Large-Scale Simulations of Complex Turbulent Flows:

Modulation of Turbulent Boundary Layer Separation and Optimization of Discontinuous Galerkin Methods for Next-Generation HPC Platforms

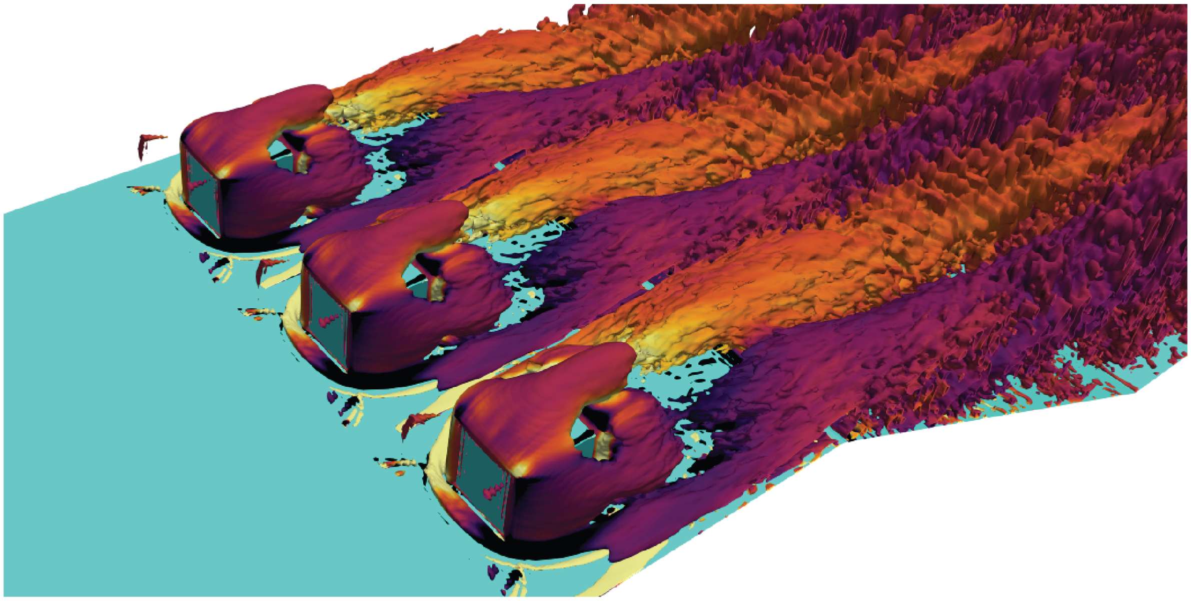
by

Suyash Tandon

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering and Scientific Computing)
in The University of Michigan
2020

Doctoral Committee:

Associate Professor Eric Johnsen, Chair
Assistant Professor Jesse Capecelatro
Associate Professor Krzysztof Fidkowski
Computational Scientist, Marc T. Henry de Frahan
Associate Professor Kevin J. Maki



SCFP Laboratory

SUYASH TANDON

Suyash Tandon
suyashtn@umich.edu
ORCID iD: 0000-0001-5025-0284

To my beloved family...*Mom, Dad, Sids*

and

The Tandons & the Khots.

ACKNOWLEDGEMENTS

I would like to begin by expressing my immense gratitude and love towards my family. My parents and my brother are my biggest source of inspiration, who have been supportive, have encouraged me to push my boundaries, and motivated me to strive for excellence. For without them, I would not be half the person I am today, and for that I dedicate this thesis to them. Of course no thesis is complete without an advisor, who has the most difficult job of advising a total stranger, adapt, learn and craft new research directions, guide the thesis development and above all do the painstaking job of reading the thesis in its entirety. To this, I'm grateful to Dr. Eric Johnsen, my advisor, for his guidance, mentorship and support, which has helped me excel in my research endeavor, and enabled me to perform at the best of my ability. I take this opportunity to thank Dr. Kevin J. Maki for his valuable inputs that have played a vital role in shaping my thesis objectives. I extend my thanks to Dr. Krzysztof Fidkowski, Dr. Jesse Capecelatro and Dr. Marc T. Henry de Frahan, for serving the role of my committee members, and for their thoughts and advice on my work.

My experience at The University of Michigan have been invaluable and have helped me chisel and polish my personality as a researcher. I would like to thank my dear friend and former colleague, Dr. Siddhesh Shinde.

Our research progressed in tandem during the early stages of my thesis inception and he has been a great mentor and support, both in and outside of the lab. I thank all my friends including Vickey for his wit, humor and delicious cooking skills; Prasad for his life-philosophies and coffee rounds at Mujos; Nikhil for his exceptional Chai sessions; Nitish for bearing me as his flat-mate, his anecdotes and PS3 gaming sessions; Pratik for his funny debates; Priyank for his crazy party and travel ideas; Jovin for his passion for coffee and sharing salsa dance enthusiasm; Sameer for Game of Thrones sessions; Summit for his special bread roll recipe and social media skills; and Namit for his cricket enthusiasm and adding Bollywood 'mirch masala' (spice) to life. It is an honor to have founded the "House-of-Bhaus" with some of them. In addition my time at The University of Michigan would not have been nearly as fulfilling without the company of my friends in the Autolab. I thank Brian for his house parties and science talks; Tyler (T.J.) for his board problems, humor, and thoughts on colormaps; Brandon for fixing my Linux machine after I found new ways of breaking it; Marc for sharing knowledge on parallel programming, LATEXtutorials and presentation skills; Mauro for proof-reading my documents and guidance on approaching research tasks; Shahab for his blender animations and guidance on scientific topics; Sam for sharing his new French press, hiking in Los Alamos, his insights into French culture; Harish for his philosophy about life; Phil for sharing his DG wisdom; Kevin for his March madness and happy hour enthusiasm; Griffin for managing the lab social media accounts; Kazuya for relieving me and taking the responsibility of managing lab hardware; Minki for managing the computational allocations; Eunhye for working and supporting Turbulence related research; and Greg and Yang for all the OpenFOAM related help. Their presence made the Autolab a fun place to work and my graduate experience a pleasant journey. During my Ph.D. I served on the board of Mechanical Engineering Graduate Council (MEGC), the Scientific Computing Student Club (SC2), Engineering Graduate Symposium (EGS). I want to thank everyone at MEGC, SC2 and EGS for giving me the opportunity to contribute my bit to improving the graduate life experience of students on campus. I also thank people outside my lab who made my time in Ann Arbor pleasant including, Josie, Preeti, Varshini, Laura, Hannah, Nolan, Aunnasha and Swaraj. Finally, I want to thank my undergrad friends from Fr. Conceicao Rodrigues College of Engineering (Fr. CRCE), Mumbai, and my childhood friends from Delhi, Bhopal, Mumbai: Bhishman, Bhumika, Pranav, Jay, Sreejit, Prashanth, Neeraj, Hina, Ankit, Akshat, Eesha, Prakhar Swedha, and Utkarsh, for all the adventures, love and craziness.

I am grateful to Curadev Pvt. Ltd., for awarding me the Graduate Fellowship towards my graduate studies at the University of Michigan between 2016-2019 and the Michigan Institute for Computational Discovery and Engineering (MICDE) for funding my final year (2019-2020) through the MICDE Catalyst grant and the MICDE Fellowship Award. Additionally, this research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, and the Extreme Science and Engineering Discovery Environment's (XSEDE) Stampede2 system at the Texas Advanced Computing Center (TACC), which is supported by the National Science Foundation grant number ASC-160041.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

The separation of spatially evolving turbulent boundary layer flow near regions of adverse pressure gradients has been the subject of numerous studies in the context of flow control. Although many studies have demonstrated the efficacy of passive flow control devices, such as vortex generators (VGs), in reducing the size of the separated region, the interactions between the salient flow structures produced by the VG and those of the separated flow are not fully understood. Here, wall-resolved large-eddy simulation of a model problem of flow over a backward-facing ramp is studied with a submerged, wall-mounted cube being used as a canonical VG. In particular, the turbulent transport that results in the modulation of the separated flow over the ramp is investigated by varying the size, location of the VG, and the spanwise spacing between multiple VGs, which in turn are expected to modify the interactions between the VG-induced flow structures and those of the separated region. The horseshoe vortices produced by the cube entrain the freestream turbulent flow towards the plane of symmetry. These localized regions of high vorticity correspond to turbulent kinetic energy production regions, which effectively transfer energy from the freestream to the near-wall regions. Numerical simulations indicate that: (i) the gradients and the fluctuations, scale with the size of the cube and thus lead to more effective modulation for large cubes, (ii) for a given cube height the different upstream cube positions affect the behavior of the horseshoe vortex—when placed too close to the leading edge, the horseshoe vortex is not sufficiently strong to affect the large-scale structures of the separated region, and when placed too far, the dispersed core of the streamwise vortex is unable to modulate the flow over the ramp, (iii) if the spanwise spacing between neighboring VGs is too small, the counter-rotating vortices are not sufficiently strong to affect the large-scale structures of the separated region, and if the spacing is too large, the flow modulation is similar to that of an isolated VG. Turbulent boundary layer flows are inherently multiscale, and numerical simulations of such systems often require high spatial and temporal resolution to capture the unsteady flow dynamics accurately. While the innovations in computer hardware and distributed computing have enabled advances in the modeling of such large-scale systems, computations of many practical problems of interest are infeasible, even on the largest supercomputers. The need for high accuracy and the evolving heterogeneous architecture of the next-generation high-performance computing centers has impelled interest in the development of high-order methods. While the new class of recovery-assisted discontinuous Galerkin (RADG) methods can provide arbitrary high-orders of accuracy, the large number of degrees of freedom increases costs associated with the arithmetic operations performed and the amount of data transferred on-node. The purpose of the second part of this thesis is to explore optimization strategies to improve the parallel efficiency of RADG. A cache data-tiling strategy is investigated for polynomial orders 1 through 6, which enhances the arithmetic intensity of RADG to make better utilization of on-node floating-point capability. In addition, a power-aware compute framework is suggested by analyzing the power-performance trade-offs when changing from double to single-precision floating-point types—energy savings of 5 W per node are observed—which suggests that a transprecision framework will likely offer better power-performance balance on modern HPC platforms.

CHAPTER 1

Introduction

Many processes in nature are inherently nonlinear. Multiscale problems are an example of such complex systems, where the dynamic coupling between a wide range of length and time scales contributes to the nonlinearity. Multiscale problems are prevalent in areas such as turbulent flows, magnetohydrodynamics, solid mechanics, and quantum mechanics, among other fields. Experiments of such complex systems have helped in advancing our fundamental understanding of the underlying physical phenomena. However, experiments of complex multiscale problems are costly to design, manufacture, and implement. Due to the wide range of spatial and temporal scales that must be investigated, and the multiphysics aspects of the complex multiscale systems, diagnostic tools can only offer limited information about the flow dynamics. The sensitivity of these systems to the initial conditions and material properties make it difficult to attain good experimental reproducibility. Computational physics has enabled researchers to harness the increase in computing power by designing and developing mathematical algorithms to conduct numerical simulations, which circumvent the challenges faced by the experiments. Numerical simulations offer a cost-effective way of exploring the relevant parameter space, isolating the physical effects of interest, and providing a complete description of the system's evolution. While numerical simulations have proven to be promising tools to study complex problems, the large number of degrees of freedom that are required to characterize multiscale systems of interest, such as turbulence, pose an exorbitantly high demand on the computational resources, which cannot be satisfied even on the most powerful supercomputers. This has spawned the need to build larger computational systems. However, to extract the benefits of these large computing systems, new methods have to be designed and developed. This issue has attracted the attention of many researchers and has propelled investigations in areas including model development, computer hardware, application, and system software, and other related areas.

1.1 Turbulence: a multiscale problem

We continuously engage with fluids of various types—gases or liquids, the motion of which can be described by the well-known Navier-Stokes equations (NSE). The strong nonlinearity of the NSE leads to one of the most intriguing features of fluid dynamics: *Turbulence*, where the flow is characterized by chaotic and highly unsteady motion of diverse spatial and temporal scales (Pope, 2000). Turbulence is prevalent in many practical applications, such as atmospheric flow, flow over an aircraft, flow over road vehicles, flow in chemical mixing chambers, and is an ongoing area of research. Understanding the universal nature of turbulent flows has eluded researchers for a long time. Feynman *et al.* (1964) has referred to turbulence as "the most important unsolved problem of classical physics". Kolmogorov (1941) hypothesized turbulence to have a continuous cascade of scales and energy, and provided a foundation connecting the underlying nature of all turbulent flows. This cascade is driven by large energetic flow structures and continues until the turbulent fluctuations are dissipated into heat at the smallest scales. Of particular interest is the problem of separation and control of wall-bounded turbulent boundary layer (TBL) flow, where previous research has shown that the presence of a wall alters the turbulence dynamics (Wu & Moin, 2009). The term wall-bounded refers to the flow configuration where the fluid motion occurs in contact with a solid surface. Wall-bounded TBL flows are common in many internal and external flow systems—atmospheric flow overland, flow in diffusers, and flow over aircraft wings are a few examples. Based on the speed of the flow under consideration, the variations in density may be significant, which is governed by the ratio of fluid velocity (u) to the speed of sound in the medium (c), known as Mach number ($\mathcal{M}$). As a rule of thumb, if $\mathcal{M} \lesssim 0.3$, the flow is considered to be incompressible and compressible in other scenarios. The problem of interest in the present work lies in the incompressible wall-bounded turbulent flow regime.

1.2 Modulation of turbulent boundary layer flow

In the vicinity of a solid surface, the flow forms a thin region where the viscous effects dominate, the shear stress is high, and the magnitude of flow velocity is low compared to the freestream. At the point where the fluid meets the solid surface, a no-slip boundary condition must be met, which mandates that the fluid is at rest relative to the surface. Since the fluid far away continues to flow at the freestream velocity, the difference in the flow behavior in the wall-normal direction results in a steep velocity gradient. The region between the surface and the point at which the flow attains 99% of the freestream velocity is referred to as the boundary layer (Prandtl, 1904) and denoted as δ. Reynolds (1883) established that the boundary layer flow is characterized as laminar or turbulent based on the Reynolds number defined as,

$$Re = \frac{UL}{\nu} \tag{1.1}$$

where U is the flow velocity, L is the characteristic length and ν is the kinematic viscosity of the fluid. For a fluid moving on a flat surface if the $Re > 5 \times 10^5$, the boundary layer transitions from a laminar to a turbulent regime.

Many external flow systems are characterized by open flow with a spatially evolving TBL, where the boundary layer thickness grows in the streamwise direction (Wu & Moin, 2009). Figure 1.1 shows that a TBL consists of an inner region (with inner variables: friction velocity u_τ and viscosity ν), and an outer region (with outer variables: U and δ). The characteristic length scale in the inner region is much smaller than the outer region, and it is densely populated with coherent structures (Reynolds, 1894; Kline *et al.*, 1967). The outer region has larger coherent structures known as the hairpin vortices, which are organized in packets, and are of the size of the boundary layer thickness (Brown & Thomas, 1977; Adrian *et al.*, 2000).

The separation of spatially evolving TBL from the surface occurs when the flow along the

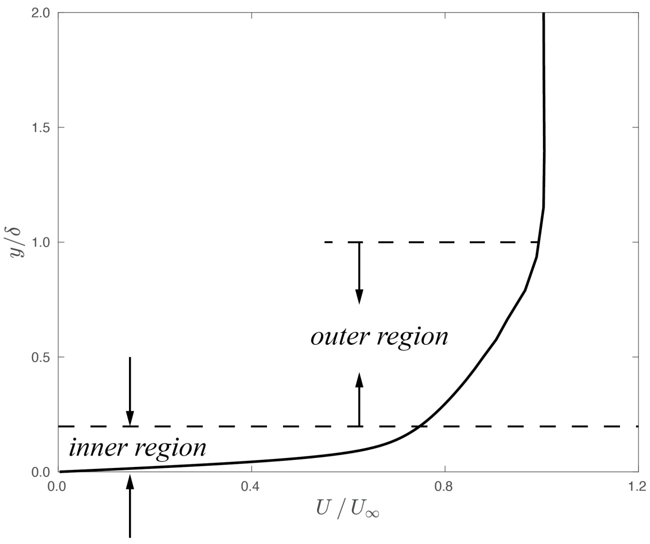


Figure 1.1: Mean turbulent boundary layer velocity profile normalized by the outer variables.

surface decelerates rapidly, either due to strong adverse pressure gradient (APG) or due to change in geometry. Figure 1.2 shows a schematic of a spatially evolving TBL separating from a smooth surface in the presence of an APG. In the limit of high *Re*, consider the streamwise momentum equation inside a TBL

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial y^2} \tag{1.2}$$

At the solid surface, if the flow is considered steady state then $\partial u/\partial t = 0$, and due to no-slip condition $u = v = 0$. This results in,

$$\frac{dp}{dx} = \mu\frac{\partial^2 u}{\partial y^2}$$

On the surface, $\partial^2 u/\partial y^2$ must always have the same sign as the pressure gradient dp/dx, and in case of APG, $dp/dx > 0$. However, at the edge of the boundary layer, $\partial^2 u/\partial y^2 < 0$. Therefore, there must exist an inflection point between the inner and outer regions. The point in the streamwise direction where the velocity gradient becomes so steep that $(\partial u/\partial y)_{y=0} = 0$ is defined as the *point of separation*. The presence of an APG affects the formation of turbulent structures and their corresponding transport inside a spatially evolving TBL (Durbin & Belcher, 1992) and gives rise

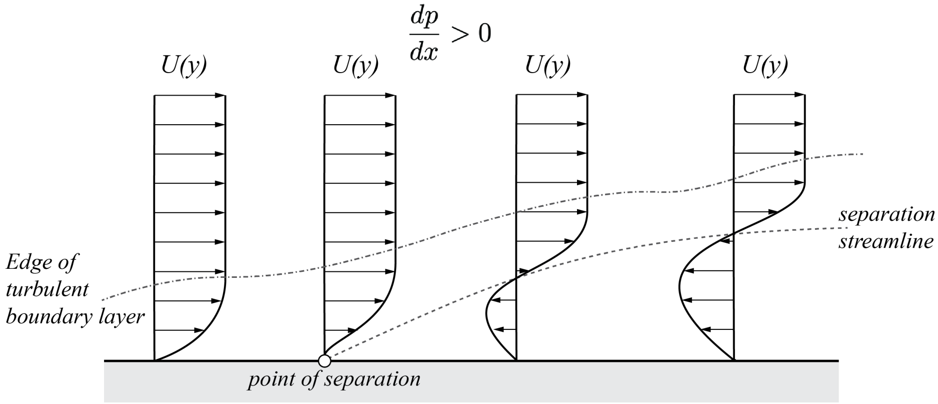


Figure 1.2: A schematic representation of turbulent boundary layer separation on a smooth surface in the presence of adverse pressure gradient.

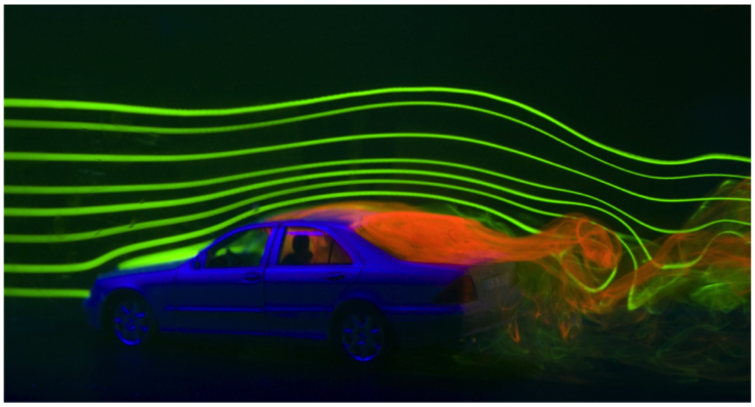


Figure 1.3: Visualization of air-flow separation for flow over a road vehicle. Photo credit:NASA (2010).

to non-equilibrium turbulent energy transport (Aubertine & Eaton, 2005). In the separation bubble, a region of recirculating flow forms, which may give rise to undesirable effects—from reducing lift on aircraft wings to increased drag on vehicles and reducing efficiency in chemical mixing chambers. For external flow over a road vehicle, figure 1.3 shows air-flow separation near the rear end of the vehicle, while figure 1.4 shows the increase in aerodynamic drag, as a result of flow separation at larger speeds.

Control of the separated flow could be exploited to reduce the losses described above. When a turbulent flow encounters a strong APG, it expends its energy to overcome the APG. According to the Bernoulli equation, the kinetic energy gradient $\left(\frac{1}{2}\Delta u^2\right)$ is balanced by the pressure gradient $(\Delta p/\rho)$. Conversely, one might expect that energizing the flow mitigates the separation of flow.

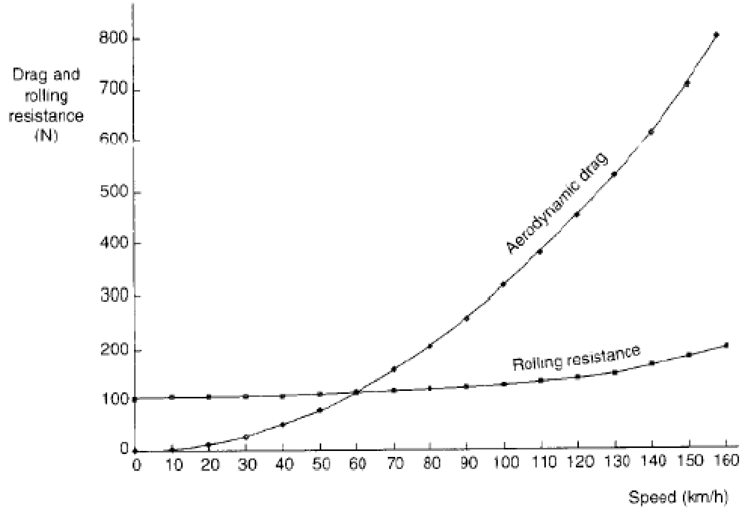


Figure 1.4: Increase in the aerodynamic drag on a road vehicle as a function of its speed. Adapted from Barnard (2001).

Energizing the TBL can be achieved by inducing strong stream-wise and span-wise vortices near the wall, which entrain high momentum fluid from the freestream to the near-wall region. The methods used to facilitate this momentum transfer can be classified into two categories: passive and active flow control. This dissertation work focuses on employing passive control strategies.

1.2.1 Wall-mounted cube as a passive vortex generator

Brown *et al.* (1968), Calarese *et al.* (1985), Englar (2001), Logdberg (2006), Mohan *et al.* (2013), Wilson *et al.* (2019), and Fisher *et al.* (2020), among others, explored passive control techniques such as vortex generators (VGs) to reduce drag and improve performance in a variety of engineering applications. VGs are effective when the point of separation is fixed spatially (Rao & Kariya, 1988). Selby *et al.* (1990) used transverse grooves and observed a reduction of the separation bubble size over a backward-facing, curved ramp when the grooves were located one boundary layer thickness upstream of the ramp. Experimental investigation of VGs with different shapes and configurations by Lin (2002) on a backward-facing, curved ramp demonstrated that the submerged VGs of height 65% below the boundary layer thickness generates streamwise vortices that enhance turbulent mixing, and was substantiated in a parallel study by Jenkins *et al.* (2002). Figure 1.5

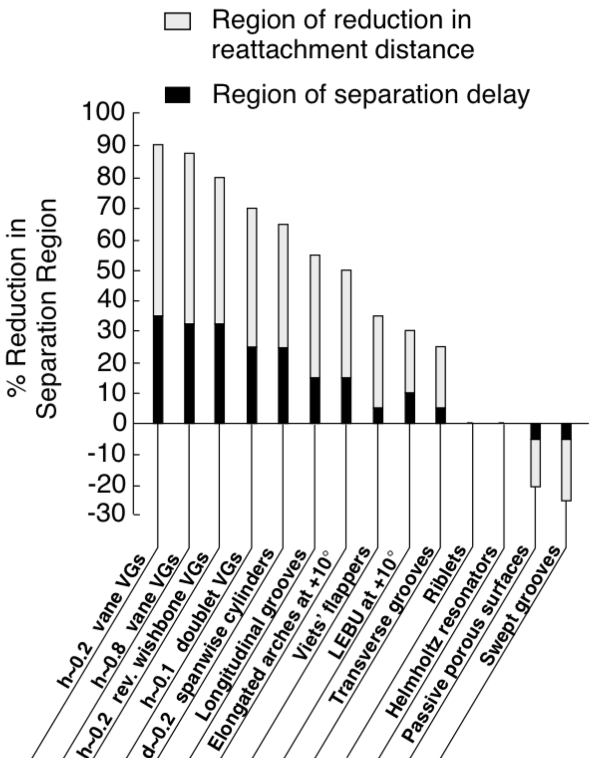


Figure 1.5: Relative effectiveness of flow separation control versus device category. Lin (2002), ©Elsevier. Reproduced with permission. All rights reserved.

shows the relative effectiveness of separation control for different device categories studied by Lin (2002). Flat-plate boundary layer flow over submerged VGs of different shapes has also been investigated in the past (Ashill *et al.*, 2002; Yao *et al.*, 2002; Elbing *et al.*, 2013; Iyer & Mahesh, 2013). However, it is difficult to generalize findings across the different shapes (e.g., Wishbone, Doublet Wheeler (Lin *et al.*, 1991)) as the flow features appear to be problem dependent. For this reason, a cubic VG is considered as a canonical geometry in this study.

The flow structures that form around a wall-mounted cube placed in fully-developed turbulent flow have been well documented in previous studies. Martinuzzi & Tropea (1993) investigated the flow over surface-mounted prismatic obstacles with an aspect ratio (width-to-height ratio W/H) in the range 1 and 24 and concluded that for a fully-developed channel flow with $W/H < 6$ the flow is nominally two dimensional in the middle of the wake. Numerical simulations (Shah & Ferziger, 1997; Krajnovic & Davidson, 1999, 2002; Hwang & Yang, 2004) indicate that the presence of the cube causes the TBL to break down and form a horseshoe vortex system (HVS) upstream of the

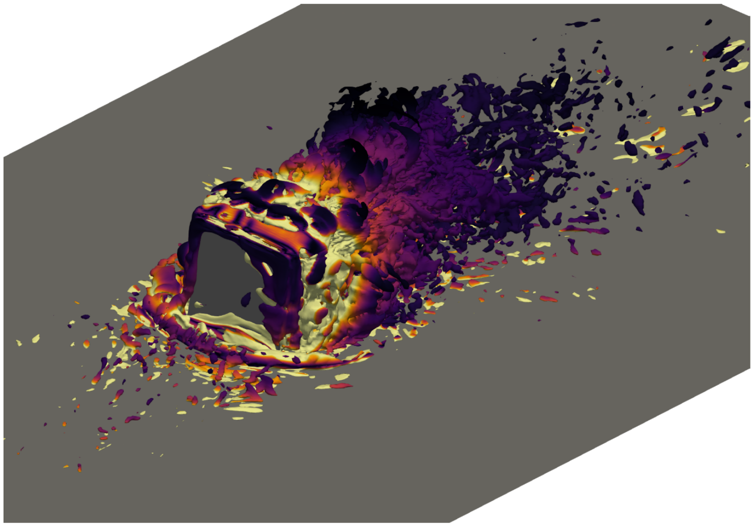


Figure 1.6: Iso-surfaces of Q-criterion colored with vorticity magnitude for flow over a wall-mounted cube in a spatially evolving turbulent boundary layer. Shinde (2018), Reproduced with permission.

cube, with small-scale hairpin and toroidal structures emanating from the top and sides of the cube and a separation bubble in the near wake, which is consistent with the experimental observations of Martinuzzi & Tropea (1993). Castro & Robins (1977) concluded that for Reynolds number over $4,000$ based on the cube height, there are no discernible Reynolds number effects on the flow separation and reattachment in the near-wake of the cube. Shinde *et al.* (2017) showed that for spatially evolving TBL flow over a wall-mounted cube on a flat plate, the flow structures that form around the wall-mounted cube are similar to those observed in the fully-developed channel flow. Shinde *et al.* (2017) further concluded that the strength of the HVS and its relative transport of TKE in the near-wake increases with the cube height relative to the boundary layer thickness. However, the TKE decays within a few cube heights as the streamwise vortices of the cube are weaker than the more complex VGs investigated by Lin (2002). Figure 1.6 shows the vortical structures around a wall-mounted cube in a spatially-evolving TBL.

