigation of Magnus effect on dimpled spheres. *Journal of Turbulence*, 13(15):1–15, 2012.

[5] E. Berger. Suppression of vortex shedding and turbulence behind oscillating cylinders. *The Physics of Fluids*, 10(9):S191–S193, 1967.

[6] M. J. Brazell, J. Sitaraman, and D. Mavriplis. An overset mesh approach for 3d mixed element high order discretizations. *53rd AIAA Aerospace Sciences Meeting*, 2015.

[7] P. Castonguay, P. E. Vincent, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes for triangular elements. *Journal of Scientific Computing*, 2011.

[8] P. Castonguay, P. E. Vincent, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes for triangular elements. *Journal of Scientific Computing*, 51(1):224–256, 2012.

[9] P. Castonguay, D. M. Williams, P. E. Vincent, M. Lopez, and A. Jameson. On the development of a high-order, multi-GPU enabled, compressible viscous flow solver for mixed unstructured grids. *AIAA paper*, 3229:2011, 2011.

[10] A. S. Chan, P. A. Dewey, A. Jameson, C. Liang, and A. J. Smits. Vortex suppression and drag reduction in the wake of counter-rotating cylinders. *Journal of Fluid Mechanics*, 679:343–382, 2011.

[11] J. Choi, W.-P. Jeon, and H. Choi. Mechanism of drag reduction by dimples on a sphere. *Physics of Fluids*, 18(041702), 2006.

[12] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal of Numerical Analysis*, 35(6):2440–2463, 1998.

[13] C. Cox, C. Liang, and M. W. Plesniak. A high-order method for solving unsteady incompressible Navier–Stokes Equations with implicit time stepping on unstructured grids. In *53rd AIAA Aerospace Sciences Meeting*, 2015.

[14] C. Cox, C. Liang, and M. W. Plesniak. A high-order solver for unsteady incompressible Navier–Stokes equations using the flux reconstruction method on unstructured grids with implicit dual time stepping. *Journal of Computational Physics*, 314:414–435, 2016.

[15] J. A. Crabill. *Towards Industry-Ready High-Order Overset Methods on Modern Hardware*. PhD thesis, Stanford University, 2018.

[16] J. A. Crabill, A. Jameson, and J. Sitaraman. A high-order overset method on moving and deforming grids. *AIAA Modeling and Simulation Technologies Conference*, (AIAA2016-3225), 2016.

[17] J. A. Crabill, F. D. Witherden, and A. Jameson. A parallel direct cut algorithm for high-order overset methods with application to a spinning golf ball. *Journal of Computational Physics*, 2018.

[18] P. R. Eiseman, S. J. Ebenezer, V. Sudharshanam, and V. Anbumani. GridPro manuals, 2017. Downloadable from www.gridpro.com.

[19] M. C. Galbraith. *A Discontinuous Galerkin Overset Solver*. PhD thesis, University of Cincinatti, 2013.

[20] A. Gelb and D. Cates. Detection of edges in spectral data III-refinement of the concentration method. *Journal of Computational Physics*, 36(1):1–43, 2008.

[21] A. Gelb and E. Tadmor. Enhanced spectral viscosity approximations for conservation laws. *Appl. Numer. Math.*, 33(1):3–21, 2000.

[22] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 2009. (Mesh-Generation Software).

[23] M. D. Gunzburger and H. C. Lee. Feedback control of karman vortex shedding. *Journal of Applied Mechanics*, 63(3):828–835, 1996.

[24] F. Hindenlang, T. Bolemann, and C.-D. Munz. Mesh curving techniques for high order discontinuous Galerkin simulations. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pages 133–152. Springer, 2015.

[25] M. Hoffmann, C.-D. Munz, and Z. Wang. Efficient implementation of the cpr formulation for the Navier–Stokes equations on GPUs. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

[26] H. C. Howdhury, B. Loganathan, Y. Wang, I. Mustary, and F. Alam. A study of dimple characteristics on golf ball drag. *Procedia Engineering*, 2016.

[27] A. Jameson and T. Baker. Solution of the Euler equations for complex configurations. In *6th Computational Fluid Dynamics Conference Danvers*, page 1929, 1983.

[28] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis. Low-storage, explicit

Runge–Kutta schemes for the compressible Navier–Stokes equations. *Applied numerical mathematics*, 35(3):177–219, 2000.

[29] A. Klöckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, 2009.

[30] J. Li, M. Tsubokura, and M. Tsunoda. Numerical investigation of the flow around a golf ball at around the critical Reynolds number and its comparison with a smooth sphere. *Flow Turbulence and Combustion*, 95:415–436, 2015.

[31] J. Li, M. Tsubokura, and M. Tsunoda. Numerical investigation of the flow past a rotating golf ball and its comparison with a rotating smooth sphere. *Flow Turbulence and Combustion*, 99:837–864, 2017.

[32] C. Liang, R. Kannan, and Z. Wang. A p-multigrid spectral difference method with explicit and implicit smoothers on unstructured triangular grids. *Computers & fluids*, 38(2):254–265, 2009.

[33] M. López-Morales, J. Bull, J. A. Crabill, T. D. Economon, D. Manosalvas, J. Romero, A. Sheshadri, J. Watkins, D. Williams, F. 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.

[34] Y. Lv and M. Ihme. Entropy-bounded discontinuous Galerkin scheme for Euler equations. *Journal of Computational Physics*, 295:715–739, 2015.

[35] G. May and J. Schöberl. Analysis of a spectral difference scheme with flux

interpolation on raviart-thomas elements. *Aachen Institute for Advanced Study in Computational Engineering Science*, 2010.