In a numerical setup, a cubical geometry offers other advantages: (i) a face-on cube has a single length dimension associated with it, as opposed to the large parameter space needed to describe the complex VGs investigated by Lin (2002), and (ii) The cube produces similar large streamwise coherent structures in its wake, the HVS, as discussed before, which are essential flow features

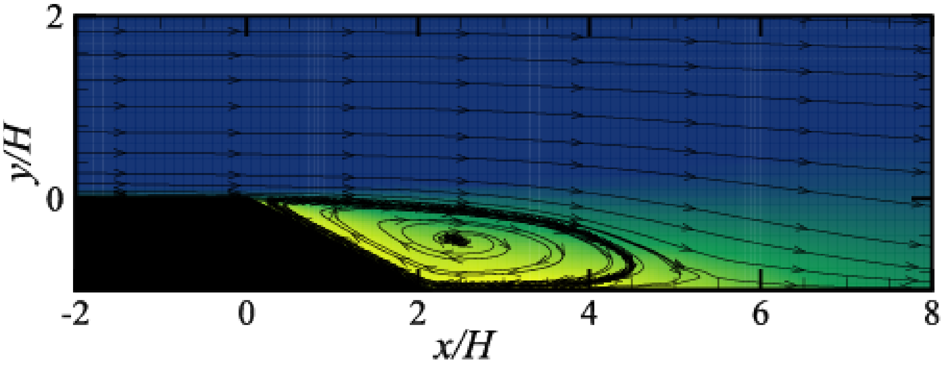


Figure 1.7: Turbulent boundary layer separation over a backward-facing ramp.

associated with VGs. The proposition to use a wall-mounted cube as a passive VG gives rise to three critical questions: (a) how does the modulation of flow separation depend on the cube height relative to the boundary layer thickness? (b) how does the position of the cube relative to the region of APG affect the flow modulation? (c) when using an array of equally-spaced cubes, how does the spacing between neighboring cubes affect the modulation of flow? We seek to answer these questions by studying flow separation and its modulation by wall-mounted cubes in a canonical flow geometry of backward-facing ramp.

1.2.2 Modulation of turbulent flow separation over a backward-facing ramp

Separation of spatially evolving TBL flow over a backward-facing ramp occurs in a variety of internal (diffusers, mixing chambers) and external (airfoils, road vehicles) flow systems. At sufficiently high Reynolds number, pressure changes in the expansion region oppose changes in viscous stresses, and ultimately lead to flow separation, with reattachment beyond the end of the ramp. Figure 1.7 illustrates flow separation over a backward-facing ramp of height H. In the separation bubble, a region of recirculating flow forms, whose size is on the order of the ramp height and the flow near the ramp surface moves in the upstream direction as opposed to freestream flow. Given that they have a fixed point of separation, backward-facing steps or ramps (Armaly *et al.*, 1983; Adams, 1984; Westphal *et al.*, 1984; Kourta *et al.*, 2015) are commonly considered to study flow separation.

The streamwise location where the separated shear layer reattaches to the bottom surface is called the reattachment point. Although the separation point is fixed, the reattachment location depends on the inflow conditions and the geometry of the flow domain (Ra & Chang, 1990). Kaiktsis *et al.* (1991) found that the interaction of the separated shear layer with the freestream results in an unsteady oscillatory flow behavior, which causes the reattachment location to oscillate. Direct numerical simulations (DNS, Le *et al.*, 1997) were used to explain the time variation of the streamwise reattachment location as a consequence of shedding large-scale structures in the separated region.

In the case of a diffuser, the flow dynamics is similar to that of a backward-facing ramp. Herbst *et al.* (2007) conducted simulations of turbulent flow in an axisymmetric diffuser and showed that the separation region is governed by the jet-like inflow, which penetrates further into the diffuser. DNS of three-dimensional separated flow in a diffuser by Ohlsson *et al.* (2010) concluded that the position of the separation line likely determines the overall diffuser pressure recovery. However, Cherry *et al.* (2008) demonstrated that there is significant sensitivity to the geometry in separated flow, and therefore, the flow behavior is problem-dependent. In other flow configurations such as turbulent flow over a backward-facing smooth, curved ramp, Song *et al.* (2000) observed elongation and lifting of eddies, which scale with the ramp height, in the region of an adverse pressure gradient. El-Askary (2009) conducted LES for flow over a smooth, curved ramp and reported the existence of high Reynolds stresses in the separated region.

The high stresses and losses associated with the separated region are often undesirable and call for flow control strategies. As explained in section 1.2.1, passive VGs have shown to be effective when the point of separation is fixed spatially, such as in the flow configuration of a backward-facing ramp. This dissertation work focusses on investigating the interactions of the VG-induced flow structures with that of the large-scale structures in the separated region. For that purpose, a wall-mounted cube is used as a canonical VG, and it is expected that the interaction of the horseshoe vortex with the hairpin structures in the separation region over the ramp will lead to flow modulation. For this canonical study, the three questions stated in section 1.2.1 can be reformulated more specifically in fluid dynamics terms: (i) how the VG height affects the interaction between the horseshoe vortex and the separated region? (ii) how the position of the VG, relative to the leading ramp edge, affects the interaction between the horseshoe vortex and the separated region? (iii) in the case of multiple VGs (array of cubes), what is the effect of the spanwise spacing between the neighboring VGs on the interaction between the horseshoe vortex and the separated region? The canonical study proposed here will help us understand the fundamentals physics underlying the modulation of spatially evolving TBL flow over a backward-facing ramp and the transport mechanism that reduces the size of the separated region.

1.3 Challenges in large-scale simulations of turbulent flows

It is known that in a turbulent flow, the range of significantly excited scales of motion in both space and time is of the order $Re^{3/4}$, and as Re increases, so does the range of scales. Many problems of scientific interest belong in the high Reynolds number regimes. Therefore, to calculate a time-dependent high Reynolds number flow in three dimensions, it is necessary to perform order $(Re^{3/4})^4 = Re^3$ computational work (Sagaut, 2001). Thus, a flow at Reynolds number $2Re$ requires roughly 10 times more computational work and poses serious limitations on numerical simulations of high Reynolds number and complex flows. This fact, amongst other challenges in computational physics, has spurred improvements in areas such as computational hardware, distributed memory computing, linear and nonlinear solver technologies, and algorithmic development.

In practice, numerical approximations of high Reynolds number turbulent flows is achieved by solving a set of surrogate equations, which contain a reduced range of scales. The process of reducing the range of scales is called filtering. Consider the set of governing equations that describe a given multiscale problem as

$$\frac{\partial u}{\partial t} + \mathcal{F}(u) = 0, \qquad x \in \Omega, \ t \in [0, t], \tag{1.3}$$

subjected to boundary and initial conditions, where $\mathcal{F}$ is a non-linear differential operator and Ω is

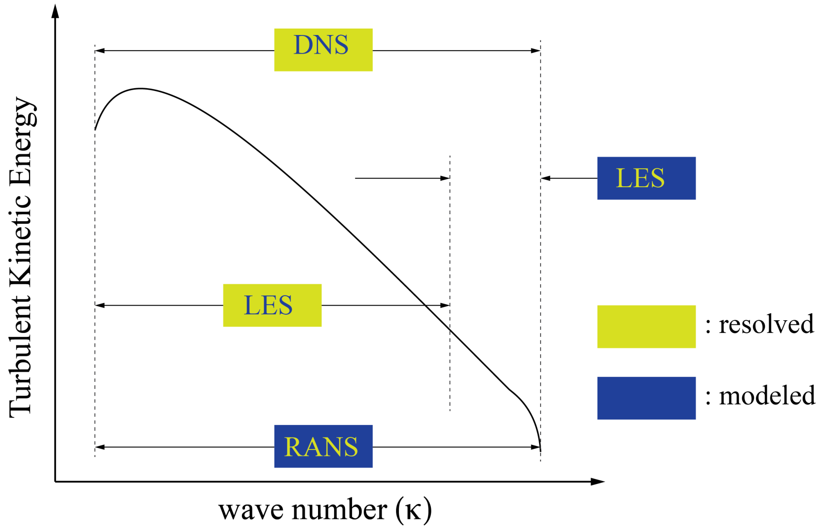


Figure 1.8: Illustration of the energy spectrum in a turbulent flow showing the different modeling techniques.

the spatial computational domain. When a filter $\mathcal{K}$ is applied to governing equations,

$$\mathcal{K}\frac{\partial u}{\partial t} + \mathcal{K}\mathcal{F}(u) = 0, \tag{1.4}$$

the range of scales in equation 1.3 can be reduced. Defining the filtered variable as $\mathcal{K}u = \tilde{u}$, adding and subtracting $\mathcal{F}(\tilde{u})$, and assuming that the filter is invariant leads to the filtered equations of motions,

$$\frac{\partial \tilde{u}}{\partial t} + \mathcal{F}(\tilde{u}) + [\mathcal{K}\mathcal{F}(u) - \mathcal{F}(\tilde{u})] = 0. \tag{1.5}$$

The scales represented in the evolution of the filtered variable in equation 1.5 can be significantly lower than that of the original governing equation (Pope, 2000). However, the bracketed terms in equation 1.5 has the full state u, which is an unknown quantity in the filtered problem and results in a *closure problem*. The unclosed term is typically modeled by a subgrid-scale model, which only depends on the filtered state. The type of filtering and the choice of subgrid-scale defines the modeling approach used.

In turbulent flows, common modeling techniques are the Reynolds-averaged Navier-Stokes (RANS), and large-eddy simulation (LES) approaches. Figure 1.8 shows a graphical illustration of an energy spectrum in an idealized turbulent flow (Davidson, 2004). While the RANS approaches use an averaging filter and model all the scales, the LES approach resolves the larger energy-producing eddies and models the residual motions. The RANS approach is the most widely used method for the practical simulation of turbulent flows and is successful in computing the first-order statistics (Slotnick *et al.*, 2014; Wilcox, 1998). However, RANS models are known to perform poorly in capturing the unsteady dynamics (Rodi, 1997). The lack of accuracy of RANS methods has led to great interest in LES methods, which provides an intermediary level of accuracy (Bose *et al.*, 2010) between the RANS-based methods and the direct numerical simulations (DNS), which resolves all the scales in the flow problem. In the context of wall-bounded turbulent boundary layer flow, this dissertation leverages the LES method, in particular, wall-resolved LES (Frère *et al.*, 2018) approach to investigate the modulation of the separation of turbulent flow. In order to understand the transport of energy in the separated region, a high wall-resolution is required with no wall-modeling (Bose & Park, 2018), such that the near-wall resolution is $\Delta_y^+ = 1.0$ in the wall-normal direction. Here the dimensionless grid spacing in wall coordinates is given as $\Delta^+ = (\Delta u_\tau/\nu)$ where Δ is the grid spacing in physical dimensions, u_τ is the friction velocity at the wall and ν is the kinematic viscosity of the fluid. Details of different modeling strategies are laid out in the literature by Pope (2000), Berselli *et al.* (2005), and Davidson (2004), among others.

The innovations in computer hardware and distributed computing have enabled advances in the modeling of complex turbulent flows. For example, figure 1.9 shows how the increase in available floating-point operations per second (FLOPS) has enabled research and development of different parts or components of an aircraft using different modeling techniques. A multiphysics LES over a full aircraft would require on order 10^{21} operations Buttner (2019), which is not achievable at present, even on the largest supercomputers. The “Vision 2030” study (Slotnick *et al.*, 2014), presented a technology road map to lay the foundation for the development of a future framework and environment where physics-based, accurate predictions of complex turbulent flows, including flow separation, can be accomplished. Figure 1.10 shows this road map, where different technologies are rated based on their readiness level (TRL) as low (red), medium (yellow), and high (green).

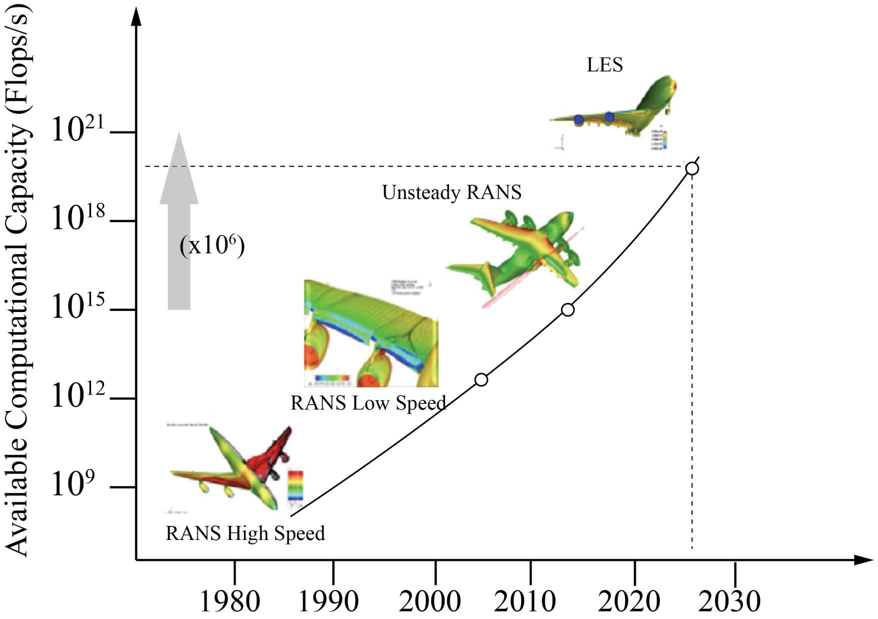


Figure 1.9: Research and development of aircraft parts and components using different turbulent modeling techniques. Adapted from Buttner (2019).

While LES is an ongoing area of research (low TRL), it is plausible that wall-resolved LES for complex 3D flows can be used at appropriate Reynolds numbers.

1.3.1 Simulations of turbulent flows on the next-generation HPC platforms

Advancing science in the areas including climate science, ocean surface modeling, high-energy-density physics, and related fields, requires the development of the next-generation computational models to satisfy the accuracy and fidelity needs of the targeted problems. The potential impact of these models on computational physics is twofold: (i) researchers can account for more physics of the problem under consideration, and (ii) increases in the resolution of the system variables, such as the number of spatial zones, or time steps, improves the simulation accuracy. Both of these impacts place higher demands on computational hardware and software.

To meet these science needs, the computational capabilities of the fastest supercomputers must continue to grow. However, the transition from current sub-petascale and petascale computing to exascale computing is expected to be as disruptive as the transition from vector to parallel computing in the 1990s (Dongarra *et al.*, 2014). Historically, these developments have been driven by

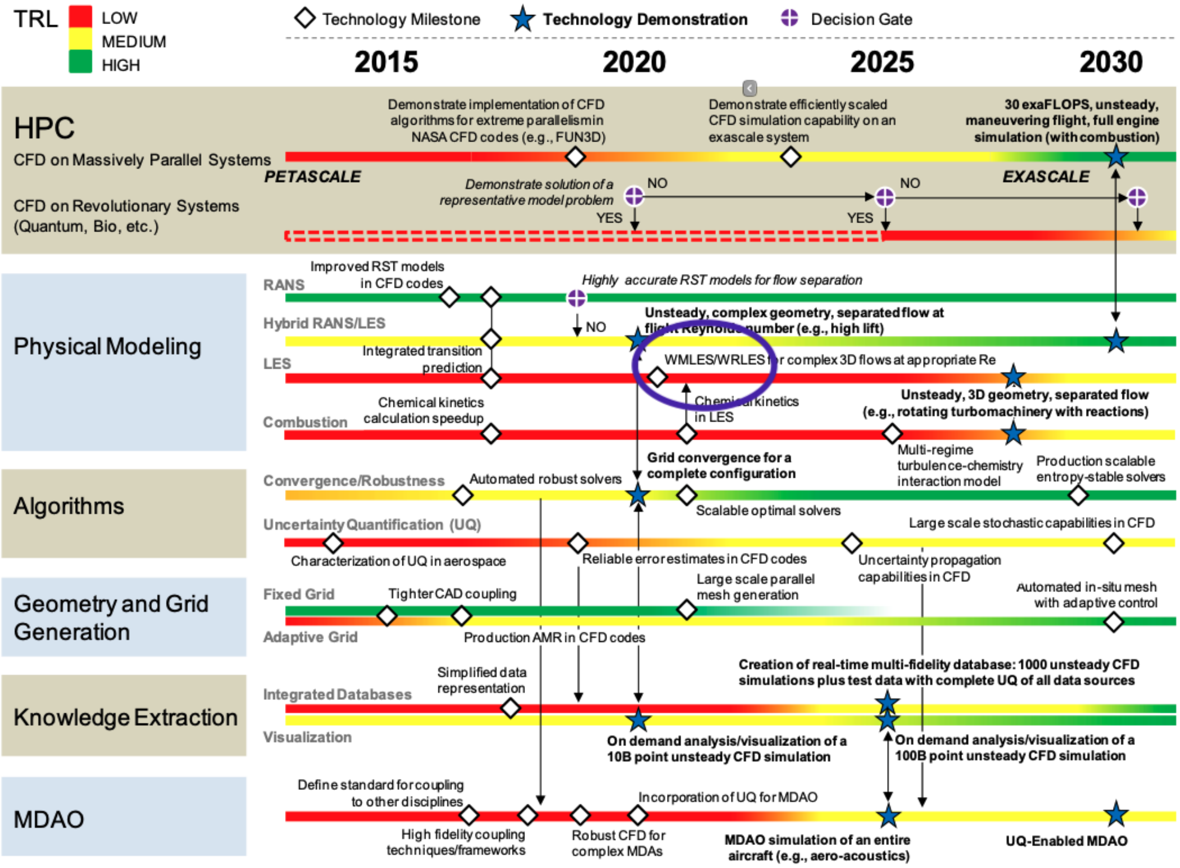

TRL
🟥 LOW
🟨 MEDIUM
🟩 HIGH
◇ Technology Milestone
★ Technology Demonstration
Decision Gate
2015
2020
2025
2030
HPC
CFD on Massively Parallel Systems
CFD on Revolutionary Systems (Quantum, Bio, etc.)
PETASCALE
EXASCALE
Demonstrate implementation of CFD algorithms for extreme parallelism in NASA CFD codes (e.g., FUN3D)
Demonstrate efficiently scaled CFD simulation capability on an exascale system
30 exaFLOPS, unsteady, maneuvering flight, full engine simulation (with combustion)
Demonstrate solution of a representative model problem
NO
YES
NO
YES
Physical Modeling
RANS
Hybrid RANS/LES
LES
Combustion
Improved RST models in CFD codes
Highly accurate RST models for flow separation
NO
Unsteady, complex geometry, separated flow at flight Reynolds number (e.g., high lift)
Integrated transition prediction
WMLES/WRLES for complex 3D flows at appropriate Re
Chemical kinetics calculation speedup
Chemical kinetics in LES
Unsteady, 3D geometry, separated flow (e.g., rotating turbomachinery with reactions)
Multi-regime turbulence-chemistry interaction model
Algorithms
Convergence/Robustness
Automated robust solvers
Grid convergence for a complete configuration
Production scalable entropy-stable solvers
Scalable optimal solvers
Uncertainty Quantification (UQ)
Large scale stochastic capabilities in CFD
Characterization of UQ in aerospace
Reliable error estimates in CFD codes
Uncertainty propagation capabilities in CFD
Geometry and Grid Generation
Fixed Grid
Adaptive Grid
Tighter CAD coupling
Large scale parallel mesh generation
Automated in-situ mesh with adaptive control
Production AMR in CFD codes
Knowledge Extraction
Integrated Databases
Visualization
Simplified data representation
Creation of real-time multi-fidelity database: 1000 unsteady CFD simulations plus test data with complete UQ of all data sources
On demand analysis/visualization of a 10B point unsteady CFD simulation
On demand analysis/visualization of a 100B point unsteady CFD simulation
MDAO
Define standard for coupling to other disciplines
Incorporation of UQ for MDAO
High fidelity coupling techniques/frameworks
Robust CFD for complex MDAs
MDAO simulation of an entire aircraft (e.g., aero-acoustics)
UQ-Enabled MDAO

Figure 1.10: CFD 2030 Vision: Technology development roadmap. Slotnick *et al.* (2014), Reproduced with permission from NASA.

the notion that larger computing systems enable simulations of higher fidelity while minimizing numerical artifacts. However, this strategy of adding more CPUs to computing clusters to achieve greater FLOP counts, motivated by Moore's law, has shifted in recent years, in large part due to the resulting power requirements (200 MW) and corresponding costs, as well as reaching Dennard's scaling (Dennard *et al.*, 1974) namely, limitations in memory bandwidth and capacity. Instead, exascale (1 exaflop = 10^{18} FLOPS) machines, are anticipated to consist of heterogeneous architectures with reduced clock speeds and memory per processor. The proposed exascale architectures present significant challenges for scalable software development and deployment.

An accurate representation of complex flow phenomena can only be achieved if the discretization error is small, which can be achieved either by increasing the mesh resolution or by employing high-order methods. High-order methods are usually defined as having an order of accuracy greater or equal to two, which implies that the error E from the numerical discretization, decreases as $E \sim O(h^n)$, where h is the characteristic grid spacing and $n \geq 2$. Tan *et al.* (2005), Desjardins *et al.* (2008), Bermejo-Moreno *et al.* (2013), Colella *et al.* (2011), Loffeld & Hittinger (2019), and others have explored the high-order variants of the traditional finite difference (FD) and finite volume (FV) methods, generally used in spatial discretization of the governing equations, and have reported that high-order schemes achieve the desired accuracy on a lower resolution mesh as compared to their respective low-order versions. This attribute of high-order methods makes them more desirable, especially for complex flow problems. However, it is well known that traditional FD schemes at high-orders are prone to aliasing errors (Rogallo & Moin, 1984), which can lead to violation of the invariance of the governing equations with erroneous results, and do not perform well on unstructured meshes that are refuired for complex geometries. On the other hand, high-order schemes in both FD and FV methods rely on large numerical stencils, which dictate how much information from the neighboring cells is needed to compute the solution approximation.

The dependency of high-order FD and FV methods on large stencils introduces several complications. In parallel computation, the increased stencil leads to more data movement and increased communication time. Boundary conditions can be difficult to implement and require the addition of ghost cells. For implicit time solvers, these high-order FV methods with larger stencils require more memory and adversely impact the stability of iterative algorithms (Fidkowski, 2004; Mavriplis, 2002). Such issues become more prominent when conducting massively parallel simulations such as the simulations of Bermejo-Moreno *et al.* (2013) and Godenschwager *et al.* (2013) with over a trillion cells on more than a million cores. Moreover, as the trends in high-performance computing (HPC) shift towards exascale computing and beyond, the gains in performance are coming from additional computational units. On these heterogeneous architectures, the decreasing power cost of FLOP has exposed the power cost of data motion. Thus, FLOP is no longer the primary (on-node) cost factor for numerical simulation. A new trade-off must be made between data motion, memory usage, and operations (Brown *et al.*, 2010; Ashby *et al.*, 2010; Lucas *et al.*, 2014; Heroux *et al.*, 2020).

The discontinuous Galerkin (DG) method combines the aspects of the finite element and the FV methods (Cockburn *et al.*, 2000). Arbitrary high-orders of accuracy can be achieved by adding degrees of freedom to each element to represent the solution as a high-order polynomial. Since the solution is allowed to be discontinuous across elements, borrowing from the FV method, the flux between immediately adjacent elements is used to exchange information, which preserves a compact stencil (Henry de Frahan, 2016). Thus, as demonstrated by Heinecke *et al.* (2014), Houba *et al.* (2019), and others, the DG method can scale easily on modern HPC platforms. Recovery-assisted DG method (RADG, Johnson, 2019) interpolates a high-order flux at the element interfaces by using the information from adjacent elements. Therefore, RADG preserves a compact stencil and further increases the accuracy of the solution approximation. While the DG methods present many advantages, the large number of degrees of freedom inside each element poses a challenge for on-node data motion and memory usage.

Furthermore, in scientific computing, most applications involving numerical computations with large dynamic range are performed using either the double-precision (binary64) or the single-precision (binary32) floating-point types, described by the IEEE 754 standard (Zuras *et al.*, 2008). In these applications, the execution of FLOP emerges as a significant contributor to energy consumption. An experimental investigation by Gautschi *et al.* (2017) shows that more than 50% of the energy consumption for a floating-point-intensive application comes from the FLOP and moving the operands from data memory to registers and vice versa. When such intensive calculations are performed at large-scale on modern HPC platforms, the power consumption and temperature control pose a developmental bottleneck (Deng *et al.*, 2013). The issue of power efficiency has garnered considerable concern in the supercomputing platform design and usage. At high-orders, the large number of degrees of freedom in the DG method is expected to increase the total FLOP count, which can potentially affect the power consumption of the nodes. Therefore, energy consumption, including challenges in memory usage, data, and task parallelism, suggests that to extract the benefits of modern HPC systems, high-order methods, including DG-based algorithms, must be optimized. The optimization of RADG constitutes the second part of this dissertation.

1.4 Objectives of this thesis

This dissertation focusses on large-scale simulations of complex turbulent flows. It has two parts: (i) physics-based investigations to understand control of flow separation over a ramp, and (ii) a numerical investigation of optimization strategies to improve the parallel efficiency of an in-house, recovery-assisted discontinuous Galerkin framework for large-scale computations of complex turbulent flows on next-generation HPC platforms. The objectives of the two parts are,

1. Modulation of turbulent boundary layer flow: to understand the role of turbulent transport in the modulation of flow separation over a backward-facing ramp using cubic vortex generators. In particular, wall-resolved large-eddy simulations are conducted to investigate the dependence of the flow in the separated region on the configuration of the VGs. The modulation of the separated flow over the ramp is expected to depend on the interaction of the large-scale structures in the separated region with the horseshoe vortex system produced by the cube, which is dictated by the size, proximity of the cube to the ramp, and spacing between the neighboring cubes (in case of multiple VGs). Two sets of studies are conducted—single

cube studies to investigate the dependence on cube height and its positions; and multiple VGS using an array of equally-spaced cubes to study the role of spanwise spacing.

2. Optimization of high-order discontinuous Galerkin method for next-generation HPC platforms: to numerically investigate optimization strategies to improve the parallel efficiency of the in-house, recovery-based discontinuous Galerkin (RADG) framework to facilitate simulations of complex turbulent flows on the next-generation HPC platforms. To minimize (on-node) data transfer and leverage the large FLOP capability on modern HPC nodes, a data tiling strategy is explored that improves the arithmetic intensity, which is the measure of the amount of work done per byte of data transferred, of the RADG framework. In addition, numerical simulations of complex flows require high precision only in a small region of interest. Therefore, it is conjectured that a lower precision calculation can be performed in other regions. Transprecision calculations have other advantages, including low memory usage and possibly reducing energy consumption. A power-aware compute framework is explored that can evaluate and help design a transprecision framework for a given flow problem.

1.5 Thesis outline and contributions

This dissertation is split into two main parts:

1. Modulation of turbulent boundary layer flow: The first part of this thesis is a physics-based study to advance the state-of-the-art in our understanding of the modulation of separated turbulent flow by passive vortex generators. To accomplish the aforementioned objectives, wall-resolved large-eddy simulations of flow over a backward-facing ramp are performed. For our findings to be applicable to many engineering applications, spatially evolving turbulent boundary layer flow over a backward-facing ramp is considered. This computational campaign is conducted using the open source OpenFOAM (Weller *et al.*, 1998) libraries, which have been shown to scale well on HPC platforms. Two sets of studies are considered,

one with a single, isolated wall-mounted cube (Tandon *et al.*, 2020*a*) and the other using an array of equally-spaced cubes (Tandon *et al.*, 2020*b*). These two studies form the basis of different chapters in this part of the thesis.

   (a) The study of the interactions of the salient flow structures that form around a vortex generator (VG) with those of the separated region for flow over a backward-facing ramp with a single, wall-mounted cube is presented in Chapter 2. The dependence of the turbulent transport on the configuration of the cube, namely its height relative to the boundary layer thickness and its position with respect to the leading ramp edge, is illustrated.

   (b) Modulation of flow over a backward-facing ramp with multiple VGs is presented in Chapter 3. A spanwise array of equally-spaced, wall-mounted cubes is used as multiple VGs. The dependence of the flow modulation on the spacing between the neighboring cubes of the array is illustrated.

2. Optimization of high-order discontinuous Galerkin method for next-generation HPC platforms: This is the second part of this thesis, which numerically investigates optimization strategies to increase the parallel efficiency of an in-house recovery-assisted discontinuous Galerkin (RADG) method for next-generation HPC platforms. The different optimization strategies form the basis of individual chapters in this part of the thesis.

   (a) The on-node cost of data transfer is evaluated for the RADG method in Chapter 4. In particular, a class of high-order RADG discretizations for hyperbolic systems of conservation laws is theoretically analyzed for spatial discretizations for polynomial orders one through six in arbitrary dimensions. Three cache models are considered: the limiting cases of no-cache, an infinite cache, and a more practical finite-sized cache model. Models are validated experimentally by measuring floating-point operations and data transfers on an XSEDE Stampede2 Kinght-Landings node. A data tiling strategy in case of finite-sized cache is shown to increase the arithmetic intensity necessary to make better utilization of on-node floating-point capabilities on modern HPC platforms (Tandon & Johnsen, 2020).

   (b) The cost of floating-point operations, in the context of energy consumption, is inspected in Chapter 5. Energy consumption for large-scale simulations on ALCF's Theta supercomputer using high precision (IEEE 754 binary64) and low precision (IEEE 754 binary32) are presented. A power-aware compute framework for RADG is demonstrated that maintains a balance between the desired accuracy and power consumed on modern HPC platforms (Tandon *et al.*, 2020*c*).

3. Important conclusions for the two parts of this thesis are summarized in Chapter 6, highlighting the major contributions and possible avenues for future research.

4. Appendices provide additional details including model validation and verification, case setup and initialization in OpenFOAM, and other details used in the two parts of this thesis.

Part I:

Modulation of Turbulent Boundary Layer Flow

CHAPTER 2

Modulation of Flow Over a Backward-Facing Ramp by a Wall-Mounted Cube

This chapter is adapted from Tandon *et al.* (2020*a*). The separation of spatially evolving turbulent boundary layer flow near regions of adverse pressure gradients has been the subject of numerous studies in the context of flow control. Although many studies have demonstrated the efficacy of passive flow control devices, such as vortex generators (VGs), in reducing the size of the separated region, the interactions between the salient flow structures produced by the VG and those of the separated flow are not fully understood. In this article, wall-resolved large-eddy simulations are conducted at a Reynolds number of 19,600 based on the inlet boundary layer thickness and freestream velocity, to study flow over a backward-facing ramp modulated by a submerged, wall-mounted cube. In particular, the turbulent transport that results in the modulation of the separated flow over the ramp is investigated by varying the size and location of the VG, which in turn is expected to modify the interactions between the VG-induced flow structures and those of the separated region. The horseshoe vortices produced by the cube entrain the freestream turbulent flow towards the plane of symmetry. These localized regions of high vorticity correspond to turbulent kinetic energy production regions, which effectively transfer energy from the freestream to the near-wall regions. While the gradients and the fluctuations scale with the size of the cube and thus lead to more effective modulation for large cubes, for a given cube height the different upstream cube positions affect the behavior of the horseshoe vortex—if placed too close to the leading edge, the horseshoe vortex is not sufficiently strong to affect the large-scale structures of the separated region and if placed too far the dispersed core of the streamwise vortex is unable to modulate the flow over the ramp.

2.1 Introduction

Separation of spatially evolving turbulent boundary layer (TBL) flow over a backward-facing ramp occurs in a variety of internal (diffusers, mixing chambers) and external (airfoils, road vehicles) flow systems. At sufficiently high Reynolds number, pressure changes in the expansion region oppose changes in viscous stresses, and ultimately lead to flow separation, with reattachment beyond the end of the ramp. In the separation bubble, a region of recirculating flow forms, which may give rise to undesirable effects—from reducing lift on aircraft wings to increased drag on vehicles and reducing efficiency in chemical mixing chambers. Given that they have a fixed point of separation, backward-facing steps or ramps (Armaly *et al.*, 1983; Adams, 1984; Westphal *et al.*, 1984; Kourta *et al.*, 2015) are commonly considered to study flow separation. Although the separation point is fixed, the reattachment location depends on the inflow conditions and the geometry of the flow domain (Ra & Chang, 1990). The shear layer exhibits an unsteady behavior as the separated flow and the freestream interact (Kaiktsis *et al.*, 1991) and thus gives rise to a moving reattachment point (Friedrich & Arnal, 1990; Ötügen, 1991). Direct numerical simulations (DNS, Le *et al.*, 1997) were used to explain the time variation of the streamwise reattachment location as a consequence of shedding large-scale structures in the separated region. In the case of a diffuser, Herbst *et al.* (2007) showed that the separation region is governed by the jet-like inflow, which penetrates further into the diffuser. Ohlsson *et al.* (2010) concluded that the position of the separation line likely determines the overall diffuser pressure recovery, though there is significant sensitivity to the geometry (Cherry *et al.*, 2008). Song *et al.* (2000) observed elongation and lifting of eddies, which scale with the ramp height, in the region of adverse pressure gradient for turbulent flow over a backward-facing curved ramp. In addition, El-Askary (2009) reported high Reynolds stresses in the separated region.

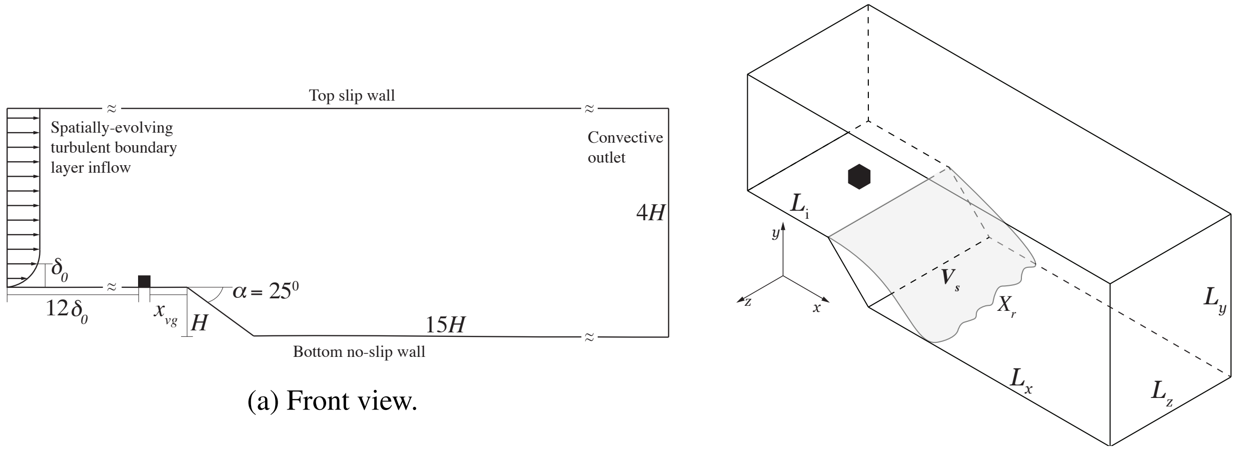
Control of the separated flow could be exploited to reduce the losses described above. Brown *et al.* (1968), Calarese *et al.* (1985), Englar (2001), Logdberg (2006), Mohan *et al.* (2013), Wilson *et al.* (2019), and Fisher *et al.* (2020), among others, explored passive control techniques such as vortex generators (VGs) to reduce drag and improve performance in a variety of engineering applications. VGs are effective when the point of separation is fixed spatially (Rao & Kariya, 1988). Selby *et al.* (1990) used transverse grooves and observed a reduction of the separation bubble size over a backward-facing, curved ramp when the grooves were located one boundary layer thickness upstream of the ramp. Experimental investigation of VGs with different shapes and configurations by Lin (2002) on a backward-facing, curved ramp demonstrated that the submerged VGs of height 65% below the boundary layer thickness generates streamwise vortices that enhance turbulent mixing, and was substantiated in a parallel study by Jenkins *et al.* (2002). Flat-plate boundary layer flow over submerged VGs of different shapes has also been investigated in the past (Ashill *et al.*, 2002; Yao *et al.*, 2002; Elbing *et al.*, 2013; Iyer & Mahesh, 2013). However, it is difficult to generalize findings across the different shapes (e.g., Wishbone, Doublet Wheeler (Lin *et al.*, 1991)) as the flow features appear to be problem dependent. For this reason, a cubic VG is considered as a canonical geometry in this study.

The flow structures that form around a wall-mounted cube placed in fully-developed turbulent flow have been well documented in previous studies. The presence of the cube causes the TBL to break down and form a horseshoe vortex system (HVS) upstream of the cube, with small-scale hairpin and toroidal structures emanating from the top and sides of the cube and a separation bubble in the near wake (Martinuzzi & Tropea, 1993; Krajnovic & Davidson, 2002; Hwang & Yang, 2004). Castro & Robins (1977) concluded that for Reynolds number over $4,000$ based on the cube height there are no discernible Reynolds number effects on the flow separation and reattachment in the near-wake of the cube. However, many external flow systems are characterized by open flow with a spatially evolving TBL, where unlike fully-developed channel flow, the boundary layer thickness grows in the streamwise direction (Wu & Moin, 2009). The presence of an adverse pressure gradient affects the formation of turbulent structures and their corresponding transport inside a spatially evolving TBL (Durbin & Belcher, 1992) and subjects the boundary layer to non-equilibrium turbulence (Aubertine & Eaton, 2005). Shinde *et al.* (2017) showed that for spatially evolving TBL flow over a wall-mounted cube on a flat plate, the strength of the HVS and its relative transport of TKE in the near-wake increases with the cube height relative to the boundary layer thickness. However, the TKE decays within a few cube heights as the streamwise vortices of the cube are weaker than the more complex VGs investigated by Lin (2002).

The objective of this work is to understand the role of turbulent transport in the modulation of flow separation over a backward-facing ramp using cubic VGs. In particular, wall-resolved large-eddy simulations (LES) are used to investigate the dependence of the flow in the separated region on the size and position of the VGs. The modulation of the separated flow over the ramp is expected to depend on the interaction of the large-scale structures in the separated region with the HVS produced by the cube, which is dictated by the size and proximity of the cube to the ramp. The problem set-up and the numerical methods are described in section 2.2. Section 2.3 examines turbulent flow modulation and transport for the baseline case, while section 2.4 studies the dependence of the flow on cube height and position. The article ends with concluding remarks in section 2.5.

2.2 Problem Description

A schematic of spatially evolving turbulent boundary layer flow over a backward-facing ramp with a submerged, wall-mounted cube present near the leading ramp edge is shown in figure 2.1. The 3-D flow domain consists of an inlet section, followed by a ramp of height H at a fixed inclination of $\phi = 25^o$ with the horizontal, and an expansion section. A cube of height h is placed in the inlet, with its downstream face at a distance of x_{vg} from the ramp edge. A turbulent boundary layer flow of thickness δ_0 and freestream velocity U_0 is prescribed at the inlet, with Reynolds number defined as $Re = U_0\delta_0/\nu$. The inlet length is $L_i = 12\delta_0 + h + x_{vg}$ and the length of the expansion section is $L_x = 15H$. The spanwise width is $L_z = 4H$ and the domain height in the expansion section is



(a) Front view.

(b) 3-D computational domain.

Figure 2.1: Backward-facing ramp flow configuration.

$L_y = 4H$.

Wall-resolved large-eddy simulations (LES) are conducted by solving the filtered incompressible Navier-Stokes equations,

$$\frac{\partial}{\partial x_k}\widetilde{u}_k = 0\,, \tag{2.1}$$

$$\frac{\partial}{\partial t}\widetilde{u}_i + \frac{\partial}{\partial x_k}\widetilde{u}_i\widetilde{u}_k - \nu\frac{\partial^2}{\partial x_k^2}\widetilde{u}_i + \frac{\partial}{\partial x_k}\tau^R_{ik} + \frac{1}{\rho}\frac{\partial}{\partial x_i}\widetilde{p} = 0, \tag{2.2}$$

where $\widetilde{u}$ is the filtered velocity, $\widetilde{p}$ is the filtered pressure, ν is the kinematic viscosity, ρ is the density, and $\tau^R_{ik} \equiv \widetilde{u_i u_k} - \widetilde{u}_i\widetilde{u}_k$ is the subgrid-scale stress that requires modeling. The indices i, j, k denote the streamwise (x), wall-normal (y), and spanwise (z) directions, respectively.

A second-order accurate finite volume approach with implicit time marching is used in the OpenFOAM framework (Weller *et al.*, 1998). The equations are solved based on the PIMPLE algorithm—a combination of the PISO (Pressure Implicit with Splitting of Operator, Issa, 1986) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations, Caretto *et al.*, 1973) algorithms—with two outer corrector steps. The dynamic k-equation eddy-viscosity LES model of Kim & Menon (1995) is used as it adequately captures the back-scatter of turbulent kinetic energy in the region upstream of the cube, as well as the unsteadiness of the vortices around the cube (Krajnovic

& Davidson, 2002). A variable time step is used with a maximum Courant number of 1.0. The numerical framework has been verified and validated against the DNS data of Le *et al.* (1997) for flow over a backward-facing step (Tandon *et al.*, 2017). The field quantities and the turbulent statistics are averaged for over 60 flow-throughs starting from $t \approx 750H/U_0$ to avoid contamination by initial transients.

At the inlet, the time-varying velocity is prescribed using a synthetic inflow method (Shinde *et al.*, 2017) for a spatially evolving turbulent boundary layer (TBL) with a freestream turbulence intensity of 1%. The synthetic inflow requires a transition length of approximately $12\delta_0$ to develop into a realistic TBL of desired thickness. The bottom wall of the inlet section, the exposed faces of the cube, the ramp surface, and the bottom wall of the expansion region have no-slip boundary conditions. A no-stress slip wall is applied at the upper boundary of the domain, with

$$v = 0, \quad \frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = 0. \tag{2.3}$$

Similar no-stress slip wall conditions are applied on the lateral sidewalls of the domain. At the outlet, a convective boundary condition is prescribed (Lowery & Reynolds, 1986),

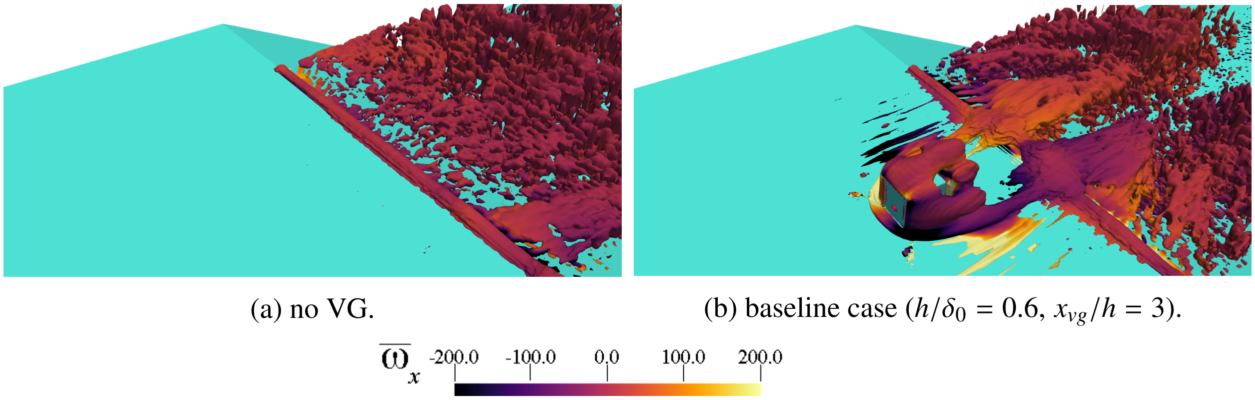
$$\frac{\partial u_i}{\partial t} + U_e \frac{\partial u_i}{\partial x} = 0, \tag{2.4}$$

where U_e is the constant mean exit velocity.

The inclination of the ramp is fixed as $\phi = 25^o$ with respect to the horizontal. Ahmed *et al.* (1984) investigated the variation of drag forces with the slant angle and found $20^o < \phi < 30^o$ to better reduce the overall drag on an idealized car body. A slant angle of $\phi = 25^o$ is a common choice for fundamental studies (Kourta *et al.*, 2015). The Reynolds number $Re_{\delta_0} = U_0\delta_0/\nu$ based on the mean inlet velocity U_0 and the boundary layer thickness δ_0, is $19,600$, which corresponds to $40,000$ based on H or $4,000$ for the smallest h under consideration. The choice of Reynolds number is sufficiently large that there is no discernible *Re*-dependence (Castro & Robins, 1977; Kourta *et al.*, 2015), and it is relevant to a variety of external aerodynamics applications. In accordance with Le *et al.* (1997), the height of the domain is tall enough to investigate the formation of turbulent shear layers on top of the cubes and at the leading edge of the ramp, yet avoid any interactions with the upper boundary of the domain. In addition, the width of the domain $L_z = 4H$ is adequate to prevent lateral confinement (Tandon *et al.*, 2018).

A non-uniform structured grid is employed, where high spatial resolution is achieved through local mesh refinement. The region between $3H$ upstream of the leading edge and $6H$ downstream in the expansion region has a uniform resolution in the spanwise and streamwise flow directions with grid spacing in wall units $\Delta_x^+ \approx \Delta_z^+ \approx 25$. The wall-normal direction has a stretched mesh with the grid spacing near the bottom wall given as $\Delta_y^+ \approx 1$. With this high wall-normal resolution, no wall-model is necessary, thus providing higher fidelity in the reattachment region.

To understand the dependence of the interaction of the horseshoe vortex system with the shear layer on the turbulent transport, different cube heights and positions are considered and compared to the corresponding flow with no VG. Based on Lin (2002), who suggests that VGs with height $h \leq 0.65\delta_0$ are likely to produce strong streamwise vortices that result in effective flow modulation, the baseline case for this work is taken to be $x_{vg} = 3h$ and $h/\delta_0 = 0.6$. For a fixed location $x_{vg}/h = 3$, the following cube heights are considered: $h/\delta_0 = 0.2$, 0.6, and 1.0; for a fixed cube height $h/\delta_0 = 0.6$, the position is varied as followed: $x_{vg}/h = 0$, 3, and 6. These latter values are motivated by studies of flat-plate TBL flow over a wall-mounted VG (Shinde *et al.*, 2017), which indicated that larger h increases the transport of TKE in the near-wake of the cube by the HVS and that the TKE decays within a few cube heights in the near-wake of the cube, with higher decay rate for larger cubes.


(a) no VG.

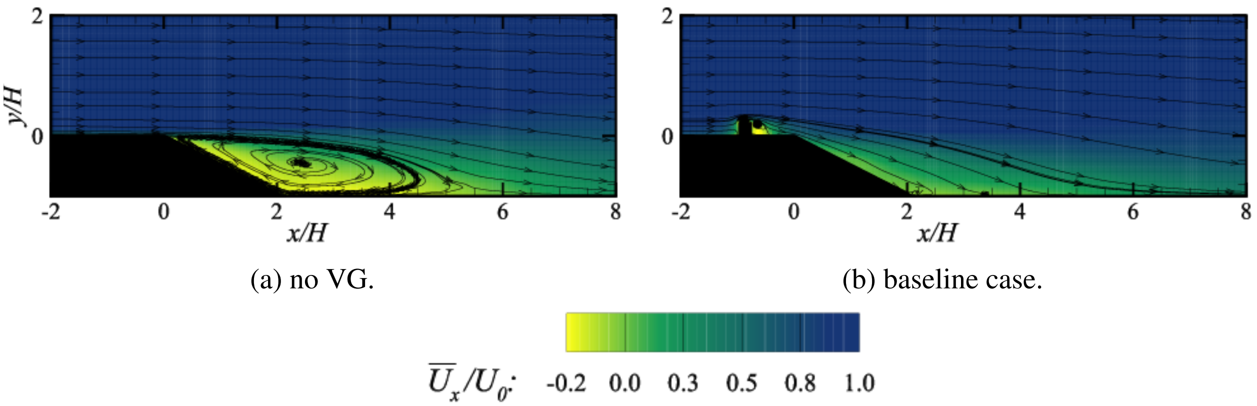
(b) baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$).

Figure 2.2: Iso-surfaces of the $Q-$criterion colored with the time-averaged streamwise vorticity ($\overline{\omega}_x$).

2.3 Modulation of separated flow over a backward-facing ramp by a wall-mounted cube

2.3.1 Velocity and vorticity fields

To provide an understanding of the impact of a vortex generator (VG) on separated flow over a ramp, the flow in the baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$) is first qualitatively compared to that with no VG. Figure 2.2 shows the iso-surfaces of the Q-criterion (Hunt *et al.*, 1988) to illustrate the vortical structures. When there is no VG present, the turbulent boundary layer (TBL) separates at the leading edge and penetrates the expansion region as a shear layer (Herbst *et al.*, 2007). The separated shear layer is dominated by turbulent structures of varying length-scales and vorticity, where the largest structures are on the order of the ramp height (Song *et al.*, 2000; Kourta *et al.*, 2015). In the presence of a VG, the HVS forms near the upstream face of the VG. The legs of the HVS extend in the near-wake to form a connected turbulent structure at the leading ramp edge with a counter-rotating flow pattern. Near the plane of symmetry in the expansion section, the counter-rotating vortex pair interacts with the eddies in the separated region, which increases the size of the counter-rotating vortex pair. Flow is entrained by the counter-rotating vortex pair towards the plane of symmetry and near the wall, which indicates a reduction of the size of the



(a) no VG.

(b) baseline case.

Figure 2.3: Time-averaged streamwise velocity ($\overline{U}_x$) contours with streamlines along the plane of symmetry ($z = 0$) for flow over a backward-facing ramp with and without a VG.

separated region over the ramp due to flow modulation by the VG. To better illustrate the separated region, figure 2.3 shows contours of the time-averaged streamwise velocity component ($\overline{U}_x$) with streamlines along the plane of symmetry for the configuration with no VG and the baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$). In the absence of a VG, the flow forms a recirculation region whose height is of the order of the ramp height H and whose streamwise length is approximately $5H$. In the presence of a VG, the spatially evolving TBL breaks down upstream of the cube and forms small recirculation zones along the top and downstream faces of the cube, as expected (Martinuzzi & Tropea, 1993). A given streamline from the top of the cube shows little deflection to the bottom of the ramp as the flow follows the outline of the recirculation region downstream of the cube and the ramp. Only a small recirculation zone is observed close to the bottom edge of the ramp; otherwise, the flow remains attached to the ramp.

More precise locations of the points of separation and reattachment are obtained by analyzing the streamwise variation of the skin friction coefficient, $C_f = \tau_w / \frac{1}{2}\rho U_\infty^2$, along the bottom wall in the plane of symmetry, where $\tau_w = \mu \left(\partial u/\partial y\right)_{y=0}$ is the shear stress. The point of reattachment X_r is the first point in the expansion section where $C_f > 0$. Figure 2.4 shows the streamwise variation of C_f evaluated based on the time-averaged velocity field. When no VG is present, the mean reattachment location is $\overline{X}_r = 5.7H$. When a VG is present, the reattachment location is $\overline{X}_r = 3.0H$, which is consistent with the smaller separation bubble size in figures 2.2 and 2.3.

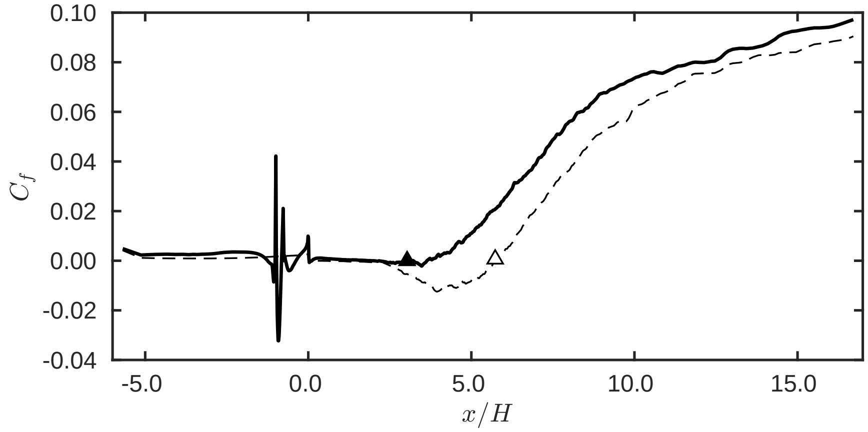


Figure 2.4: Streamwise variation of the skin friction coefficient along the plane of symmetry ($z = 0$) for (——) the baseline case (▲, $\overline{X}_r = 3.0H$) and (– – –) the no VG (△, $\overline{X}_r = 5.7H$) flow.

Analysis of the instantaneous flow field reveals that in both the cases, the reattachment region oscillates at a well defined period about a mean value, consistent with previous studies of separated flow over backward-facing ramps and steps (Kaiktsis *et al.*, 1991; Le *et al.*, 1997; Kourta *et al.*, 2015). Figure 2.5 shows the time evolution of X_r. When no VG is present, the reattachment location oscillates between $5.0H < X_r < 7.5H$, with a period of $20H/U_0$ corresponding to a Strouhal number of 0.05. The presence of a VG gives rise to oscillations of similar amplitude ($2.0H < X_r < 5.5H$) but with a shorter period of $10H/U_0$, corresponding to a Strouhal number of 0.1. This behavior is consistent with the smaller recirculation region size.

The vorticity is examined to better understand the dynamics leading to flow modulation in the separated region. Figure 2.6 shows the time-averaged streamwise vorticity ($\overline{\omega}_x$) contours along the y-z planes in the inlet section and over the ramp. In this figure, the streamwise direction is directed into the page. The spatially evolving TBL breaks down upstream of the cube to form the HVS (Martinuzzi & Tropea, 1993; Hwang & Yang, 2004; Devenport & Simpson, 1990). In figures 2.6a and 2.6b, the y-z plane passes through the downstream face of the cube located at $x/H = -0.75$, which corresponds to $x_{vg}/h = 3$. Localized regions of high vorticity magnitude are visible along the surface of the cube and the bottom wall, corresponding to the HVS. The sign of these high vorticity centers indicates a counter-rotating flow pattern in which the flow from the freestream is

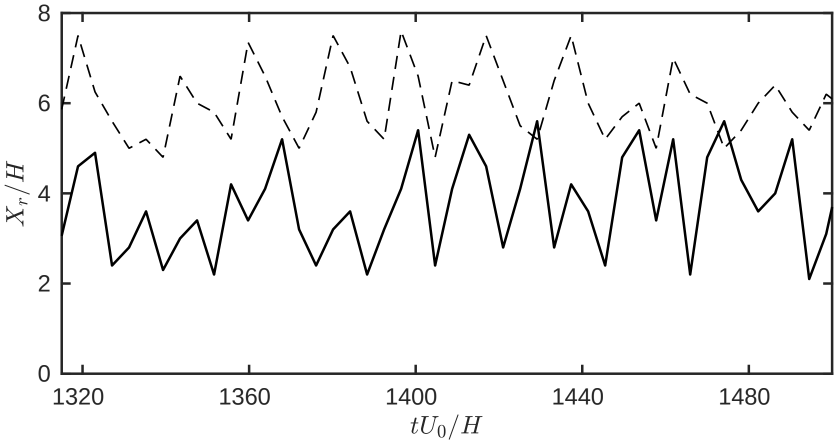
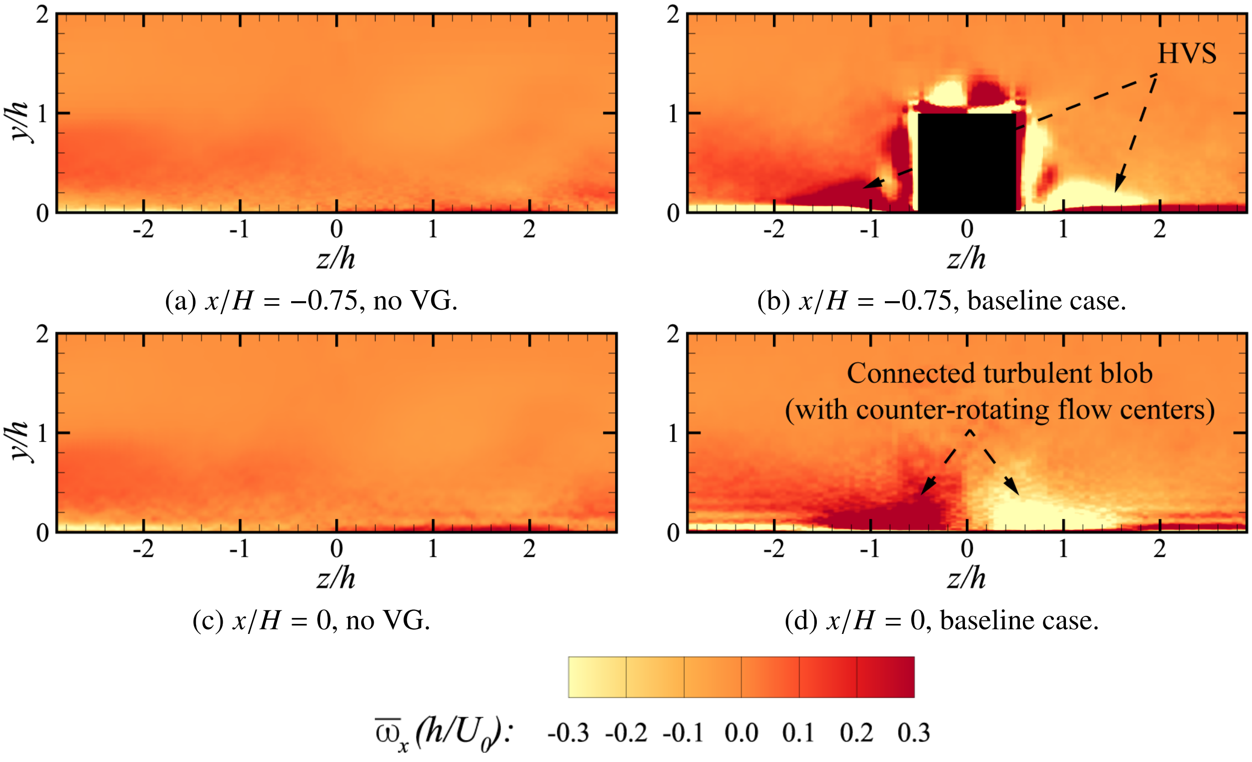


Figure 2.5: Time evolution of the reattachment point for: ——, the baseline case; − − −, no VG flow.

entrained towards the plane of symmetry down towards the bottom wall. The counter-rotating flow gives rise to spanwise velocity in the two legs of the HVS, which draws them closer to one another (Devenport *et al.*, 1997; Iyer & Mahesh, 2013; Leweke *et al.*, 2016; Asselin & Williamson, 2017). This interaction between the HVS legs results in a connected counter-rotating vortex pair near the leading ramp edge at $x/H = 0$. In figures 2.6c and 2.6d, the y-z plane passing through the leading ramp edge at $x/H = 0$ shows the evolution of turbulent structures into a counter-rotating vortex pair in the near-wake of the VG. The vicinity of the counter-rotating vortex pair to the bottom wall subjects it to significant strain, thus increasing its lateral width.