[36] R. D. Mehta. Aerodynamics of sports balls. *Annual Review of Fluid Mechanics*, 17:151–189, 1985.

[37] J. Morton, F. D. Witherden, A. Jameson, and M. J. Kochenderfer. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[38] M. Muto, M. Tsubokura, and N. Oshima. Negative Magnus lift on a rotating sphere at around the critical Reynolds number. *Physics of Fluids*, 24(014102), 2012.

[39] C. Nastase, D. Mavriplis, and J. Sitaraman. An overset unstructured mesh discontinuous galerkin approach for aerodynamic problems. *49th AIAA Aerospace Sciences Meeting*, 2011.

[40] R. W. Noack. A direct cut approach for overset hole cutting. *18th AIAA Computational Fluid Dynamics Conference*, 2007.

[41] D. S. Park, D. M. Ladd, and E. W. Hendricks. Feedback control of von Kármán vortex shedding behind a circular cylinder at low reynolds numbers. *Physics of Fluids*, 6(7):2390–2405, 1994.

[42] J. Park, K. Kwon, and H. Choi. Numerical solutions of flow past a circular cylinder at reynolds numbers up to 160. *KSME international Journal*, 12(6):1200–1205, 1998.

[43] J. S. Park, F. D. Witherden, and P. E. Vincent. High-order implicit large-eddy simulations of flow over a NACA0021 aerofoil. *AIAA Journal*, 55(7):2186–2197, 2017.

[44] J. D. Romero. *On the development of the direct flux reconstruction scheme for high-order fluid flow simulations*. PhD thesis, Stanford University, 2017.

[45] K. Roussopoulos. Feedback control of vortex shedding at low Reynolds numbers. *Journal of Fluid Mechanics*, 248:267–296, 1993.

[46] B. Sharman, F.-S. Lien, L. Davidson, and C. Norberg. Numerical predictions of low reynolds number flows over two tandem circular cylinders. *International Journal for Numerical Methods in Fluids*, 47(5):423–447, 2005.

[47] J. Sitaraman. Tioga: Topology independent overset grid assembly library. https://github.com/jsitaraman/tioga, 2015. (Overset Connectivity Library for CFD).

[48] C. E. Smith, N. Beratlis, E. Balaras, K. Squires, and M. Tsunoda. Numerical investigation of the flow over a golf ball in the subcritical and supercritical regimes. *International Journal of Heat and Fluid Flow*, 31:262–273, 2010.

[49] Y. Sun, Z. Wang, and Y. Liu. Efficient implicit non-linear lu-sgs approach for compressible flow computation using high-order spectral difference method. *Journal of Computational Physics*, 5(2-4):760–778, 2009.

[50] L. L. Ting. Application of CFD technology analyzing the three-dimensional aerodynamic behavior of dimpled golf balls. *ASME International Mechanical Engineering Congress & Exposition*, 2002.

[51] L. L. Ting. Effects of dimple size and depth on golf ball aerodynamics. *4th ASME/JSME Joint Fluids Engineering Conference*, 2003.

[52] B. C. Vermeire, F. D. Witherden, and P. E. Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 2017.

[53] P. E. Vincent, P. Castonguay, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes. *Journal of Scientific Computing*, 47(1):50–72, 2011.

[54] D. M. Williams, P. Castonguay, P. E. Vincent, and A. Jameson. Energy stable flux reconstruction schemes for advection–diffusion problems on triangles. *Journal of Computational Physics*, 250:53–76, 2013.

[55] D. M. Williams, P. Castonguay, P. E. Vincent, and A. Jameson. Energy stable flux reconstruction schemes for advection-diffusion problems on triangles. *Journal of Computational Physics*, 2013.

[56] A. Wissink. Helios solver developments including strand meshes. Oral presentation, 11th Symposium on Overset Composite Grids and Solution Technology, 2012.

[57] A. Wissink. An overset dual-mesh solver for computational fluid dynamics. *7th International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

[58] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An open source framework for solving advection–diffusion type problems on streaming archi-

tectures using the flux reconstruction approach. *Computer Physics Communications*, 185(11):3028–3040, 2014.

[59] F. D. Witherden, A. M. Farrington, and P. E. Vincent. PyFR: An open source framework for solving advection-diffusion type problems on streaming architectures using the flux reconstruction approach. *Computer Physics Communications*, 185:3028–3040, 2014.

[60] F. D. Witherden, B. C. Vermeire, and P. E. Vincent. Heterogeneous computing on mixed unstructured grids with PyFR. *Computers and Fluids*, 120:173–186, 2015.

[61] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, 1984.

[62] B. D. Wozniak, F. D. Witherden, F. P. Russell, P. E. Vincent, and P. H. Kelly. GiMMiK: Generating bespoke matrix multiplication kernels for accelerators: Application to high-order computational fluid dynamics. *Computer Physics Communications*, 202:12–22, 2016.

[63] S. Yoon and A. Jameson. Lower-upper symmetric-gauss-seidel method for the Euler and Navier–Stokes equations. *AIAA journal*, 26(9):1025–1026, 1988.

[64] X. Zhang and C. W. Shu. Positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations with source terms. *Journal of Computational Physics*, 230(4):1238–1248, 2011.

[65] X. Zhang and C.-W. Shu. Numerische Mathematik A minimum entropy

principle of high order schemes for gas dynamics equations. *Numer. Math*, 121:545–563, 2012.

[66] X. Zhang and C. W. Shu. Positivity-preserving high order finite difference WENO schemes for compressible Euler equations. *Journal of Computational Physics*, 231(5):2245–2258, 2012.