Figure 2.7 shows the interaction of the counter-rotating vortex pair of the VG with the large-scale hairpin structures in the expansion section over the ramp at $x/H = 0.5$. The separated flow over the ramp is dominated by large hairpin structures (Doligalski, 1994; Wu & Moin, 2009) that are of the order of the ramp height (Kourta *et al.*, 2015). The counter-rotating vortex pair with high $\overline{\omega}_x$ stretches, turns, and entrains the turbulent hairpin structures in the separated region around itself (Corsiglia *et al.*, 1976). The flow entrainment increases the size of the counter-rotating vortex pair. Due to the proximity of the counter-rotating vortex pair to the bottom wall in figures 2.6 and 2.7, the shear produced along the wall gives rise to the formation of a secondary vortex sheet (Harvey & Perry, 1971), which is stretched and turned by the primary HVS (Luton & Ragab, 1997; Dehtyriov



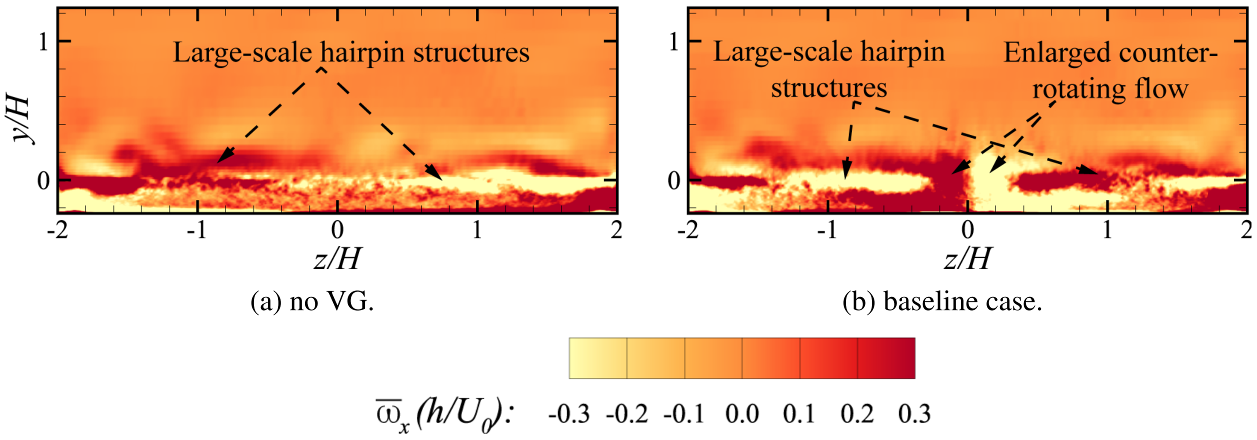
(a) $x/H = -0.75$, no VG.

(b) $x/H = -0.75$, baseline case.

(c) $x/H = 0$, no VG.

(d) $x/H = 0$, baseline case.

Figure 2.6: Time-averaged streamwise vorticity ($\overline{\omega}_x$) contours along y-z planes in the inlet section. The streamwise flow direction ($+x$) is directed into the paper and the downstream face of the cube is shown in black. The inlet section is normalized by h.



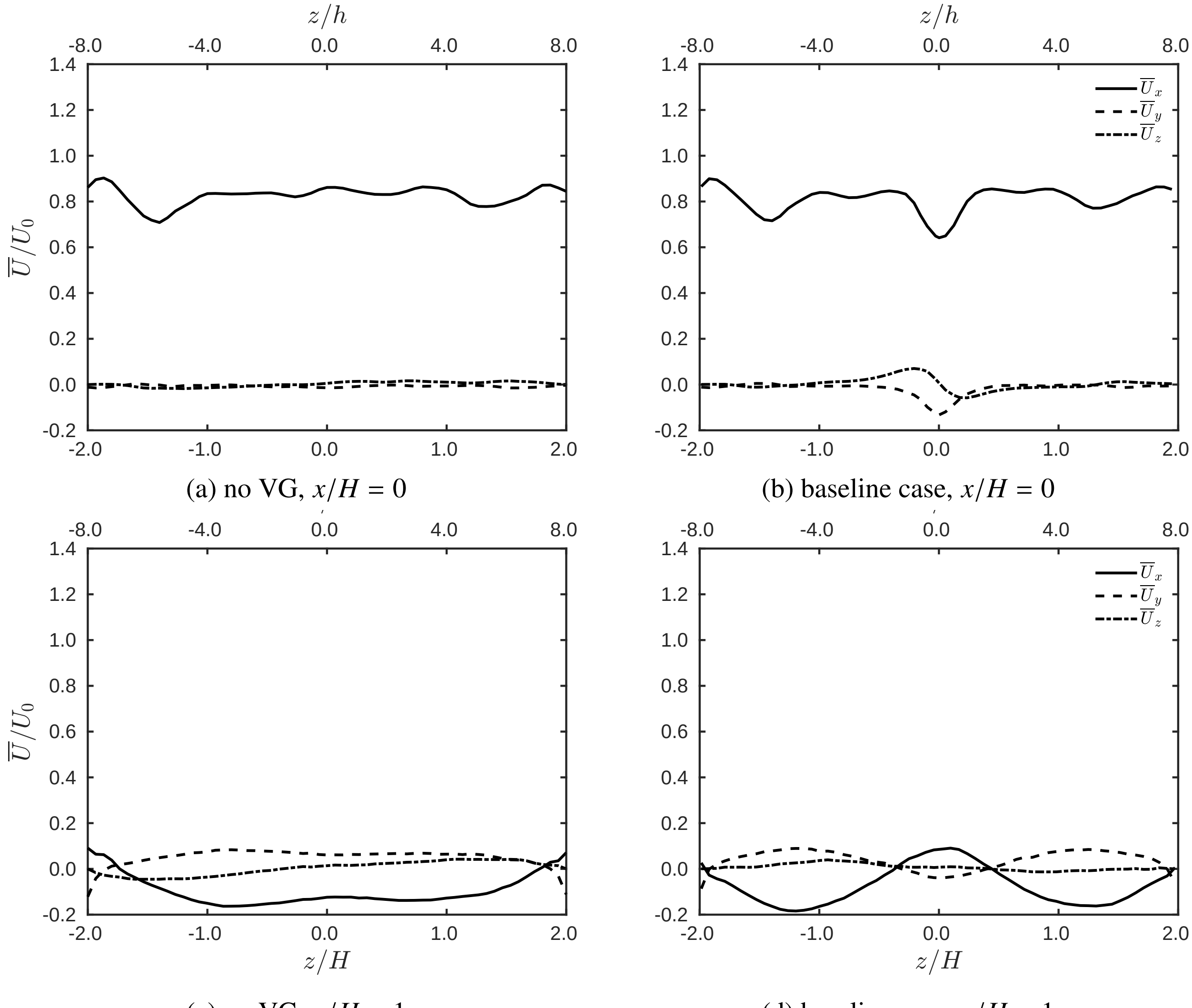
(a) no VG.

(b) baseline case.

Figure 2.7: Time-averaged streamwise vorticity ($\overline{\omega}_x$) contours along y-z plane in the expansion section over the ramp at $x/H = 0.5$. The streamwise flow direction ($+x$) is directed into the paper and the expansion section is normalized by H.

et al., 2020), causing it to spread over a wider spanwise distance. These vorticity contours illustrate that flow modulation in the separated region is caused by interactions of vortex structures produced by the VG with those in the separated region.

The entrainment of freestream fluid toward the ramp is more precisely elucidated by examining the spanwise variation of the time-averaged velocity $\overline{U}$. Figure 2.8 shows the mean velocity components in the spanwise direction near the ramp edge at $x/H = 0$ and over the ramp at $x/H = 1$. The TBL flow in the inlet is attached whether or not a VG is present. The streamwise component $\overline{U}_x$ is an order of magnitude greater than the other components. In the presence of a VG, a 30% decrease in $\overline{U}_x$ in the plane of symmetry between $-1 < z/h < 1$ is observed, while $\overline{U}_y$ becomes negative and $\overline{U}_z$ exhibits symmetric behavior within a length of one cube height. Negative $\overline{U}_y$ indicates that the flow is directed towards the bottom wall, and positive $\overline{U}_z$ in the negative z half-domain implies flow towards the plane of symmetry. Such behavior is characteristic of a counter-rotating vortex pair (Angele & Muhammad-Klingmann, 2005), where the entrainment of flow from the sides and the outer-flow regions towards the near-wall regions leads to reattachment. Over the ramp at $x/H = 1$, the separated flow typically has negative $\overline{U}_x$ and positive $\overline{U}_y$, a manifestation of the recirculation bubble. The counter-rotating vortex pair induced by the VG alters the flow dynamics of the separated region such that near the plane of symmetry (between $-2 < z/h < 2$) $\overline{U}_x$ is positive and



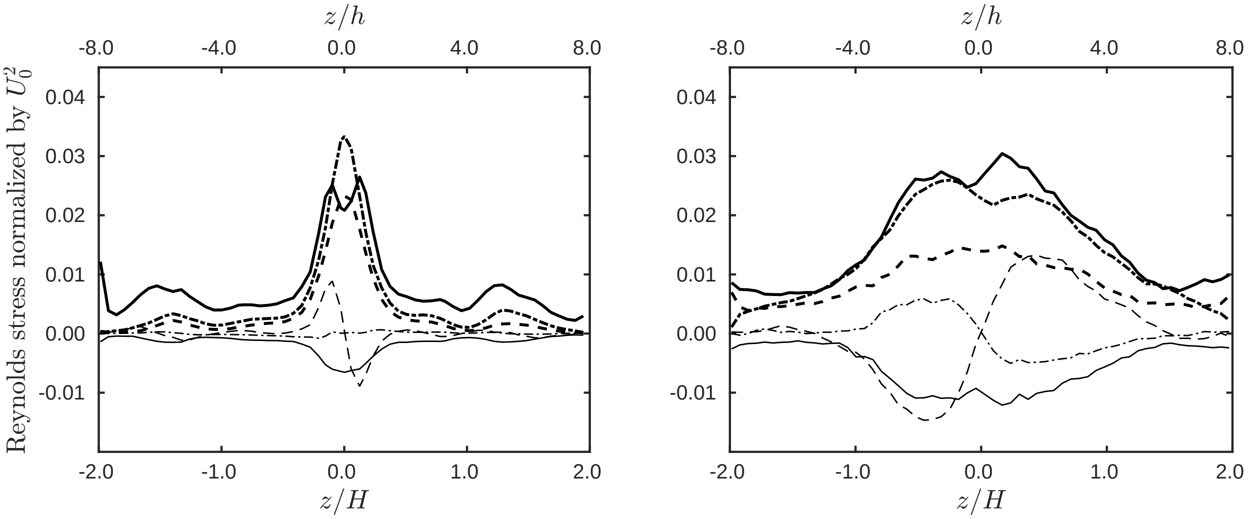
(a) no VG, $x/H = 0$

(b) baseline case, $x/H = 0$

(c) no VG, $x/H = 1$

(d) baseline case, $x/H = 1$

Figure 2.8: Spanwise variation of the time-averaged velocity field ($\overline{U}$) at $x/H = 0$ and $x/H = 1$, and $y = 0.5h$.



(a) $x/H = 0$.

(b) $x/H = 1$.

Figure 2.9: Spanwise variation of the Reynold stresses at $y = 0.5h$ for the baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$): ——, $\overline{u'^2}$; - - - -, $\overline{v'^2}$; -·-·-, $\overline{w'^2}$; ——, $\overline{u'v'}$; — — —, $\overline{u'w'}$; —·—·, $\overline{v'w'}$.

$\overline{U}_y$ is negative, indicating that the flow is attached and directed towards the bottom wall in the downstream direction. The spanwise variations of the mean velocity over the ramp further support the notion that the modulation of flow over the ramp is due to the interaction of the HVS with the separated region.

The entrainment of flow by the counter-rotating HVS towards the plane of symmetry and the wall enhances the momentum of the near-wall flow. The high-momentum flow structure (Bross *et al.*, 2019) affects the changes in viscous stresses and balances the pressure changes in the expansion section, which opposes flow separation. To understand the corresponding momentum transport, figure 2.9 shows the spanwise variation of the Reynolds stress components for the baseline case in the inlet section and over the ramp. Near the leading ramp edge at $x/H = 0$, the positive peaks of $\overline{u'^2}$ on either side of the plane of symmetry correspond to the center of the counter-rotating vortices and indicates transfer of momentum in the streamwise direction by the HVS, which is characteristic of a counter-rotating streamwise flow. The peaks in $\overline{v'^2}$ and $\overline{w'^2}$ in the plane of symmetry represent wall-normal and spanwise transport of momentum by the HVS to the near-wall region and towards the plane of symmetry, respectively. As illustrated by the negative peak of $\overline{u'v'}$, the streamwise momentum ($u' \geq 0$) is transported towards the bottom wall ($v' \leq 0$), *i.e.* from the outer region to the near-wall region. In the negative z half-domain, the streamwise momentum ($u' \geq 0$) is transported towards the positive spanwise direction ($w' \geq 0$), *i.e.* towards the plane of symmetry, giving rise to the positive peak in $\overline{u'w'}$ in the left half. Similar behavior is observed in the expansion region, with the difference being that the turbulent fluctuations in the expansion region are on the order of the ramp height. Therefore, the HVS of the wall-mounted cube enhances the momentum in the near-wall region, which modulates the separated flow and reduces the size of the separation region over the ramp.

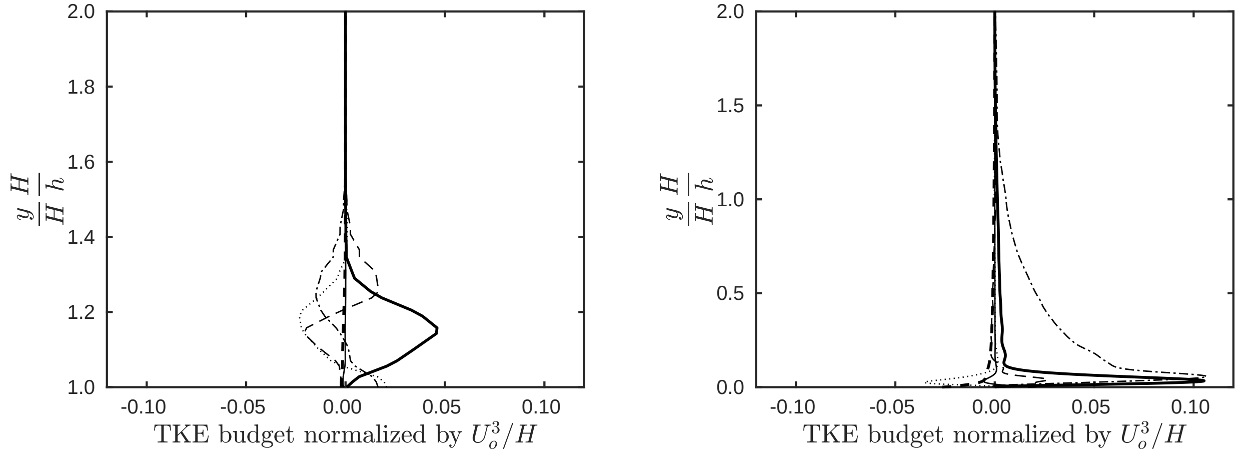
2.3.2 Turbulent kinetic energy transport to the near-wall region

The horseshoe vortex system (HVS) produced by the cube draws fluid from the freestream and injects it into the near-wall region, thus energizing the boundary layer flow. To better understand energy transfer by the VG, the evolution of turbulent kinetic energy (TKE) is considered,

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{1}{2}q^2\right) &= \underbrace{-\frac{1}{2}U_j\overline{(u'_i u'_i)_{,j}}}_{C_k}\underbrace{-\overline{(u'_i u'_j)}U_{i,j}}_{P_k}\underbrace{-\frac{1}{2}\overline{(u'_i u'_i u'_j)_{,j}}}_{T_k} \\ &\quad \underbrace{+\frac{1}{2}\left(\frac{1}{Re}+\nu_\tau\right)\overline{(u'_i u'_i)_{,jj}}}_{D_k}\underbrace{-\frac{1}{2}\left(\frac{1}{Re}+\nu_\tau\right)\overline{(u'_{i,j} u'_{i,j})}}_{\epsilon_k}\underbrace{-\overline{u'_i p'_{,i}}}_{\Pi_k} \qquad (2.5)\end{aligned}$$

where $q^2/2$ is the TKE, Re is the Reynolds number based on mean velocity U_0 and boundary layer thickness δ_0, and ν_τ is the eddy viscosity in the subgrid-scale model. The over-bar represents the time-averaging and the terms on the right hand side of equation 2.5 are convective (C_k), production (P_k), turbulence transport (T_k), viscous diffusion (D_k), viscous dissipation (ϵ_k) and velocity-pressure gradient (Π_k).

In the presence of a VG, the production and transport of TKE downstream in the wake of the cube by the shear layer formed along the top and flow structures generated around the cube is better understood by inspecting the terms of the TKE budget in equation 2.5 along different sections of the ramp. Figure 2.10 shows the spanwise-averaged TKE contributions in the upstream inlet section



(a) $x/H = -0.875$ $(x/h = -3.5)$.

(b) $x/H = 0$.

Figure 2.10: Normalized span-averaged TKE contributions in the inlet section for baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$): –·–·, convection; ——, production; – – –, transport; ——, diffusion; - - - -, dissipation; · · · · · ·, velocity-pressure gradient.

for the baseline case. The main contributions to the TKE production $(\overline{(u'_i u'_j)} U_{i,j})$ are the steep velocity gradients along the surfaces of the cube and the Reynolds stress tensor, which depends on the turbulent structures that form around the cube. The spike in TKE production in figure 2.10a lies inside the shear layer ($y/h \approx 1.2$) produced at the top of the cube and corresponds to the negative peak of the turbulent transport term, which indicates TKE transfer away from the outer regions of the shear layer to the near-wall region at the top of the cube. At the leading ramp edge, the spanwise velocity gradient is zero near the plane of symmetry, such that the only term contributing to the production of TKE in the plane of symmetry is $\overline{u'v'} U_{x,y}$. On the other hand, the counter-rotating flow imparts spanwise velocity to the HVS due to which $\overline{u'w'} U_{x,z}$ also contributes to the production of TKE. Therefore, in figure 2.10b the attached flow near the leading edge has significant TKE production while energy is transferred from the freestream to the near-wall regions. The positive convective term within $y/h < 1.0$ implies that the HVS convects the TKE introduced by the cube downstream to the near-wall region and towards the plane of symmetry. For flow around the VG in the inlet section, dissipation of TKE is negligible relative to its production. Figure 2.11 shows the ratio P_k/ϵ_k along the y-z plane near the downstream face of the cube ($x/H = -0.75$). Consistent with the observations of Shinde (2018), regions with large P_k/ϵ_k ratio exist around the VG, thus

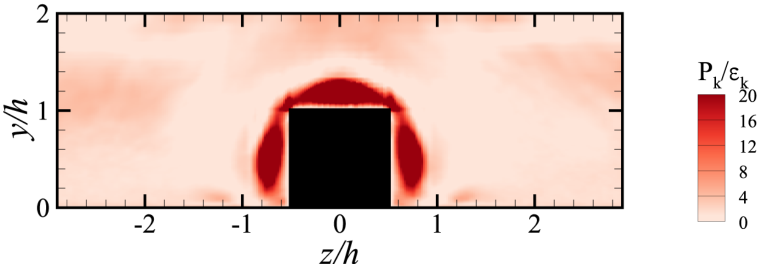
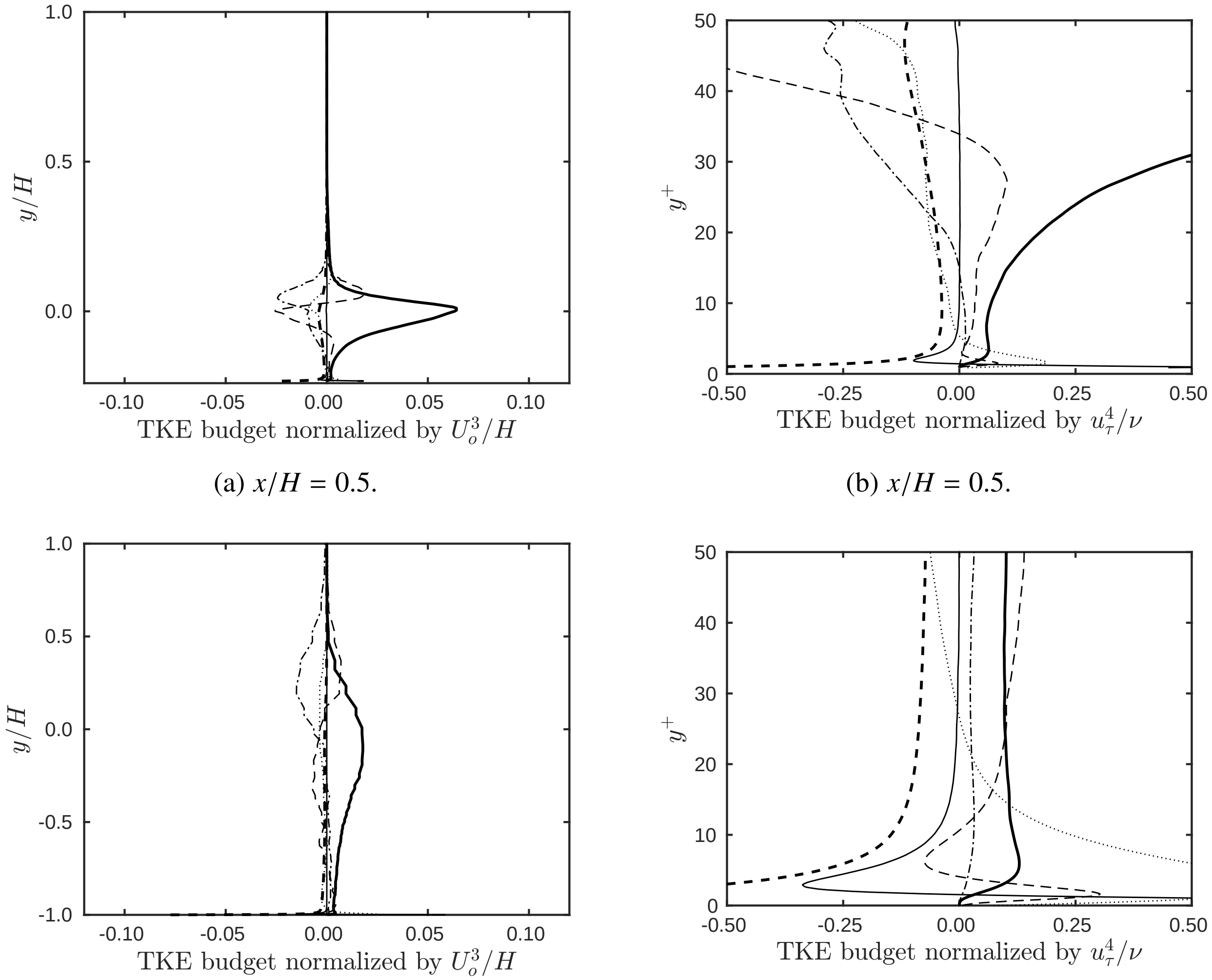


Figure 2.11: Non-equilibrium turbulent regions observed along the y-z plane at $x/H = -0.75$ for the baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$).

denoting regions of non-equilibrium turbulence.

The TKE contributions in the expansion region are shown in figures 2.12 and 2.13 both in terms of outer and inner coordinates to illustrate the dependence of the large-scale structures on the outer variables (mean velocity U_0 and ramp height H) and the variation in near-wall behavior with respect to the inner variables (friction velocity u_τ and kinematic viscosity ν). In the recirculation region, between $0 < x/H < 6$, the larger turbulent fluctuations introduced by the counter-rotating flow and the resulting vortex breakdown leads to a higher magnitude of TKE production than in the inlet section. The impingement of the shear layer at the reattachment location $x/H \approx 6$ generates fluctuations and leads to production of TKE. In the outer region, low dissipation with high TKE production is consistent with the observations of Le *et al.* (1997) for a backward-facing step. The negative peak in the turbulent transport signals energy transfer to the near-wall region. As expected, the dissipation and diffusion dominate inside the viscous layer for $y^+ < 5$.

Figure 2.13 shows the TKE contributions in the recovery zone at locations $x/H = 10$ and $x/H = 15$. The significant decrease in the magnitude of TKE production in the outer regions and the higher dissipation and diffusion, especially in the near-wall region, explains the overall drop in the magnitude of the TKE. Moreover, the attached flow results in a favorable pressure gradient with $\overline{p'_{,x}} < 0$, such that the velocity-pressure gradient has a positive peak within $y^+ < 5$. Positive convection indicates that the attached flow convects the TKE in the streamwise direction towards the plane of symmetry. At $15H$ downstream from the ramp, the TBL is in a state of non-equilibrium turbulence as the production of TKE is not balanced by its dissipation. In addition, the flow exhibits



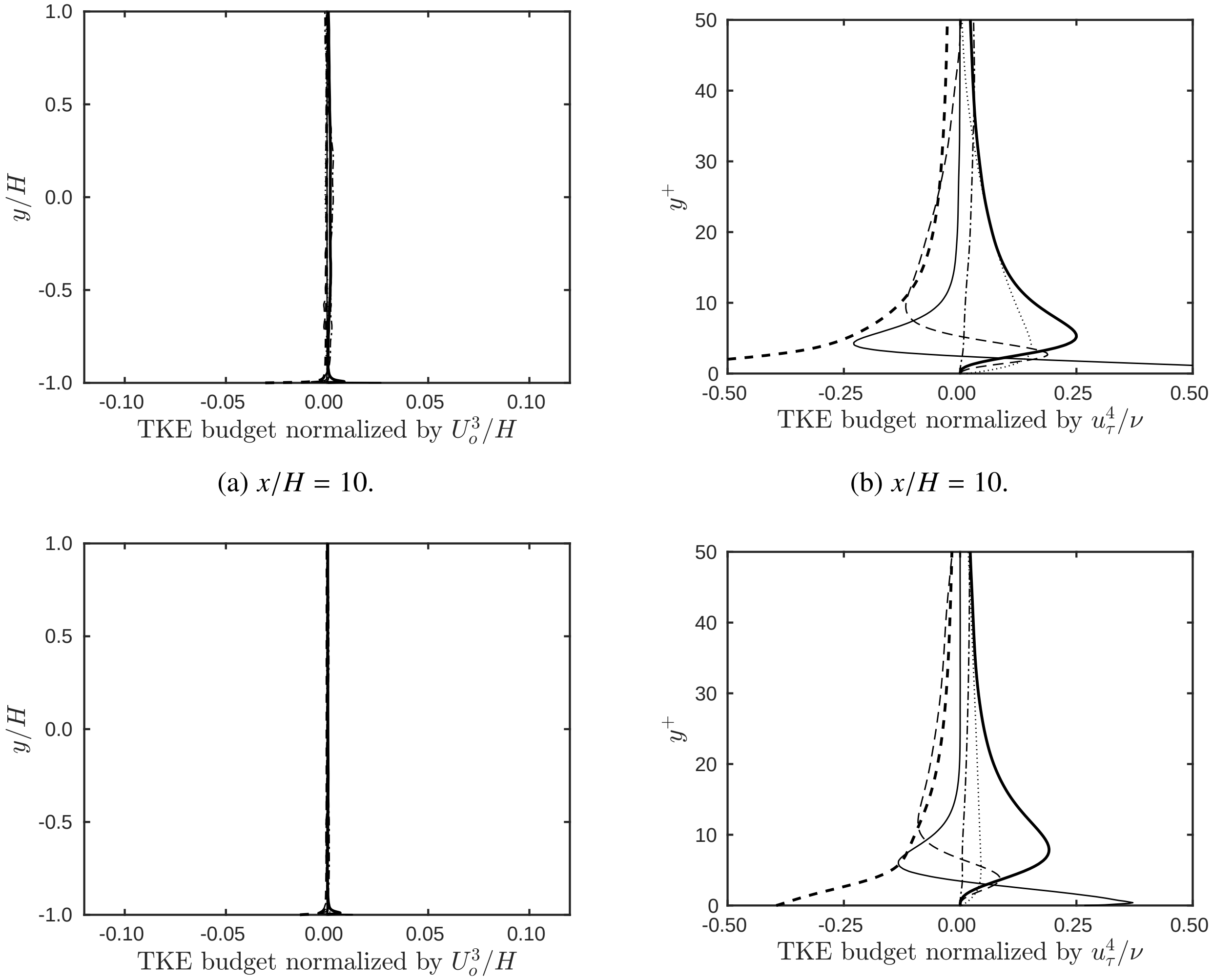
(a) $x/H = 0.5$.

(b) $x/H = 0.5$.

(c) $x/H = 6$.

(d) $x/H = 6$.

Figure 2.12: Normalized span-averaged TKE contributions in the recirculation region for baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$): −·−·, convection; ——, production; − − −, transport; ——, diffusion; - - - -, dissipation; ··· ···, velocity-pressure gradient.



(a) x/H = 10.

(b) x/H = 10.

(c) x/H = 15.

(d) x/H = 15.

Figure 2.13: Normalized span-averaged TKE contributions in the recovery region for baseline case (h/δ_0 = 0.6, x_{vg}/h = 3): −·−·, convection; ——, production; − − −, transport; ——, diffusion; - - - -, dissipation; · · · · · ·, velocity-pressure gradient.

non-zero transport and non-zero convection of TKE inside the shear layer, which implies continual transfer of energy in the streamwise direction and towards the bottom wall. Although boundary layer recovery is not the primary focus of this work, it is observed that the TBL does not recover by the end of the domain.

2.4 Dependence of the flow modulation on the cube configuration

The intensity of the interaction of the horseshoe vortex system with the separated region is expected to depend both of the cube size and its proximity to the ramp edge. This section examines the flow dependence on these two parameters.

2.4.1 Dependence of the flow modulation on the cube height

First, the dependence of the flow modulation in the separated region on the cube height (relative to the boundary layer thickness) is examined by considering a fixed cube location of $x_{vg}/h = 3$ and varying the cube heights $h/\delta_0 = 0.2$, 0.6, and 1.0. Earlier studies of Escauriaza & Sotiropoulos (2011) and Krajnovic & Davidson (2002) on turbulent boundary layer (TBL) flow around a bluff body placed in a channel showed that the formation of the HSV upstream of the cube causes the flow to separate at an upstream location X_s^{vg} where the backflow from the cube meets the incoming inflow. Thereafter, the flow reattaches at a downstream location X_r^{vg} in the near-wake of the cube, such that the flow is attached within a few cube heights downstream. The point of separation and reattachment are obtained from the skin friction coefficient C_f. Figure 2.14 shows the streamwise variation of C_f for different cube heights in the inlet section near the VG, evaluated based on the time-averaged velocity. Table 2.1 lists the values of X_s^{vg} and X_r^{vg} obtained from figure 2.14 for the cube heights under consideration. Consistent with the study of Shinde *et al.* (2017), when scaled by h, the separation and reattachment lengths of the cube are effectively constant, which indicates that these separated regions upstream and downstream of the cube depend on the cube height only.

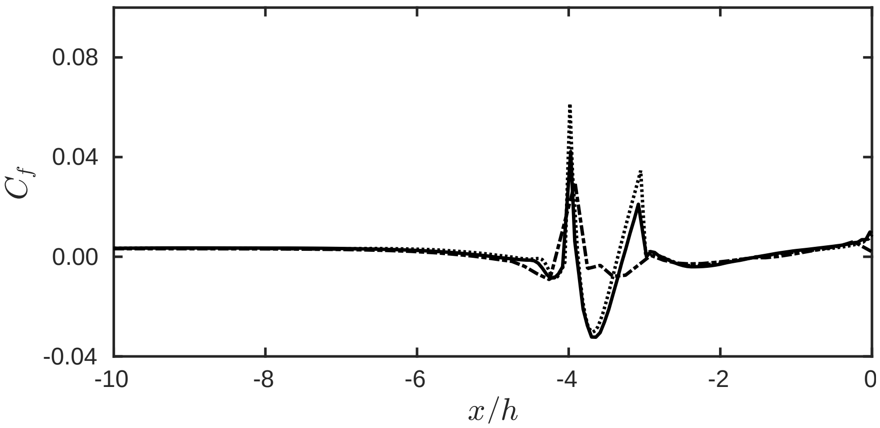


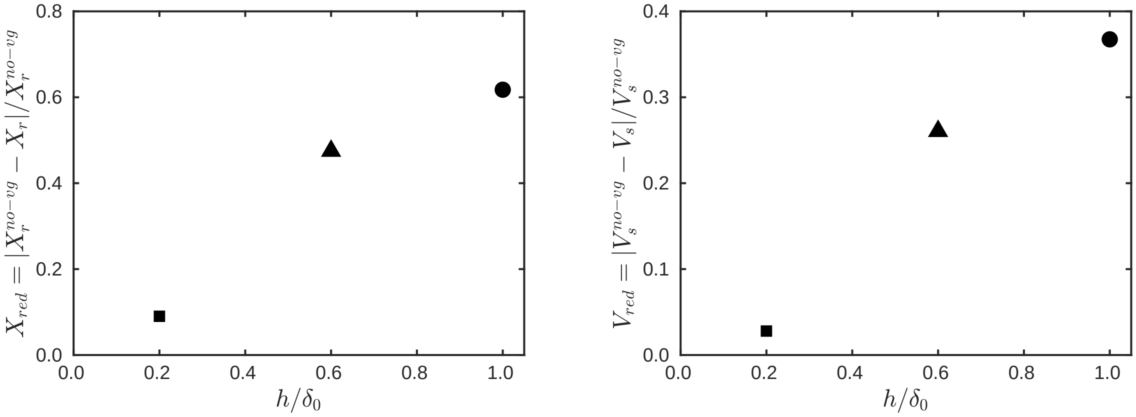
Figure 2.14: Streamwise variation of the skin friction coefficient along the plane of symmetry (z = 0) for cube heights: -·-·-, h/δ_0 = 0.2; ——, h/δ_0 = 0.6; · · · · · ·, h/δ_0 = 1.0.

h/δ_0	x_{vg}/h	X_s^{vg}/h	X_r^{vg}/h
0.2	3.0	1.08	1.50
0.6	3.0	0.99	1.51
1.0	3.0	0.93	1.53

Table 2.1: Separation (X_s^{vg}) and reattachment (X_r^{vg}) lengths for the cubes along the plane of symmetry (z = 0).

In the present context, this observation implies that for a larger cube size the reattachment location is closer to the ramp edge. It follows that the size of the separation region (length and volume) depend on the cube height. Figure 2.15 shows the reduction in the length and volume of the separated region relative to the case with no VG. Clearly, an increase in cube height offers more significant modulation of flow over the ramp. However, it must be noted that varying the cube height also affects the total forces acting on the cube. The coefficient of drag C_d is evaluated by adding the contributions from the skin friction on the exposed faces of the cube and the form drag due to flow separation around the cube. Table 2.2 lists the values of mean reattachment location $\overline{X}_r$ and C_d for each cube height. For the h/δ_0 = 1.0 cube, the reduction in the volume of the separated region is approximately 40% more than that of the h/δ_0 = 0.6 cube, but C_d increases by 25%.

To better understand the energy transfer, the evolution of turbulent kinetic energy (TKE) is examined for the different cube heights. Figure 2.16 shows the spanwise-averaged TKE contributions



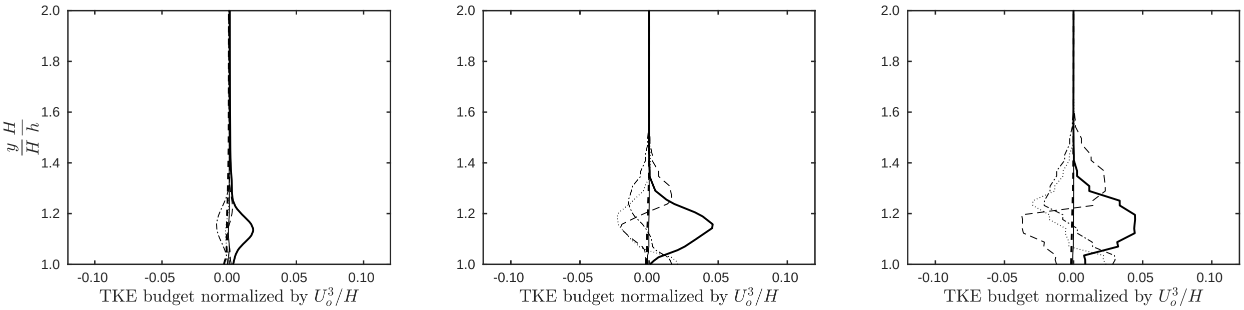
(a) Reduction of the separation length.

(b) Reduction of the separation volume.

Figure 2.15: Reduction in the size of the separation region for cube heights: ■, $h/\delta_0 = 0.2$; ▲, $h/\delta_0 = 0.6$; ●, $h/\delta_0 = 1.0$.

h/δ_0	x_{vg}/h	$\overline{X}_r/H$	V_s/H^3	C_d
no VG	–	5.73	6.97	–
0.2	3.0	5.22	6.79	0.6
0.6	3.0	3.01	5.17	0.8
1.0	3.0	2.19	4.42	1.0

Table 2.2: Mean reattachment location ($\overline{X}_r$) measured along the plane of symmetry ($z = 0$), the volume of separation (V_s), and the coefficient of drag (C_d) for different cube heights.



(a) $h/\delta_0 = 0.2$.

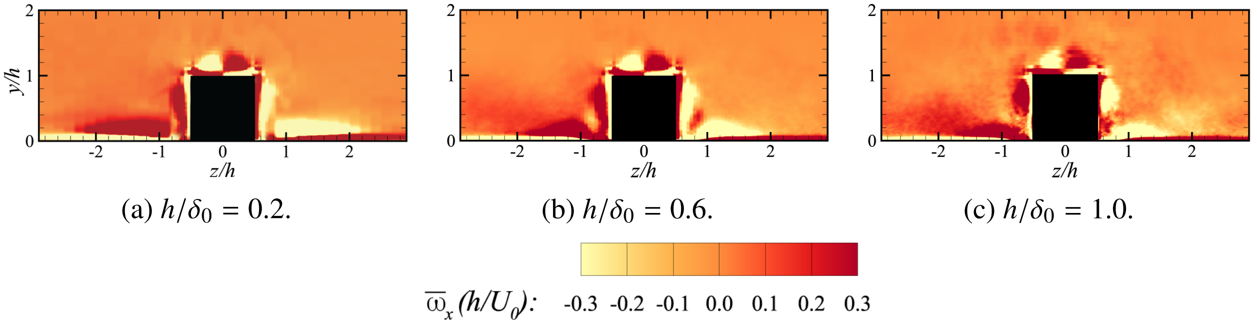
(b) $h/\delta_0 = 0.6$.

(c) $h/\delta_0 = 1.0$.

Figure 2.16: Normalized span-averaged TKE contributions at the top of the cube at $x/h = -3.5$ for different cube heights: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\ \cdots$, velocity-pressure gradient.

from equation 2.5 on the top of the VG ($x/h - 3.5$) for the three cube heights. The production of TKE in the shear layer at the top of the cube increases with increasing cube height as the contributions to TKE production from the turbulent fluctuations and the velocity gradients increase with increasing cube height. As described in the previous section, the positive peak of TKE production corresponds to the negative peak in TKE transport, which indicates transfer of energy from outer shear layer to the near-wall region. The magnitude of negative peak of TKE transport increases for larger cube heights, which indicates that the energy transfer depends on the cube height. The influence of cube height on the size of turbulent structures can be visualized by examining vorticity contours. Figure 2.17 shows $\overline{w}_x$ contours along the y-z plane passing through the downstream face of the cube at $x_{vg} = 3h$ for all the three cube heights. The horseshoe vortex system (HVS) scales with the cube height, as its height is approximately $0.4h$ for all cubes. However, the spreading of the HVS in the spanwise direction is inversely related to the cube height. Figure 2.18 shows pressure contours in the neighbourhood of the cubes to quantify the role of spanwise pressure gradients present in such flows (Simpson, 2001). The cube with $h/\delta_0 = 1.0$, due to its larger size, results in a larger region under the influence of spanwise pressure gradient. Hence, the HVS of cube with $h/\delta_0 = 1.0$ is compact and constrained closer to the cube compared to smaller cube heights.

The compact HVS of larger wall-mounted cubes increases the vortex strength of the HVS, thereby increasing the effective entrainment for flow modulation. The vortex strength Γ is evaluated

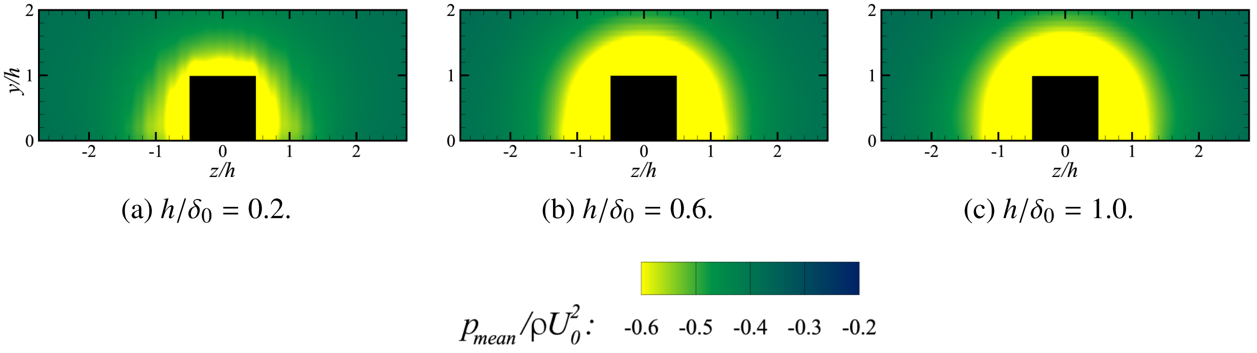


(a) h/δ_0 = 0.2.

(b) h/δ_0 = 0.6.

(c) h/δ_0 = 1.0.

Figure 2.17: Contours of the streamwise component of the time-averaged vorticity ($\overline{\omega}_x$) in the upstream inlet section at $x/H = -0.75$ for different cube heights. The black face depicts the downstream face of the cube. The streamwise flow direction ($+x$) is directed into the paper and the inlet section is normalized by h.



(a) h/δ_0 = 0.2.

(b) h/δ_0 = 0.6.

(c) h/δ_0 = 1.0.

Figure 2.18: Contours of the time-averaged pressure along the y-z plane at $x/H = -0.75$ upstream form the leading ramp edge for different cube heights. The black face depicts the downstream face of the cube and the inlet section is normalized by h.

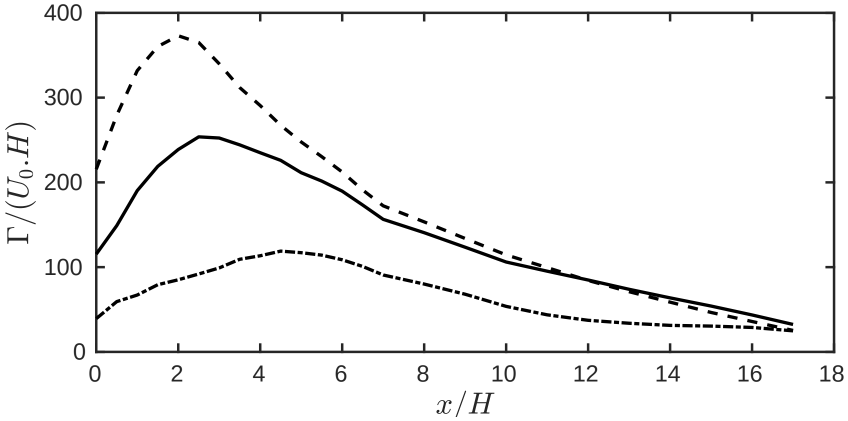
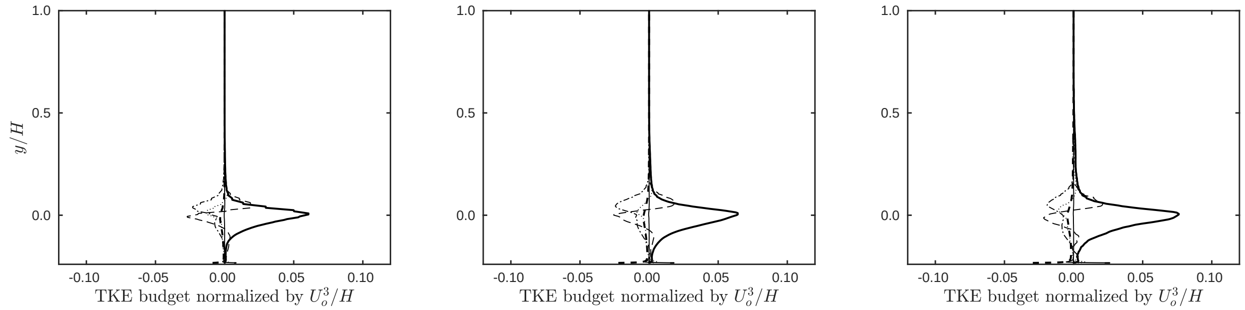


Figure 2.19: Streamwise vortex strength of the streamwise vortices for the cube heights: -·-·-, $h/\delta_0 = 0.2$; ——, $h/\delta_0 = 0.6$; - - - -, $h/\delta_0 = 1.0$.

by considering the area integral of $\overline{w}_x$ inside a fixed rectangular region along a y-z plane of width $2H$ in the positive z direction and height $y = 2H$ from the bottom. By considering a half-domain in the spanwise direction, the vortex strength of one of the legs of the counter-rotating pair of the HVS can be examined. Figure 2.19 shows the streamwise vortex strength of the HVS. As expected, the strength of the HVS is larger for larger cubes. Between $0 < x/H < 6$, the separated shear layer is subjected to large strain and deformation due to which the vortices disperse and their circulation decreases. Thereafter, strong dissipation near the bottom wall in the attached flow further dissipates the turbulent structures, giving rise to a decrease in vortex strength.

As explained in section 2.3, the HVS entrains the large-scale structures in the separated region, thus modulating the flow near the plane of symmetry. The strength and size of the HVS affect the entrainment: the stronger and larger HVS of the $h/\delta_0 = 1.0$ cube entrains more flow from the freestream to the near-wall regions, consequently increasing the transport of momentum. In addition, the size and strength of the HVS also affects the turbulent fluctuations that are generated due to the flow interactions in the separated region, which in turn affects the TKE production and transfer in the expansion region. Figure 2.20 shows the spanwise-averaged TKE contributions over the ramp at $x/H = 0.5$. The production and transfer of TKE increases with the cube height. Similar to the baseline case, the separated shear layer has negligible dissipation of TKE for all the cube heights considered.



(a) $h/\delta_0 = 0.2$.

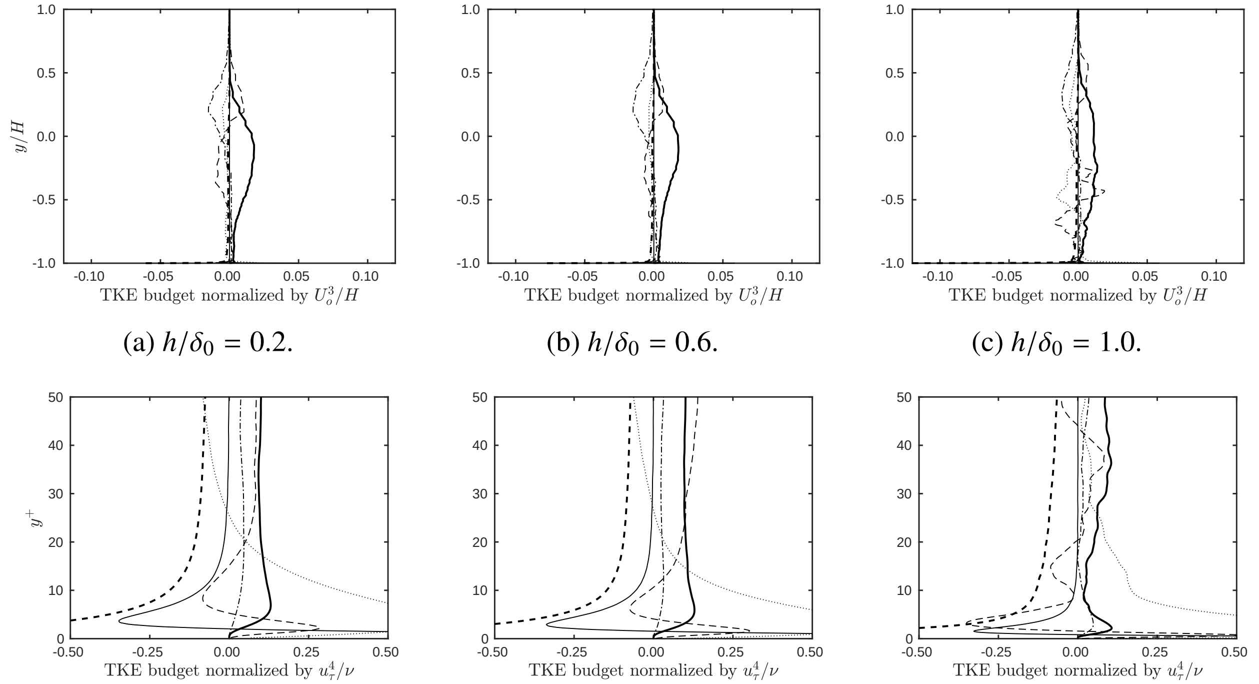
(b) $h/\delta_0 = 0.6$.

(c) $h/\delta_0 = 1.0$.

Figure 2.20: Normalized span-averaged TKE contributions at $x/H = 0.5$ for different cube heights: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\cdots$, velocity-pressure gradient.

Figures 2.21 and 2.22 show the spanwise-averaged TKE contributions in both inner and outer-coordinates to examine the effect of cube height on the transfer of energy in the expansion section. At $x/H = 6$, the relative decrease in the TKE production for all the cases is attributed to the decay of the strength of the counter-rotating vortex pair in the expansion region. The strain and the velocity gradients in the expansion section increase with the cube height, which facilitates faster decay of turbulent structures for larger cubes. However, in the near-wall region, the peaks of the TKE including production, diffusion, and dissipation shift closer to the bottom wall as the cube height is increased because in the attached flow the height of the boundary layer scales inversely with the cube height. Furthermore, the attached flow has a favorable pressure gradient and convects the TKE in the streamwise direction and towards plane of symmetry.

Figure 2.22 shows the TKE contributions near the end of the domain at $x/H = 15$ for the different cube heights. Compared to the flow in the recirculation region ($0 < x/H < 6$), the TKE production is significantly lower in the outer-flow for all the cube heights and is balanced by dissipation of TKE. However, close to the bottom wall the velocity gradients continue to contribute to the TKE production, which is convected downstream and towards the plane of symmetry.



(a) $h/\delta_0 = 0.2$.

(b) $h/\delta_0 = 0.6$.

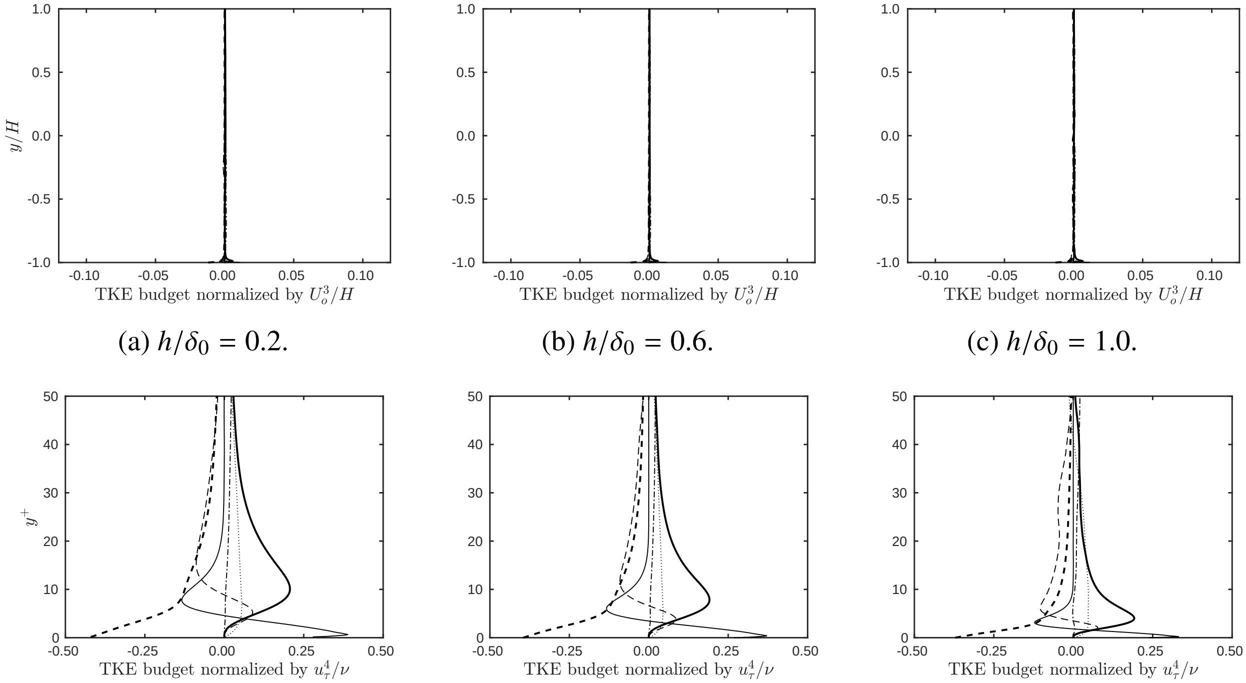
(c) $h/\delta_0 = 1.0$.

(d) $h/\delta_0 = 0.2$.

(e) $h/\delta_0 = 0.6$.

(f) $h/\delta_0 = 1.0$.

Figure 2.21: Normalized span-averaged TKE contributions at $x/H = 6$ for different cube heights: −·−·, convection; ——, production; − − −, transport; ——, diffusion; - - - -, dissipation; ······, velocity-pressure gradient.



(a) $h/\delta_0 = 0.2$.

(b) $h/\delta_0 = 0.6$.

(c) $h/\delta_0 = 1.0$.

(d) $h/\delta_0 = 0.2$.

(e) $h/\delta_0 = 0.6$.

(f) $h/\delta_0 = 1.0$.

Figure 2.22: Normalized span-averaged TKE contributions at $x/H = 15$ for different cube heights: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\cdots$, velocity-pressure gradient.

2.4.2 Dependence of flow modulation on the cube location

The dependence of the flow modulation in the separated region on the proximity of the cube to the leading edge of the ramp is examined by considering cubes of a fixed height $h/\delta_0 = 0.6$ and varying their positions $x_{vg}/h = 0$, 3, and 6. The size of the inlet is adjusted such that the inflow is located $12\delta_0$ from the upstream face of the cube. The flow separation X_s^{vg} and reattachment X_r^{vg} around the cube is obtained from the skin friction coefficient C_f. Figure 2.23 shows the skin friction coefficient in the streamwise direction in the inlet section along the plane of symmetry. Table 2.3 lists the values of X_s^{vg} and X_r^{vg} for the cube positions under consideration. Except for the case $x_{vg}/h = 0$, where the downstream face of the cube is aligned with the leading ramp edge and the separated flow around the cube does not reattach in the near-wake, similar values are obtained for the separation and reattachment points for different cube positions. This behavior indicates that the flow around the cube is primarily influenced by the cube height and has little dependence on the proximity of the cube to the region of adverse pressure gradient, unless the cube is within $1.5h$ from the ramp. Le *et al.* (1997) showed that on a backward-facing step the effect of adverse pressure gradient is observed close to the step and does not affect the flow in the inlet section. Accordingly, the cube placed at $x_{vg}/h = 0$ has a more prominent effect on the pressure changes than the cubes placed farther upstream, which is further substantiated from the skin friction values where the peak of C_f is lower for the cube placed at $x_{vg}/h = 0$ (Durbin & Belcher, 1992). The size of the separate region depends on the interaction between the HSV and the shear layer, and thus on the location of the VG. Figure 2.24 shows the reduction in the size of the separation region relative to that of the flow with no VG. A non-monotonic relation is observed, such that the reduction in separation length is highest for the cube located at $x_{vg}/h = 3$. The reduction in the volume of the separated region is on the same order for all the different cube locations. In addition, the coefficient of drag C_d is evaluated by adding the contributions from the skin friction on the exposed faces of the cube and the form drag due to the separation of flow around the cube. Table 2.4 lists the values of the mean reattachment location $\overline{X}_r$ and C_d for each cube position. For the cube located at $x_{vg}/h = 0$,

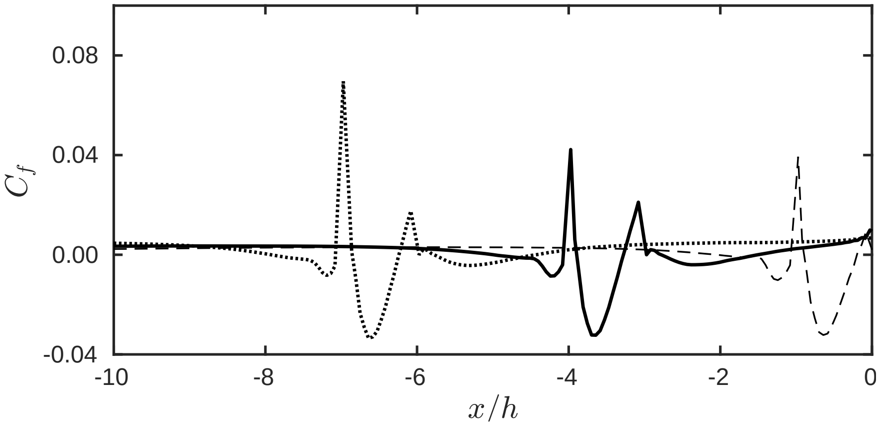


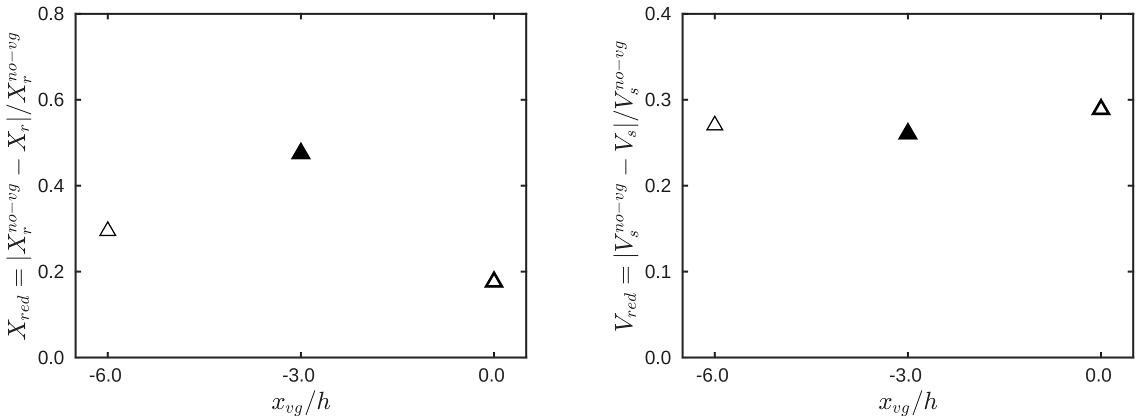
Figure 2.23: Streamwise variation of the skin friction coefficient (C_f) along the plane of symmetry for cube positions: – – –, $x_{vg}/h = 0$; ——, $x_{vg}/h = 3$; $\cdots\cdots$, $x_{vg}/h = 6$.

h/δ_0	x_{vg}/h	X_s^{vg}/h	X_r^{vg}/h
0.6	0.0	1.05	–
0.6	3.0	0.99	1.51
0.6	6.0	0.91	1.61

Table 2.3: Separation (X_s^{vg}) and reattachment (X_r^{vg}) lengths for the cubes along the plane of symmetry ($z = 0$).

the contributions from the form drag due to proximity of the cube to the adverse pressure gradient region increases the overall drag to $C_d \approx 1$. Of the locations considered in the present study, the $x_{vg}/h = 3$ produces the greatest reduction in separation length and separation volume over the ramp, while maintaining a lower coefficient of drag.

The non-monotonic effect on the flow modulation observed in figure 2.24 can be better understood by examining the formation and interaction of turbulent structures in the vorticity contours. Figure 2.25 shows the time-averaged streamwise vorticity contours near the leading ramp edge at $x/H = 0$ for the different cube positions. For the cube located at $x_{vg}/h = 0$, the downstream face of the cube is aligned with the leading ramp edge at $x/H = 0$, and the spatially evolving turbulent boundary layer (TBL) forms localized regions of high vorticity corresponding to the horseshoe vortex system (HVS, Martinuzzi & Tropea, 1993). The case $x_{vg}/h = 3$, corresponds to the baseline case examined in section 2.3, where the HVS extends in the near-wake of the cube to form a



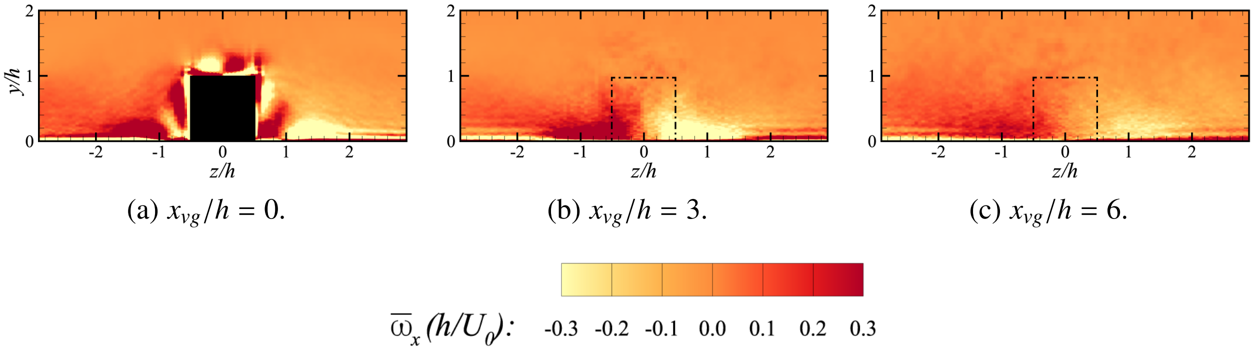
(a) Reduction of the separation length.

(b) Reduction of the separation volume.

Figure 2.24: Reduction in the size of the separation region for cube positions: Δ, $x_{vg}/h = 0$; ▲, $x_{vg}/h = 3$; △, $x_{vg}/h = 6$.

h/δ_0	x_{vg}/h	$\overline{X}_r/H$	V_s/H^3	C_d
no VG	–	5.73	6.97	–
0.6	0.0	4.72	4.97	1.0
0.6	3.0	3.01	5.17	0.8
0.6	6.0	4.04	5.10	0.8

Table 2.4: Mean reattachment location ($\overline{X}_r$) measured along the plane of symmetry ($z = 0$), the volume of separation (V_s), and the coefficient of drag (C_d) for different cube locations.



(a) $x_{vg}/h = 0$.

(b) $x_{vg}/h = 3$.

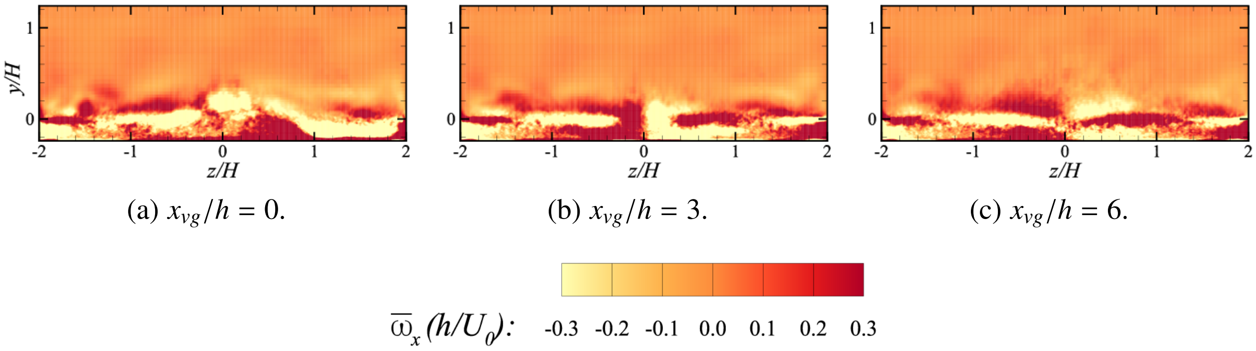
(c) $x_{vg}/h = 6$.

Figure 2.25: Contours of the streamwise component of the time-averaged vorticity ($\overline{\omega}_x$) along the leading ramp edge at $x/H = 0$ for different cube positions. The black face in (a) depicts the downstream face of cube, while the dash-dotted line in (b) and (c) depicts the upstream location of the cube. The streamwise flow direction ($+x$) is directed into the paper and the inlet section is normalized by h.

counter-rotating vortex pair. Although a similar flow behavior is observed for the case $x_{vg}/h = 6$, the vortex pair traverses a larger streamwise distance before encountering the leading ramp edge and is subjected to prolonged diffusion due to wall effects, which results in a more stretched, dispersed and a less intense counter-rotating vortex pair.

To illustrate the interaction of the counter-rotating vortex pair with the large hairpin structures in the separated region, figure 2.26 shows vorticity contours along the y-z plane at $x/H = 0.5$ in the expansion section over the ramp. For the cube placed at $x_{vg}/h = 0$, the small-scale turbulent eddies with high vorticity are asymmetrically turned and stretched by the large hairpin structures in the separated region, allowing modulation in the positive z half-domain. Furthermore, the high strain in the expansion region causes the small-scale structures to decay rapidly, thus reducing the effectiveness of flow modulation. On the other hand, the counter-rotating vortex pair for the cubes placed at $x_{vg}/h = 3$ and 6, modulate the flow along the plane of symmetry, as expected. The dispersed counter-rotating vortex pair in the latter cube configuration modulates flow over a larger spanwise region of $-3 < z/h < 3$.

The effective entrainment of flow by the HVS and the resulting counter-rotating vortex pair depends on its vortex strength. The vortex strength Γ is evaluated by considering the area integral of $\overline{w}_x$ inside a fixed rectangular region along a y-z plane of width $2H$ in the positive z direction and



(a) $x_{vg}/h = 0$. (b) $x_{vg}/h = 3$. (c) $x_{vg}/h = 6$.

Figure 2.26: Contours of the streamwise component of the time-averaged vorticity ($\overline{\omega}_x$) over the ramp at $x/H = 0.5$ for different cube positions. The streamwise flow direction ($+x$) is directed into the paper and the expansion section is normalized by H.

height $y = 2H$ from the bottom. Figure 2.27 examines the vortex strength of the counter-rotating vortex pair for the three cube positions. As expected, the compact HVS for the cube the placed at $x_{vg}/h = 0$ results in high vortex strength at the leading ramp edge. However, the small-scale turbulent eddies decay rapidly in the expansion section and are unable to entrain the large hairpin structures in the separated region by contrast to the larger counter-rotating vortex pair. Thus, for the cube placed at $x_{vg}/h = 0$, the modulation of flow over the ramp is different from that of other cube configurations. On the other hand, the dispersed counter-rotating vortex pair for the cube located at $x_{vg}/h = 6$ has a lower vortex strength at the leading ramp edge and remains consistently lower than that of the cube placed at $x_{vg}/h = 3$, which indicates that the flow modulation for the latter configuration is more effective.

The counter-rotating flow draws the fluid from the freestream and injects it into the near-wall region, thereby energizing the boundary layer flow. The evolution of the turbulent kinetic energy (TKE) in equation 2.5 demonstrates the effect of upstream cube position on the production and transfer of TKE to the inner-wall region. Figure 2.28 shows the spanwise-averaged TKE contributions at the leading ramp edge at $x/H = 0$. For the cube placed at $x_{vg}/h = 0$, the steep velocity gradients near the surface and the turbulent structures formed around the cube, contribute to the production of TKE. In addition, the shear layer separating on the top of the cube results in TKE production in regions beyond $y/h > 1$. The counter-rotating vortex pair of cubes placed at $x_{vg}/h = 3$

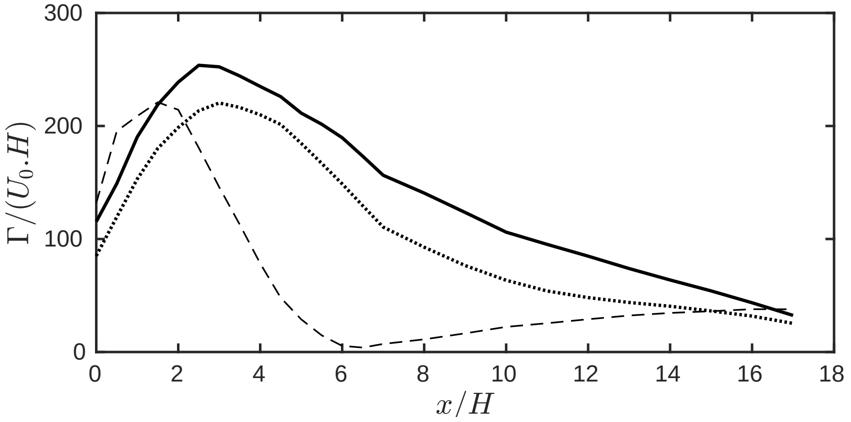
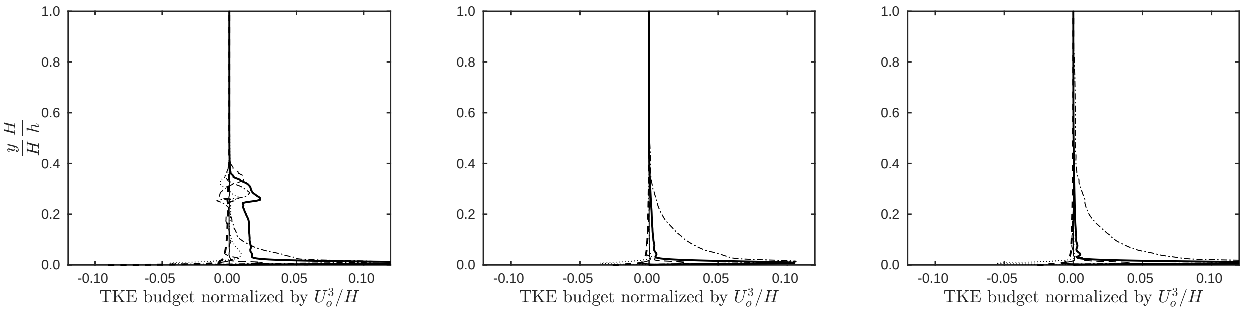


Figure 2.27: Streamwise vortex strength of the time-averaged streamwise vorticity for cube positions: ——, $x_{vg}/h = 0$; ——, $x_{vg}/h = 3$; $\cdots\ \cdots$, $x_{vg}/h = 6$.

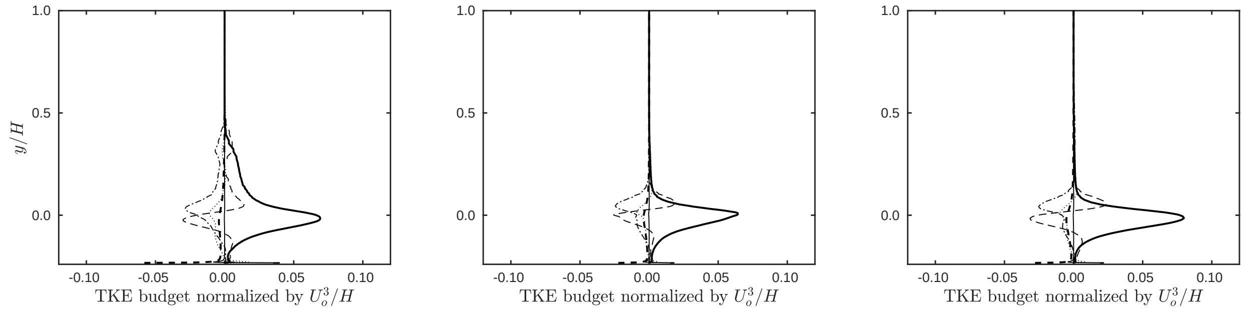


(a) $x_{vg}/h = 0$.

(b) $x_{vg}/h = 3$.

(c) $x_{vg}/h = 6$.

Figure 2.28: Normalized span-averaged TKE contributions near the leading ramp edge at $x/H = 0$ for different cube positions: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\ \cdots$, velocity-pressure gradient.



(a) $x_{vg}/h = 0$.

(b) $x_{vg}/h = 3$.

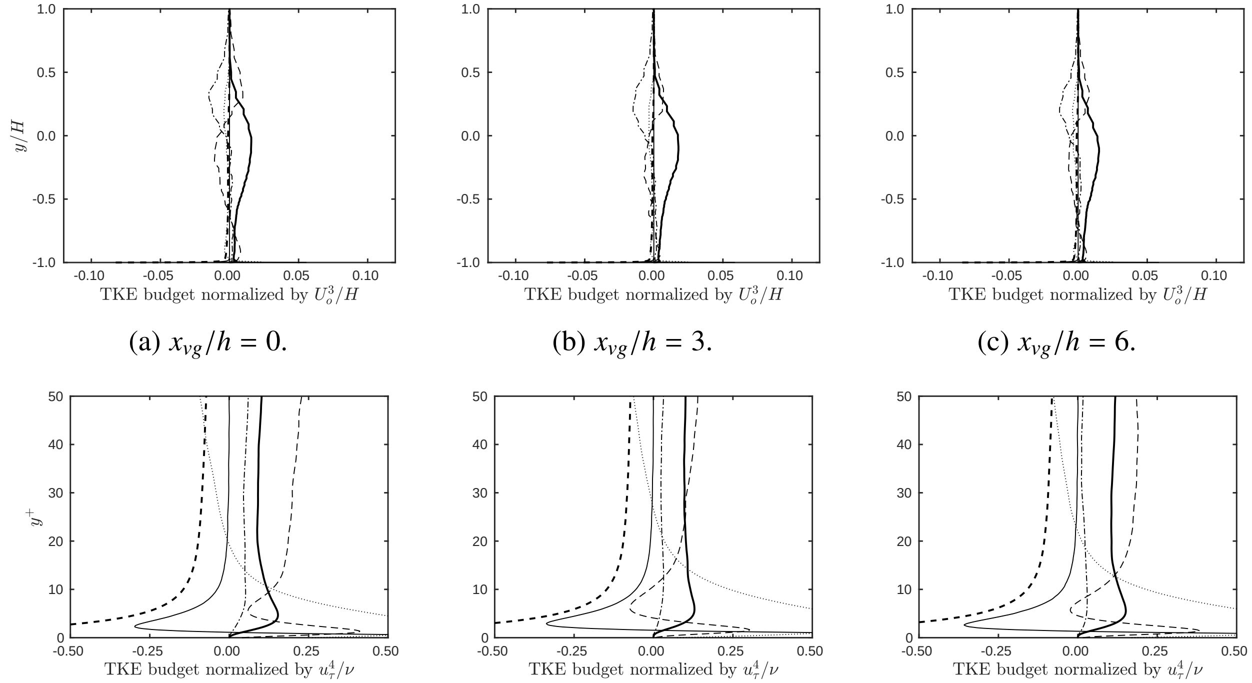
(c) $x_{vg}/h = 6$.

Figure 2.29: Normalized span-averaged TKE contributions over the ramp at $x/H = 0.5$ for different cube positions: −·−·, convection; ——, production; − − −, transport; ——, diffusion; - - - -, dissipation; · · · · · ·, velocity-pressure gradient.

and 6 has contributions from $\overline{u'v'}U_{x,y}$ and $\overline{u'w'}U_{x,z}$ to TKE production, consistent with the behavior observed in section 2.3.2. However, the more dispersed counter-rotating vortex pair for $x_{vg}/h = 6$ generates more turbulent fluctuations resulting in a greater magnitude of TKE production near the ramp edge. Positive convection and the negative TKE transport in all the cases indicates that the energy is transported downstream towards the plane of symmetry and near the wall.

Figure 2.29 shows the transfer of TKE over the ramp in the expansion section at $x/H = 0.5$. The counter-rotating vortex pair produced by the cube interacts with the separated flow and generated turbulent fluctuations, which leads to production of TKE. For the cube placed at $x_{vg}/h = 0$, the interaction of the turbulent structures generated around the cube with the separated shear layer contributes to the production of TKE in the region $0 < y/H < 0.5$. However, for the more spread and dispersed counter-rotating vortex pair in the case with $x_{vg}/h = 6$, a larger region of flow over the ramp is under the influence of the counter-rotating flow, which explains the slightly larger peak in TKE production. Similar to the baseline case the flow over the ramp in all the cases has negligible dissipation and the peak in TKE production corresponds to the negative peak in turbulent transport, thus indicating transfer of energy from the freestream towards the bottom wall.

Figures 2.30 and 2.31 show the TKE contributions near the reattachment location at $x/H = 6$ and near the end of the domain at $x/H = 15$, where the TKE production, transport, convection and diffusion is of similar order in both the outer-flow and inner-wall regions for the three cube



(a) $x_{vg}/h = 0$.

(b) $x_{vg}/h = 3$.

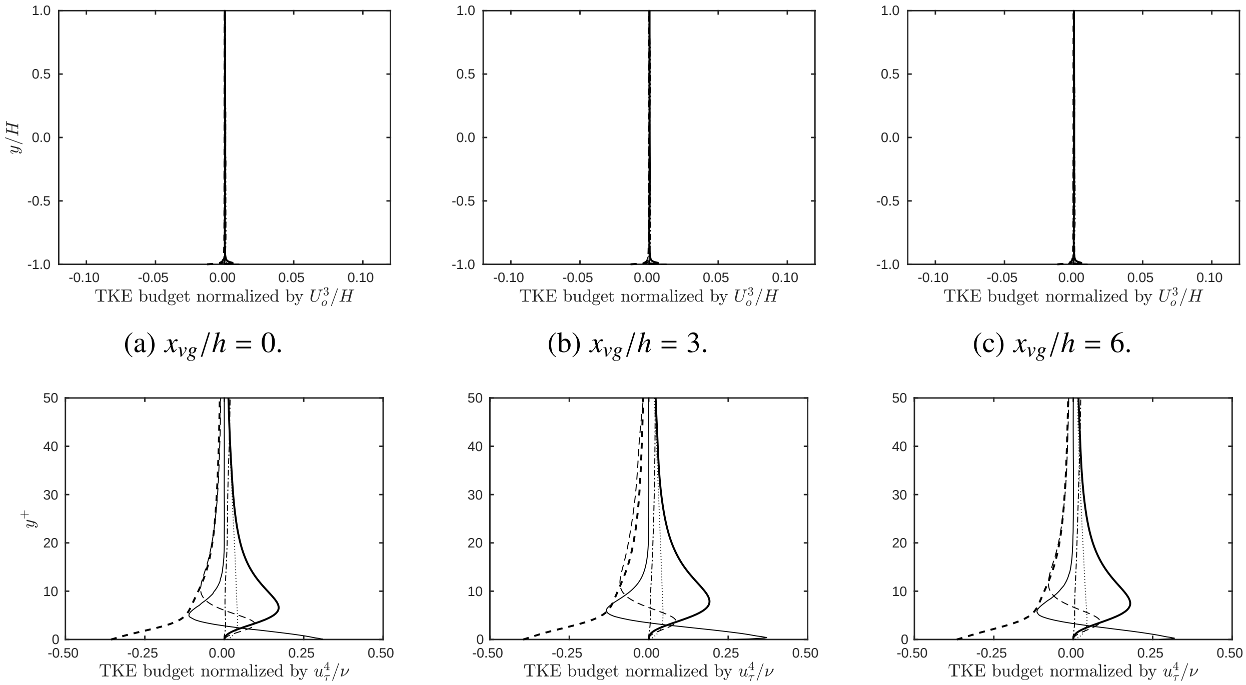
(c) $x_{vg}/h = 6$.

(d) $x_{vg}/h = 0$.

(e) $x_{vg}/h = 3$.

(f) $x_{vg}/h = 6$.

Figure 2.30: Normalized span-averaged TKE contributions at $x/H = 6$ for different cube positions: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\cdots$, velocity-pressure gradient.



(a) $x_{vg}/h = 0$.

(b) $x_{vg}/h = 3$.

(c) $x_{vg}/h = 6$.

(d) $x_{vg}/h = 0$.

(e) $x_{vg}/h = 3$.

(f) $x_{vg}/h = 6$.

Figure 2.31: Normalized span-averaged TKE contributions at $x/H = 15$ for different cube positions: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\cdots$, velocity-pressure gradient.

positions. Similar energy transfer for all cube positions demonstrates that beyond $x/H \geq 6$, the flow behavior is independent of the proximity of the cube to the region of adverse pressure gradient. Similar to the observations in section 2.3.2, it follows that in the attached TBL beyond $x/H \geq 6$ the velocity gradients near the bottom wall continue to contribute to the production of TKE. In addition, the TBL is in a state of non-equilibrium turbulence as the TKE production is not balanced by its dissipation, and does not recover by the end of the domain for all the cube positions.

2.5 Conclusions

Wall-resolved large-eddy simulations are conducted at a Reynolds number of 19,600, based on the inlet boundary layer thickness and freestream velocity, to study flow over a backward-facing ramp modulated by a submerged, wall-mounted cube used as a canonical vortex generator (VG). In particular, the turbulent transport that results in the modulation of the separated flow over the ramp is investigated by varying the size and location of the VG, which in turn is shown to modify the interactions between the VG-induced flow structures and those of the separated region. Numerical results indicate that the horseshoe vortex formed upstream of the wall-mounted cube interacts with the hairpin vortices in the separated region, stretching, turning, and entraining them, which results in a counter-rotating vortex pair near the plane of symmetry. Analysis of the spanwise variation of Reynolds stresses in the inlet and expansion section indicates the transfer of momentum towards the plane of symmetry and the near-wall region, which supports the claim that flow modulation by the wall-mounted cube is localized near the plane of symmetry. Moreover, these localized regions of high vorticity are associated with the production of turbulent kinetic energy, thus illustrating an effective mechanism of energy transfer from the freestream to the near-wall regions.

Modulation of the separated flow depends on the relative intensity of the interaction between the horseshoe vortex system and the shear layer, which depend on the size and location of the cube. Since the horseshoe vortex scales with the height of the cube, larger cubes induce a stronger counter-rotating flow with more effective transport of momentum over wider region around the plane of symmetry of the ramp. However, the drag originating from the skin friction forces and pressure forces increases with cube height. On the other hand, the upstream position of the cube changes the behavior of the counter-rotating flow induced by the horseshoe vortex in the inlet section. When the cube is placed far upstream, the strength of the dispersed counter-rotating vortex pair is too low to modulate the separated flow. If the cube is too close to the leading edge, the small-scale turbulent structures formed are entrained by the larger vortical structures in the separation region before fully developing, thus failing to produce a counter-rotating flow of sufficient strength to effectively reduce the size of the separated region. The optimal configuration in the present study is a cube with $h/\delta_0 = 0.6$ and $x_{vg}/h = 3$, as evidenced by the significant reductions in the separation length and the volume of separation over the ramp, while maintaining low coefficient of drag.

The present study is limited to a single cube. The VG geometry may affect the horseshoe vortex and thus affect the flow separation region in different ways. Furthermore, in practice, interactions of flow structures produced by multiple VGs in different arrangements would likely also affect the separated region. Future studies of these phenomena would improve our understanding of separation modulation using passive vortex generators.

CHAPTER 3

Modulation of Flow Over a Backward-Facing Ramp by an Array of Wall-Mounted Cubes

This chapter is adapter from Tandon *et al.* (2020*b*). The efficacy of passive flow control devices, such as vortex generators (VGs), in reducing the size of the separated region has been demonstrated in numerous studies. However, the interactions between the salient flow structures produced by the VGs and those of the separated flow are not fully understood. In this article, wall-resolved large-eddy simulations are conducted at a Reynolds number of 19,600 based on the inlet boundary layer thickness and freestream velocity, to study flow over a backward-facing ramp modulated by an array of equally-spaced, submerged, wall-mounted cubes. In particular, the turbulent transport that results in the modulation of the separated flow over the ramp is investigated by varying the spanwise spacing between the neighboring cubes, which in turn is expected to modify the interactions between the VG-induced flow structures and those of the separated region. The counter-rotating vortex pairs produced by the VGs entrain the freestream turbulent flow towards the near-wall region. These localized regions of high vorticity correspond to turbulent kinetic energy production regions, which effectively transfer energy from the freestream to the near-wall regions. The size of the vortex pairs depends on the height of the cubes, and thus for a given cube height and upstream position, the spanwise spacing between the cubes affect the behavior of the counter-rotating vortices—if the spacing is low, the counter-rotating vortex pairs are not sufficiently strong to affect the large-scale structures of the separated region, and if the spacing is too large, the flow modulation is similar to that of an isolated VG.

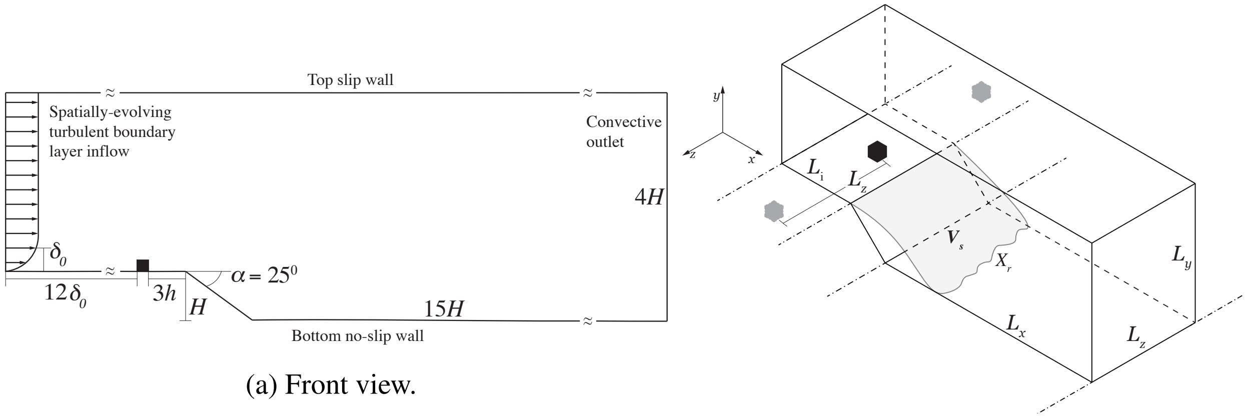
3.1 Introduction

Separation of spatially evolving turbulent boundary layer (TBL) flow near regions of adverse pressure gradient occurs in a variety of flow systems (internal and external). The separated flow, characterized by recirculating flow, may give rise to undesirable effects—reducing lift on wings, increasing drag on ground vehicles, and reducing efficiency in chemical mixing chambers. Given their fixed point of separation, backward-facing steps or ramps (Westphal *et al.*, 1984; Kaiktsis *et al.*, 1991; Le *et al.*, 1997; Herbst *et al.*, 2007; Kourta *et al.*, 2015) are commonly considered to study flow separation. Control of flow separation could be exploited to reduce the losses described above.

Brown *et al.* (1968), Calarese *et al.* (1985), Englar (2001), Logdberg (2006), Mohan *et al.* (2013), Wilson *et al.* (2019), and Fisher *et al.* (2020), among others, explored passive control techniques such as vortex generators (VGs) to reduce drag and improve performance in a variety of engineering applications. VGs are effective when the point of separation is fixed spatially (Rao & Kariya, 1988). Experimental investigation of VGs with different shapes and configurations by Lin (2002) on a backward-facing, curved ramp demonstrated that the submerged VGs of height 65% below the boundary layer thickness generates streamwise vortices that enhance turbulent mixing, and was substantiated in a parallel study by Jenkins *et al.* (2002). Flat-plate boundary layer flow over submerged VGs of different shapes has also been investigated in the past (Ashill *et al.*, 2001; Yao *et al.*, 2002; Elbing *et al.*, 2013; Iyer & Mahesh, 2013). However, it is difficult to generalize findings across the different shapes (e.g., Wishbone, Doublet Wheeler (Lin *et al.*, 1991)) as the flow features appear to be problem dependent. For this reason, a cubic VG has been considered as a canonical geometry in previous studies of Shinde *et al.* (2017), Shinde (2018) and in Chapter 2.

Modulation of a spatially evolving TBL flow over a backward-facing ramp by a single submerged, wall-mounted cube (Chapter 2) showed that the horseshoe vortex system (HVS) of the cube produces a counter-rotating vortex pair that stretches, turns and entrains the large hairpin structures in the separated region around itself and reduces flow separation near the plane of symmetry. Furthermore, the localized flow modulation by a single cube is dependent on the cube size and its proximity to the leading ramp edge. To increase the efficacy of flow modulation multiple VGs have also been used in the past (Ashill *et al.*, 2001; Jenkins *et al.*, 2002; Pujals *et al.*, 2010). Shinde (2018) reported that for a spatially evolving TBL flow over an array of equally-spaced, submerged wall-mounted cubes of height h with spacing $3h$ on a flat-plate, the interaction between neighboring HVS results in the wall-normal ejection of low-momentum flow, which lead to the amplification of large-scale structures and increase in TKE production in the near-wake of the cubes. However, for a larger spacing of $7h$, the flow behavior is similar to that of the single cube case. The wall-normal ejection results in secondary turbulent flows (Barros & Christensen, 2014; Vanderwel & Ganapathisubramani, 2015; Yang & Anderson, 2018), which depend on the spanwise spacing of VGs and alters the flow dynamics in the outer region, such that for small spanwise spacing, high-momentum pathway are observed above the wall-mounted cubes that decreases the drag coefficient (Yang *et al.*, 2019).

The objective of this work is to understand the role of turbulent transport in the modulation of flow separation using an array of equally-spaced cubic VGs. In particular, wall-resolved large-eddy simulations (LES) are used to investigate the dependence of the flow in the separated region on the spanwise spacing of the VGs. Modulation of the separated flow over the ramp is expected to depend on the interaction of the large-scale structures in the separated region with the turbulent structures that are formed due to the interaction between the HVS of neighboring cubes, which is dictated by the spanwise spacing between the cubes in an array. The problem set-up and the numerical methods are described in section 3.2. Section 3.3 examines turbulent flow modulation and transport for an array of wall-mounted cubes and studies the dependence of the flow on the spanwise spacing between the cubes. The article ends with concluding remarks in section 3.4.



(a) Front view.

(b) 3-D computational domain.

Figure 3.1: Backward-facing ramp flow configuration.

3.2 Problem Description

A schematic of spatially evolving turbulent boundary layer flow over a backward-facing ramp with an array of equally-spaced, submerged, wall-mounted cubes present near the leading ramp edge is shown in figure 3.1. A similar computational geometry and domain is considered as in Chapter 2, so that the fluid dynamics of a line array of VGs can be compared to those corresponding to a single VG. The 3-D flow domain consists of an inlet section, followed by a ramp of height H at a fixed inclination of $\phi = 25^o$ with the horizontal, and an expansion section. A line array of wall-mounted cubes of a fixed height h and separated by a spanwise distance L_z, is located at a distance of x_{vg} from the ramp edge. A turbulent boundary layer flow of thickness δ_0 and free-stream velocity U_0 is prescribed at the inlet, with Reynolds number defined as $Re = U_0\delta_0/\nu = 19,600$. The inlet length is $L_i = 12\delta_0 + h + x_{vg}$, length of the downstream expansion region is $L_x = 15H$ and the domain height in the expansion region is $L_y = 4H$.

Similar to the previous study (Chapter 2), wall-resolved large-eddy simulations (LES) are conducted by solving the filtered Navier-Stokes equations,

$$\frac{\partial}{\partial x_k}\widetilde{u}_k = 0\,, \tag{3.1}$$

$$\frac{\partial}{\partial t}\widetilde{u}_i + \frac{\partial}{\partial x_k}\widetilde{u}_i\widetilde{u}_k - \nu\frac{\partial^2}{\partial x_k^2}\widetilde{u}_i + \frac{\partial}{\partial x_k}\tau_{ik}^R + \frac{1}{\rho}\frac{\partial}{\partial x_i}\widetilde{p} = 0, \tag{3.2}$$

where $\widetilde{u}$ is the filtered velocity, $\widetilde{p}$ is the filtered pressure, ν is the kinematic viscosity, ρ is the density, and $\tau_{ik}^R \equiv \widetilde{u_i u_k} - \widetilde{u}_i\widetilde{u}_k$ is the subgrid-scale stress that requires modeling. The indices i, j, k denote the streamwise (x), wall-normal (y), and spanwise (z) directions, respectively.

A second-order accurate finite volume approach with implicit time marching is used in the OpenFOAM framework (Weller *et al.*, 1998). The equations are solved based on the PIMPLE algorithm—a combination of of PISO (Pressure Implicit with Splitting of Operator, Issa, 1986), and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations, Caretto *et al.*, 1973) algorithm—with two outer corrector steps. To adequately capture the back-scatter of the turbulent kinetic energy observed in the region upstream of the cube (Krajnovic & Davidson, 1999), the dynamic k-equation eddy-viscosity LES model of Kim & Menon (1995) is used. A maximum Courant number of 1.0 enables variable time step size. The field quantities and the turbulent statistics are averaged for over 60 flow-throughs starting from $t \approx 750H/U_0$ to avoid contamination by initial transients.

At the inlet, a time-varying velocity field is prescribed using a synthetic inflow method (Shinde *et al.*, 2017) for a spatially evolving turbulent boundary layer (TBL) with a free-stream turbulence intensity of 1%. The synthetic inflow requires a transition length of approximately $12\delta_0$ to develop into a realistic TBL. The bottom wall of the inlet section, the exposed faces of the cube, the ramp surface, and the bottom wall of the expansion region have no-slip boundary conditions. A no-stress slip wall is applied at the upper boundary of the domain, and at the outlet, a convective boundary condition is prescribed (Lowery & Reynolds, 1986). A line array of wall-mounted cubes is represented by a single cube in the inlet section, with periodic boundaries enforced along the lateral sidewalls.

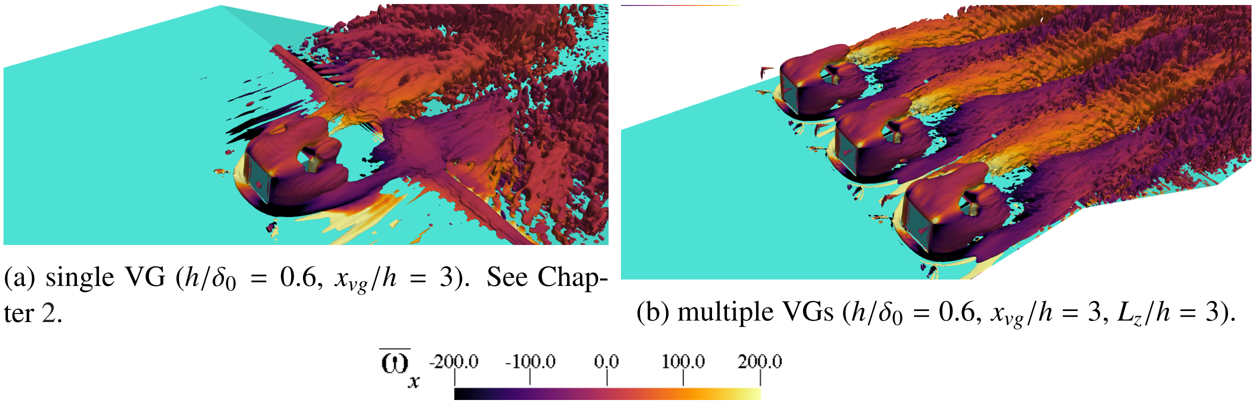
A non-uniform structured grid is employed, where high spatial resolution is achieved through local mesh refinement. The region between $3H$ upstream of the leading edge and $6H$ downstream in the expansion region has a uniform resolution in the spanwise and streamwise flow directions with grid spacing in wall units given as $\Delta_x^+ \approx \Delta_z^+ \approx 20$. To provide high fidelity in the reattachment region, the near-wall region is resolved with $\Delta_y^+ \approx 1$, such that no wall-model is necessary.

To understand the dependence of the interaction of multiple horseshoe vortex systems (HVS) with the shear layer on the turbulent transport, different spanwise spacings between the neighboring cubes are considered and compared to the flow with a single VG (Chapter 2), which suggests the optimal height and upstream position as $h/\delta_0 = 0.6$ and $x_{vg}/h = 3$ respectively. It is expected that at small spacings the HSV of neighboring cubes strongly interact, though perhaps not optimally; as the spacing is increased, an ”optimal” configuration is achieved such that the HVSs exhibit strongest interactions. Beyond that spacing, the interaction are likely less intense, up a critical spacing beyond which the HSVs no longer interact in the separated region. For the fixed height and position of the cube, the following spanwise spacings between the neighboring cubes in a line array are considered: $L_z/h = 3$, 5 and 7. Shinde (2018) varied the inter-cube spacing between $3h$ and $7h$ for TBL flow over an array of equally-spaced cubes on a flat plate and reported that with spacing $3h$ the interaction between the adjacent VG-induced flow structures results in the wall-normal ejection of low-momentum flow in the near-wake of the cubes with amplification of large-scales in the outer flow regions. However, for spacing $7h$, the flow dynamics reached the limit of a single cube case.

3.3 Dependence of the flow modulation over a backward-facing ramp on the spanwise spacing between the wall-mounted cubes

3.3.1 Velocity and vorticity fields

The dependence of the flow modulation in the separated region on the spanwise spacing between cubes is examined by considering array of cubes with fixed location $x_{vg}/h = 3$ and height $h/\delta_0 = 0.6$, and varying the spanwise spacing $L_z/h = 3$, 5, and 7. To provide an understanding of the



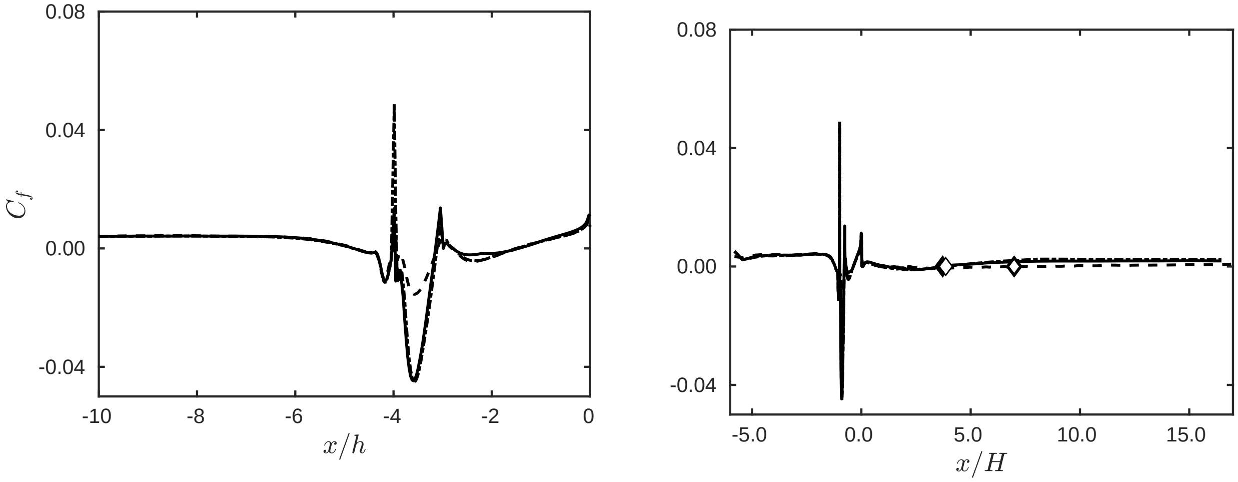
(a) single VG (h/δ_0 = 0.6, x_{vg}/h = 3). See Chapter 2.

(b) multiple VGs (h/δ_0 = 0.6, x_{vg}/h = 3, L_z/h = 3).

Figure 3.2: Iso-surfaces of the $Q-$criterion colored with the time-averaged streamwise vorticity ($\overline{\omega}_x$).

impact of multiple vortex generators (VGs) on separated flow over a ramp, the flow with $L_z/h = 3$ is first qualitatively compared to that with single VG baseline case ($h/\delta_0 = 0.6$, $x_{vg}/h = 3$) of (Chapter 2). Figure 3.2 shows the iso-surfaces of the Q-criterion (Hunt *et al.*, 1988) to illustrate the vortical structures. When there is a single VG present, the horseshoe vortex system (HVS, Martinuzzi & Tropea, 1993) produced by the cube generates a counter-rotating vortex pair in the near-wake of the cube that interacts with the large-scale structures in the separated region over the ramp (Song *et al.*, 2000; Kourta *et al.*, 2015) and entrains flow towards the plane of symmetry. In the presence of multiple VGs, identical HVS are formed around each cube, which interacts with the HVS of the neighboring cube and forms similar counter-rotating vortex pairs in the near-wake of each cube at the leading ramp edge. Multiple counter-rotating vortex pairs interact and entrain the separated flow, which results in flow modulation of a larger region over the ramp in contrast to the single VG flow. The flow behavior described in figure 3.2 suggests that the interaction of the HVS between neighboring cubes depends on the spanwise spacing between cubes, which is expected to influence the modulation of the separated flow over the ramp.

The flow separation X_s^{vg} and reattachment X_r^{vg} around the cubes (Escauriaza & Sotiropoulos, 2011; Krajnovic & Davidson, 2002) is obtained from the skin friction coefficient $C_f = \tau_w/\frac{1}{2}\rho U_\infty^2$, where $\tau_w = \mu\left(\partial u/\partial y\right)_{y=0}$ is the shear stress. Figure 3.3 shows C_f along the streamwise direction for the different spanwise spacing between the cubes, evaluated based on the time-averaged velocity.



(a) Flow separation and reattachment around the cubes in inlet section.

(b) Flow reattachment in the expansion section.

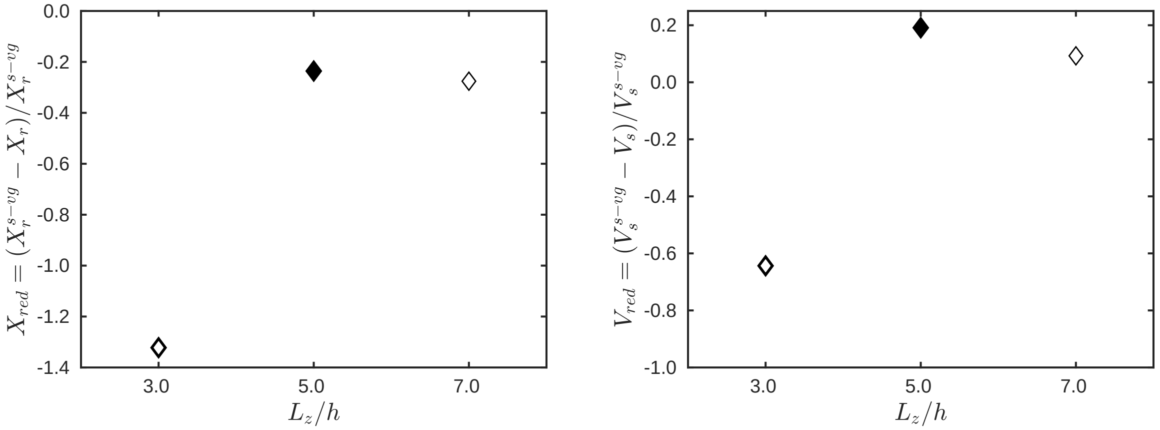
Figure 3.3: Streamwise variation of the skin friction coefficient (C_f) along the plane of symmetry for spanwise spacing: - - - -, $L_z/h = 3$; ——, $L_z/h = 5$; -·-·-, $L_z/h = 7$.

h/δ_0	x_{vg}/h	L_z/h	X_s^{vg}/h	X_r^{vg}/h
0.6	3.0	3.0	0.78	1.54
0.6	3.0	5.0	0.82	1.58
0.6	3.0	7.0	0.82	1.54

Table 3.1: Separation (X_s^{vg}) and reattachment (X_r^{vg}) lengths for the cubes along the plane of symmetry ($z = 0$).

In the inlet section, the point of separation is the first point where $C_f < 0$ near the upstream face of the cubes, and the point of reattachment is the first point when $C_f > 0$ in the near-wake of the VGs. Table 3.1 lists the values of X_s^{vg} and X_r^{vg} for the spanwise spacing of the cubes under consideration. Similar values are obtained for the separation and reattachment points for different spanwise spacing, which indicates that for the configuration of cube arrays considered in this study, the flow around the multiple VGs is primarily influenced by the cube height and has little dependence on the spanwise spacing between the neighboring cubes. Yang *et al.* (2019) evaluated drag forces on sparsely packed cube arrays and found that when the spacing between the cubes decreases, secondary turbulent flow manifests due to the interaction of flow structures of neighboring cubes and reduces the drag forces on the cubes, which explains the low peak in C_f for $L_z/h = 3$ as opposed to the flow with $L_z/h = 7$. In the expansion section, the mean reattachment location $\overline{X}_r$ is the first point where $C_f > 0$ after the turbulent boundary layer (TBL) flow separates at the leading edge. As the spanwise spacing between the cubes decreases the separation length increases, which suggests that the size of the separated region depends on the interaction between the HSV and the shear layer, and thus on the spanwise spacing between the VGs. Figure 3.4 shows the reduction in the size of the separation region relative to that of the flow with the single VG baseline case (Chapter 2). Negative values in figure 3.4 indicates that multiple VGs with $L_z/h = 3$ significantly increase the size of the separated region. While for higher L_z, the length of the separated region is approximately 20% lower than the single VG case, the volume of the separated region is reduced as much as 20% for $L_z/h = 5$. In addition, the coefficient of drag C_d is evaluated by adding the contributions from the skin friction on the exposed faces of the cubes and the form drag due to the separation of flow around the cubes. Table 3.2 lists the values of the mean reattachment location $\overline{X}_r$ and C_d for the different cases. As explained before, the reduction in spanwise spacing between VGs reduces the drag, which explains the low coefficient of drag of $C_d = 0.58$ for $L_z/h = 3$ case. Of the spanwise spacing considered in the present study, the $L_z/h = 5$ produces the greatest reduction in separation volume over the ramp, while maintaining a lower coefficient of drag.

The effect of varying the spanwise spacing between the VGs on the flow modulation observed



(a) Reduction of the separation length.

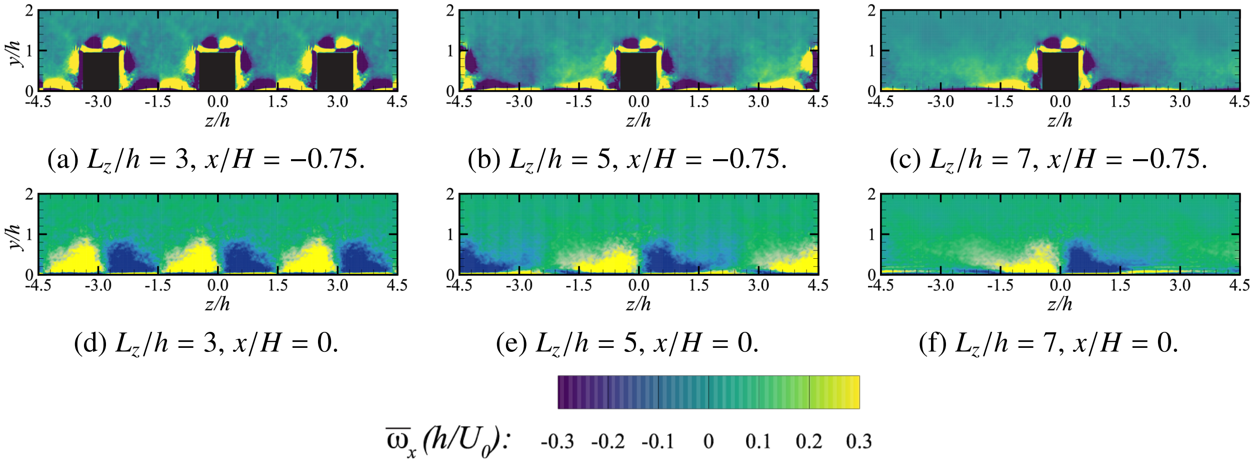
(b) Reduction of the separation volume.

Figure 3.4: Reduction in the size of the separation region for the spanwise spacings: ◇, $L_z/h = 3$; ◆, $L_z/h = 5$; ◇, $L_z/h = 6$.

h/δ_0	x_{vg}/h	L_z/h	$\overline{X}_r/H$	V_s/H^3	C_d
0.6	3.0	–	3.01	2.45	0.80
0.6	3.0	3	6.99	4.02	0.64
0.6	3.0	5	3.72	1.96	0.78
0.6	3.0	7	3.84	2.21	0.82

Table 3.2: Mean reattachment location ($\overline{X}_r$) measured along the plane of symmetry ($z = 0$), the volume of separation (V_s) is evaluated near the ramp in a region $10H \times 1.25H \times 2H$ (L×W×H), and the coefficient of drag (C_d) for different spanwise spacings. The values in the first row correspond to the single VG baseline case of [TODO: cite JFM].

in figure 3.4 can be better understood by examining the formation and interaction of turbulent structures in the vorticity contours. Figure 3.5 shows the time-averaged streamwise vorticity contours in the inlet section for the different cases. At $x/H = -0.75$ near the downstream faces of the cubic VGs, the decrease in spanwise spacing increases the spanwise blockage, which results in the horseshoe vortex system (HVS) of individual cubes in $L_z/h = 3$ case to be spatially constrained near the cube surface as opposed to the larger L_z cases considered. The high vorticity centers extend in the near-wake of individual cubes to form multiple pairs of counter-rotating vortex-pairs near the leading ramp edge at $x/H = 0$. The height of the counter-rotating vortex pairs is approximately $0.8h$ for all the cases, which suggests that the effective size of the counter-rotating vortex pairs depends only on the height of the array and has a negligible effect of the spanwise spacing; however, the lateral spreading of the counter-rotating vortices increases with L_z. Figure 3.6 shows the effect of the spanwise spacing between the cubes on the interaction between the separated region and the counter-rotating vortex pairs over the ramp surface for different cases. Half-way down the ramp at $x/H = 0.5$, the interaction between the separated region and the HVS results in the formation of enlarged counter-rotating vortex pairs (Chapter 2), which are of the order of the ramp height. As before, the spreading of the vortices increases with an increase in the spanwise spacing. Furthermore, localized regions of high vorticity are observed in outer flow regions between $0 < y/H < 0.5$ for $L_z/h = 3$ and 5, a manifestation of turbulent interactions between the neighboring HVS. For $L_z/h = 7$, the large spanwise spacing results in negligible interaction between the neighboring VG-induced turbulent structures, and the flow behavior is similar to that of an isolated single VG. Near the downstream ramp edge at $x/H \approx 2$, the compact counter-rotating vortex pairs for $L_z/h = 3$ case succumb to the large velocity gradients and strain in the expansion section over the ramp, which causes dispersion and decay of turbulent structures and ultimately separation of flow. In the case of larger spanwise spacing with $L_z/h = 5$ and 7, while the turbulent structures disperse due to large strain in the expansion section, the continuous entrainment of flow by the counter-rotating vortex pairs contributes to the increase in their size, which facilitates the reduction of the size of the separated region over the ramp.



(a) $L_z/h = 3$, $x/H = -0.75$.

(b) $L_z/h = 5$, $x/H = -0.75$.

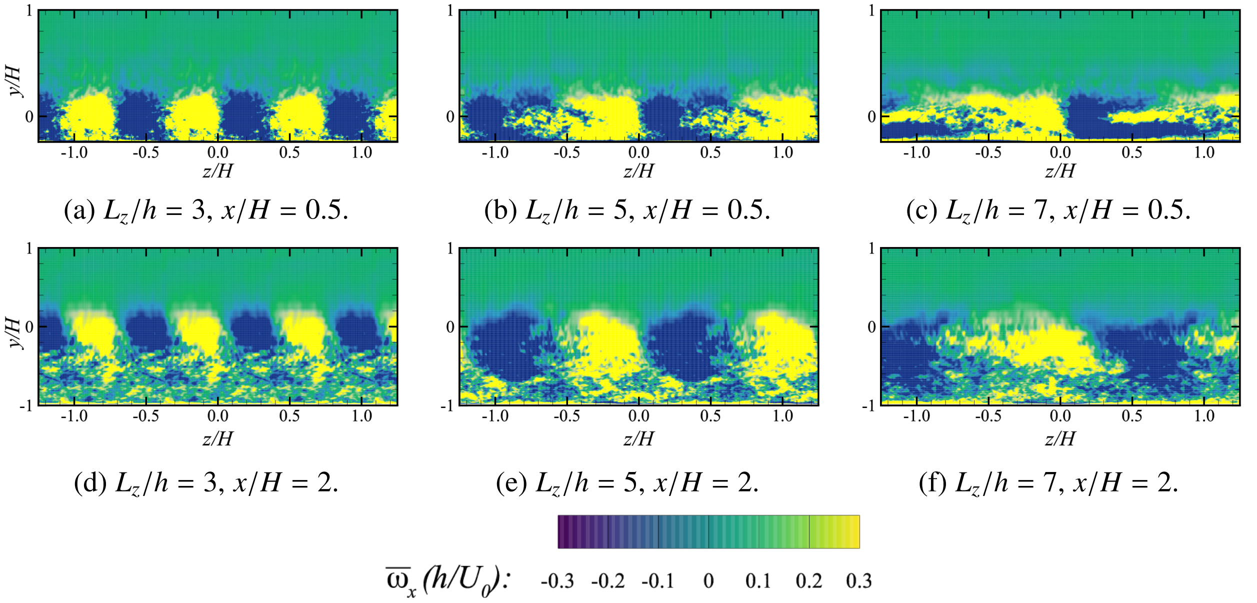
(c) $L_z/h = 7$, $x/H = -0.75$.

(d) $L_z/h = 3$, $x/H = 0$.

(e) $L_z/h = 5$, $x/H = 0$.

(f) $L_z/h = 7$, $x/H = 0$.

Figure 3.5: Contours of the streamwise component of the time-averaged vorticity ($\overline{\omega}_x$) in the inlet section near the downstream face of the cubic VGs at $x/H = -0.75$ and along the leading ramp edge at $x/H = 0$ for the different spanwise spacings. The black face in (a)-(c) depicts the downstream face of cubes. The streamwise flow direction ($+x$) is directed into the paper and the inlet section is normalized by h.



(a) $L_z/h = 3$, $x/H = 0.5$.

(b) $L_z/h = 5$, $x/H = 0.5$.

(c) $L_z/h = 7$, $x/H = 0.5$.

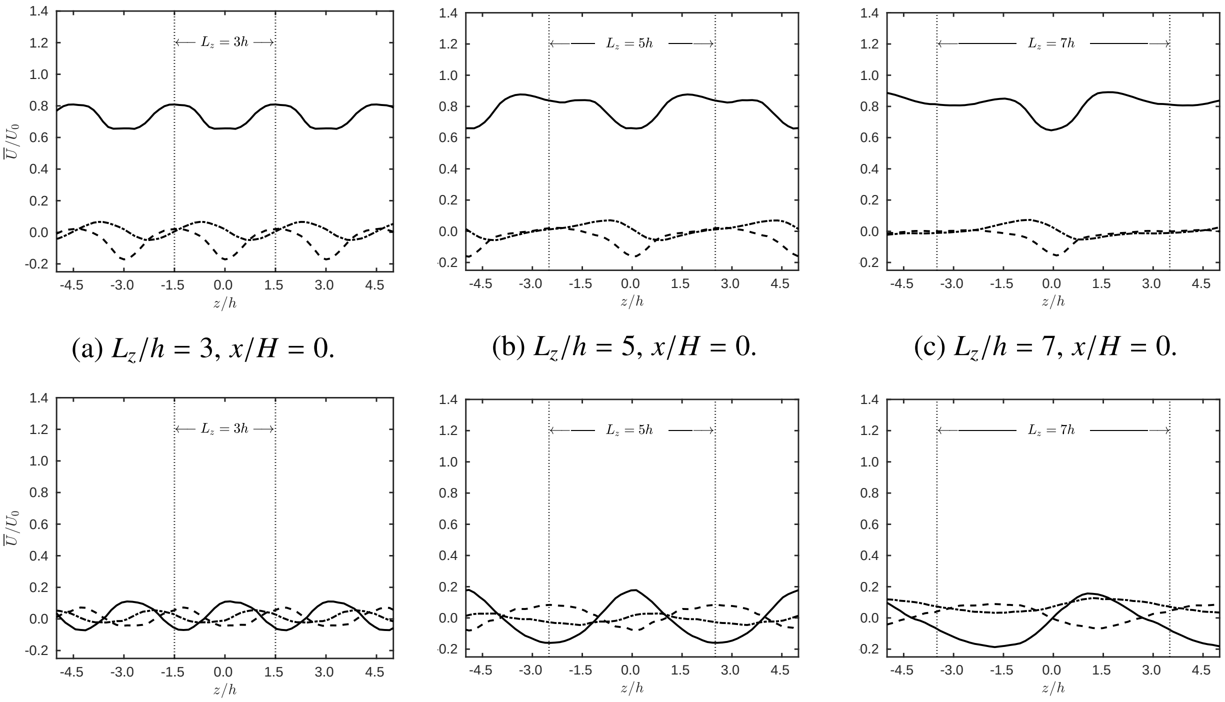
(d) $L_z/h = 3$, $x/H = 2$.

(e) $L_z/h = 5$, $x/H = 2$.

(f) $L_z/h = 7$, $x/H = 2$.

Figure 3.6: Contours of the streamwise component of the time-averaged vorticity ($\overline{\omega}_x$) over the ramp at $x/H = 0.5$ and near the downstream ramp edge at $x/H = 2$ for the different spanwise spacings. The streamwise flow direction ($+x$) is directed into the paper and the expansion section is normalized by H.

The effect of spanwise spacing between the cubes on the entrainment of freestream fluid towards the near-wall region can be more precisely elucidated by examining the spanwise variation of time-averaged velocity $\overline{U}$ in the inlet and expansion section for the three cases. Figure 3.7 shows the mean velocity components in the spanwise direction near the ramp edge at $x/H = 0$ and over the ramp at $x/H = 1$. The TBL flow in the inlet is attached in all the cases, where the streamwise component $\overline{U}_x$ is an order of magnitude greater than the other components. In the presence of multiple VGs, the flow is spanwise-periodic and results in the formation of multiple counter-rotating vortex pairs. In each counter-rotating vortex pair a 30% decrease in $\overline{U}_x$ in its plane of symmetry is observed, while $\overline{U}_y$ becomes negative and $\overline{U}_z$ exhibits a sine-wave like variation (Pujals *et al.*, 2010), with a wavelength of L_z. Negative $\overline{U}_y$ indicates that the flow is directed towards the bottom wall, and positive $\overline{U}_z$ in the negative z half-domain implies flow towards the plane of symmetry. Such behavior is characteristic of a counter-rotating vortex pair (Angele & Muhammad-Klingmann, 2005), where the entrainment of flow from the sides and the outer-flow regions towards the near-wall regions leads to reattachment. For lower spanwise spacings $L_z/h = 3$ and 5, slightly positive $\overline{U}_y$ near regions of interaction of neighboring counter-rotating vortex pairs suggests wall-normal ejection of low momentum fluid, which produces secondary turbulent flows (Barros & Christensen, 2014; Vanderwel & Ganapathisubramani, 2015; Yang & Anderson, 2018) and alters the outer flow dynamics. However, for $L_z/h = 7$, secondary flow regions are not observed due to negligible interaction between the neighboring vortex pairs. Over the ramp at $x/H = 1$, the interaction between the counter-rotating vortex pairs and the separated region leads to modulation of flow such that in the plane of symmetry of individual vortex pairs, the attached flow has positive $\overline{U}_x$ and negative $\overline{U}_y$. For $L_z/h = 3$, the width of the region of flow modulated by each counter-rotating vortex pair is about $1.8h$, which is 60% of its spanwise spacing, whereas for $L_z/h = 5$ and 7 the width of the modulated region is $2.4h$ ($\approx$ 45% of its L_z) and $2.8h$ ($\approx$ 40% of its L_z) respectively. Therefore, flow modulation by multiple VGs depends on the height of the cubes and the spanwise spacing between them. However, the lower magnitude of the peaks of $\overline{U}_x$ and $\overline{U}_y$ in $L_z/h = 3$ case suggests that the modulation of flow by the counter-rotating vortex pairs is weaker



(a) $L_z/h = 3$, $x/H = 0$.

(b) $L_z/h = 5$, $x/H = 0$.

(c) $L_z/h = 7$, $x/H = 0$.

(d) $L_z/h = 3$, $x/H = 1$.

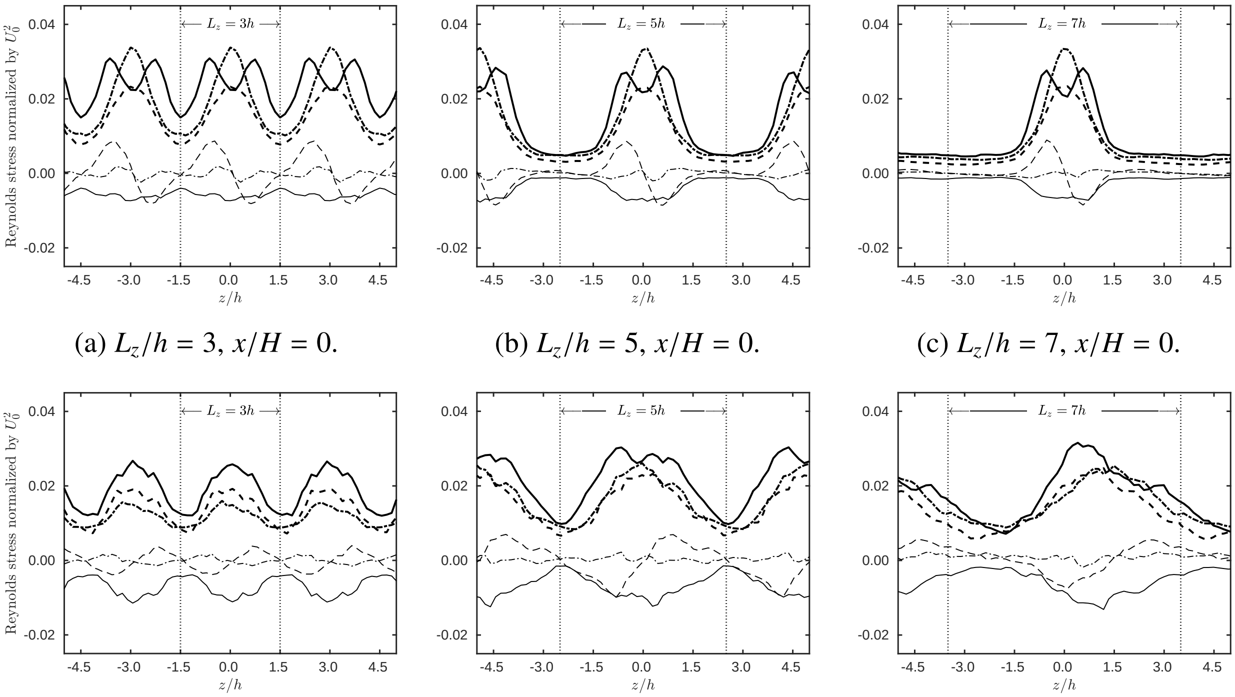
(e) $L_z/h = 5$, $x/H = 1$.

(f) $L_z/h = 7$, $x/H = 1$.

Figure 3.7: Spanwise variation of the time-averaged velocity field ($\overline{U}$) at $x/H = 0$ and $x/H = 1$, $y = 0.5h$: ——, $\overline{U}_x$; - - - -, $\overline{U}_y$; -·-·-, $\overline{U}_z$. The vertical dotted lines depict the spanwise width of the domain.

in comparison to that of $L_z/h = 5$ and 7.

The entrainment of flow by the counter-rotating vortex pairs enhances the momentum of the near-wall flow. The high-momentum flow structure (Bross *et al.*, 2019) affects the changes in viscous stresses and balances the pressure changes in the expansion section, which opposes flow separation. To understand the corresponding momentum transport, figure 3.8 shows the spanwise variation of the Reynolds stress components for the different cases in the inlet section and over the ramp. Near the leading ramp edge at $x/H = 0$, consistent with the single VG study, two positive peaks of $\overline{u'^2}$ in each counter-rotating vortex pair are observed, which corresponds to the centers of the vortices and indicates the transfer of momentum in the streamwise direction by the HVS. The peaks in $\overline{v'^2}$ and $\overline{w'^2}$ in the center of each vortex pair represents wall-normal and spanwise transport of momentum by the HVS to the near-wall region and towards the center plane of the vortex pair, respectively. As illustrated by the negative peak of $\overline{u'v'}$, the streamwise momentum ($u' \geq 0$) is



(a) $L_z/h = 3$, $x/H = 0$.

(b) $L_z/h = 5$, $x/H = 0$.

(c) $L_z/h = 7$, $x/H = 0$.

(d) $L_z/h = 3$, $x/H = 1$.

(e) $L_z/h = 5$, $x/H = 1$.

(f) $L_z/h = 7$, $x/H = 1$.

Figure 3.8: Spanwise variation of the Reynold stresses at $x/H = 0$ and $x/H = 1$, and $y = 0.5h$: ——, $\overline{u'^2}$; - - - -, $\overline{v'^2}$; -·-·-·, $\overline{w'^2}$; ——, $\overline{u'v'}$; − − −, $\overline{u'w'}$; −·−·, $\overline{v'w'}$. The vertical dotted lines depict the spanwise width of the domain.

transported by each vortex pair towards the bottom wall ($v' \leq 0$), *i.e.* from the outer region to the near-wall region. In the left half of the vortex pair, the streamwise momentum ($u' \geq 0$) is transported towards the right half ($w' \geq 0$), *i.e.* towards the plane of symmetry of each vortex pair, giving rise to the positive peak in $\overline{u'w'}$ in the left half. The interactions between the neighboring vortex pairs are more prominent for $L_z/h = 3$ case, where, the wall-normal ejection of flow creates small recirculation zones between the neighboring cubes and thus results in a larger $\overline{u'v'}$ value. Similar behavior is observed in the expansion region, with the difference being that the turbulent fluctuations in the expansion region are on the order of the ramp height. However, the lower peak of $\overline{u'^2}$, $\overline{v'^2}$ and $\overline{w'^2}$ for $L_z/h = 3$ case indicates that the momentum transport to the near-wall region is less effective as compared to that of $L_z/h = 5$ and 7.

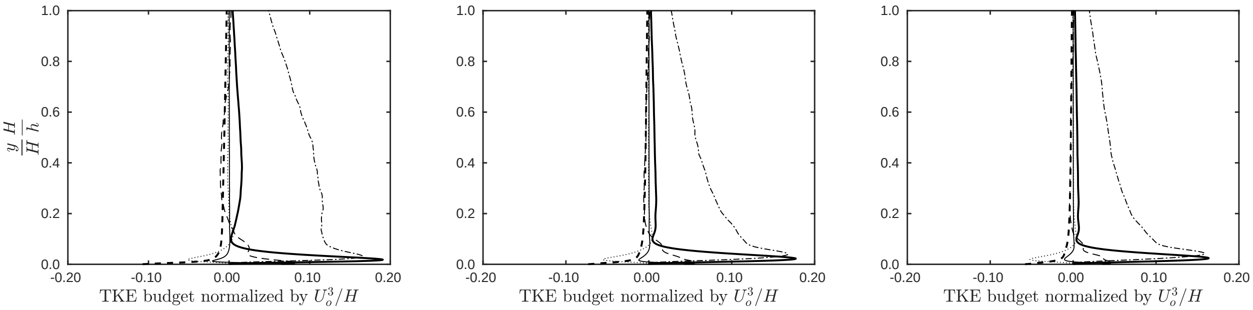
3.3.2 Turbulent kinetic energy transport to the near-wall region

The counter-rotating vortex pairs produced by the array of cubes draw fluid from the freestream and injects it into the near-wall region, thus energizing the turbulent boundary layer (TBL) flow. To better understand energy transfer by multiple VGs, the evolution of turbulent kinetic energy (TKE) is considered,

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{1}{2}q^2\right) &= \underbrace{-\frac{1}{2}U_j\overline{(u'_iu'_i)_{,j}}}_{C_k}\underbrace{-\overline{(u'_iu'_j)}U_{i,j}}_{P_k}\underbrace{-\frac{1}{2}\overline{(u'_iu'_iu'_j)_{,j}}}_{T_k}\\ &\quad\underbrace{+\frac{1}{2}\left(\frac{1}{Re}+\nu_\tau\right)\overline{(u'_iu'_i)_{,jj}}}_{D_k}\underbrace{-\frac{1}{2}\left(\frac{1}{Re}+\nu_\tau\right)\overline{(u'_{i,j}u'_{i,j})}}_{\epsilon_k}\underbrace{-\overline{u'_ip'_{,i}}}_{\Pi_k}\end{aligned} \tag{3.3}$$

where $q^2/2$ is the TKE, Re is the Reynolds number based on mean velocity U_0 and boundary layer thickness δ_0, and ν_τ is the eddy viscosity in the subgrid-scale model. The over-bar represents the time-averaging and the terms on the right hand side of equation 3.3 are convective (C_k), production (P_k), turbulence transport (T_k), viscous diffusion (D_k), viscous dissipation (ϵ_k) and velocity-pressure gradient (Π_k).

The evolution of TKE demonstrates the effect of spanwise spacing between the cubes on the



(a) $L_z/h = 3$.

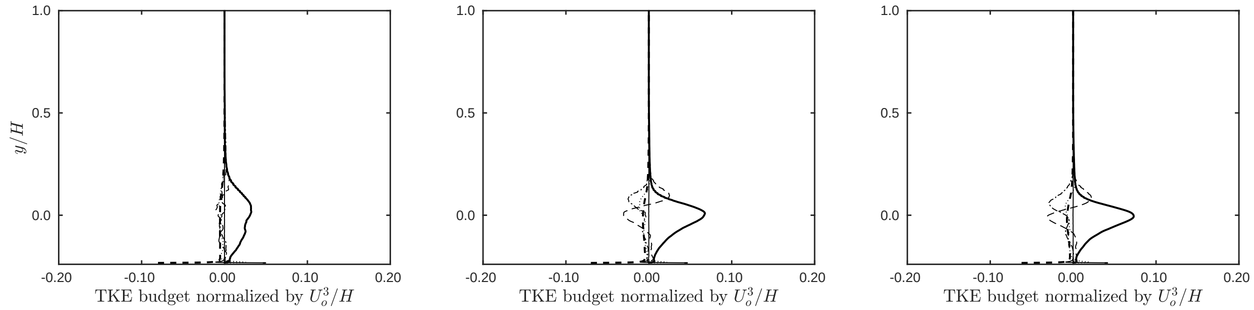
(b) $L_z/h = 5$.

(c) $L_z/h = 7$.

Figure 3.9: Normalized span-averaged TKE contributions near the leading ramp edge at $x/H = 0$ for different spanwise spacings: —·, convection; ——, production; − − −, transport; ——, diffusion; - - - -·, dissipation; · · · · · ·, velocity-pressure gradient.

production and transfer of TKE to the inner-wall region. Figure 3.9 shows the spanwise-averaged TKE contributions at the leading ramp edge at $x/H = 0$. The counter-rotating vortex pairs of cubes have contributions from $\overline{u'v'}U_{x,y}$ and $\overline{u'w'}U_{x,z}$ to TKE production, consistent with the behavior observed in single VG flow (Chapter 2). However, the lower spacing of $L_z/h = 3$ generates a larger number of compact counter-rotating vortex pairs for a given ramp width that generates more turbulent fluctuations resulting in a greater magnitude of TKE production near the ramp edge. In addition, wall-normal ejection of flow between neighboring vortex pairs for $L_z/h = 3$ and 5 results in the formation of secondary turbulent flows that contribute to TKE production beyond $y/h > 0.2$. The negative peak in TKE transport corresponds with the peak in TKE production in all the cases and indicates the transport of energy near the wall. The convection of TKE is defined in equation 3.3 in terms of outer-scale velocity U. The amplification of large-scale structures due to wall-normal ejection results in an increased magnitude of convection for the $L_z/h = 3$ case.

Figure 3.10 shows the transfer of TKE over the ramp in the expansion section at $x/H = 0.5$. The counter-rotating vortex pairs produced by multiple VGs interact with the separated flow and generates turbulent fluctuations, which leads to the production of TKE. For the spanwise spacing $L_z/h = 3$, the interaction of the secondary turbulent flow with the separated shear layer contributes to the production of TKE in the region $0 < y/H < 0.5$. However, for the larger counter-rotating vortex pairs in the case with $L_z/h = 5$ and 7, a larger region of flow over the ramp is under the



(a) $L_z/h = 3$.

(b) $L_z/h = 5$.

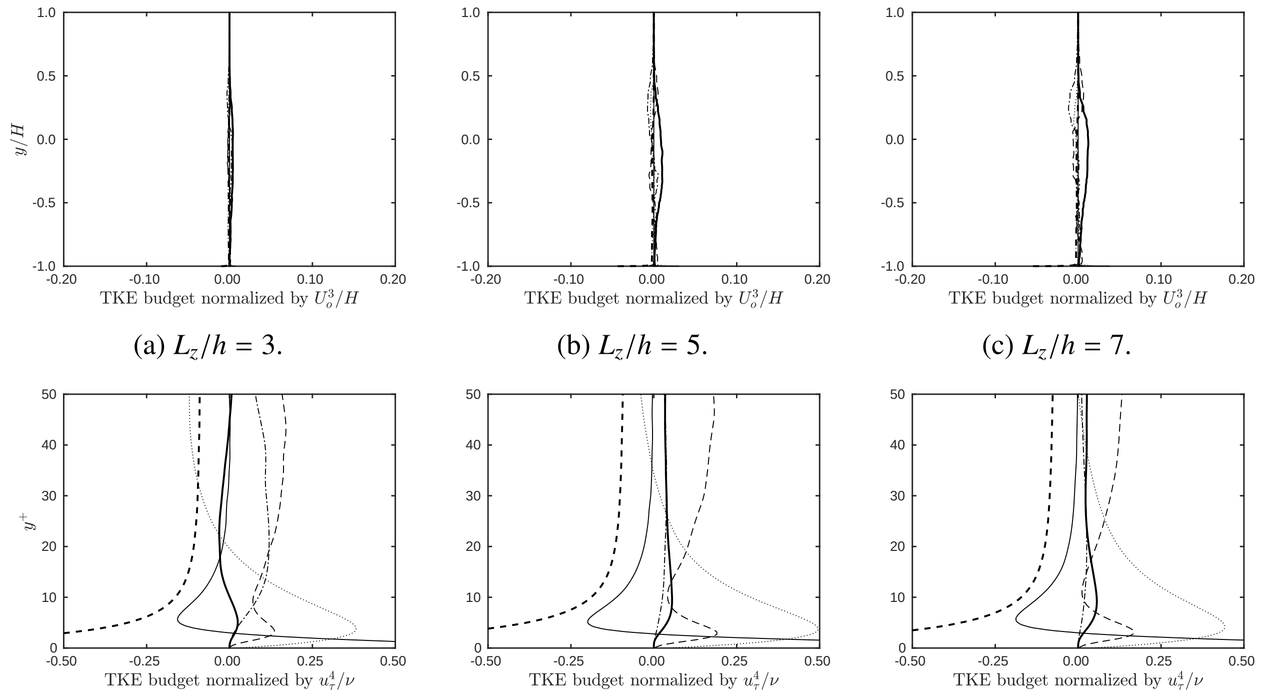
(c) $L_z/h = 7$.

Figure 3.10: Normalized span-averaged TKE contributions over the ramp at $x/H = 0.5$ for different spanwise spacings: $-\cdot-\cdot$, convection; ——, production; $- - -$, transport; ——, diffusion; - - - -, dissipation; $\cdots\ \cdots$, velocity-pressure gradient.

influence of the counter-rotating flow, which explains the larger peak in TKE production. Similar to the single VG study (Chapter 2), the flow over the ramp in all the cases has negligible dissipation, and the peak in TKE production corresponds to the negative peak in turbulent transport, thus indicating the transfer of energy from the freestream towards the bottom wall.

Further downstream in the expansion section, the TKE contributions are shown in figures 3.11 to 3.13 both in terms of outer and inner coordinates to illustrate the dependence of the large-scale structures on the outer variables (mean velocity U_0 and ramp height H) and the variation in near-wall behavior with respect to the inner variables (friction velocity u_τ and kinematic viscosity ν). At $x/H = 6$, the turbulent flow is attached in all the cases except for $L_z/h = 3$. As explained with the vorticity distribution, high strain and large velocity gradients in the expansion section cause decay and dispersion of turbulent structures, which explains the overall reduction in TKE for all the cases. In the outer region, low dissipation with high TKE production is consistent with the single VG study(Chapter 2). The negative peak in the turbulent transport for $L_z/h = 5$ and 7 signals energy transfer to the near-wall region in the attached flow. As expected, the dissipation and diffusion dominate inside the viscous layer for $y^+ < 5$.

Figures 3.12 and 3.13 show the TKE contributions at $x/H = 10$ and near the end of the domain at $x/H = 15$, respectively, where the significant decrease in the magnitude of TKE production in the outer regions and the higher dissipation and diffusion, especially in the near-wall region, ex-



(a) $L_z/h = 3$.

(b) $L_z/h = 5$.

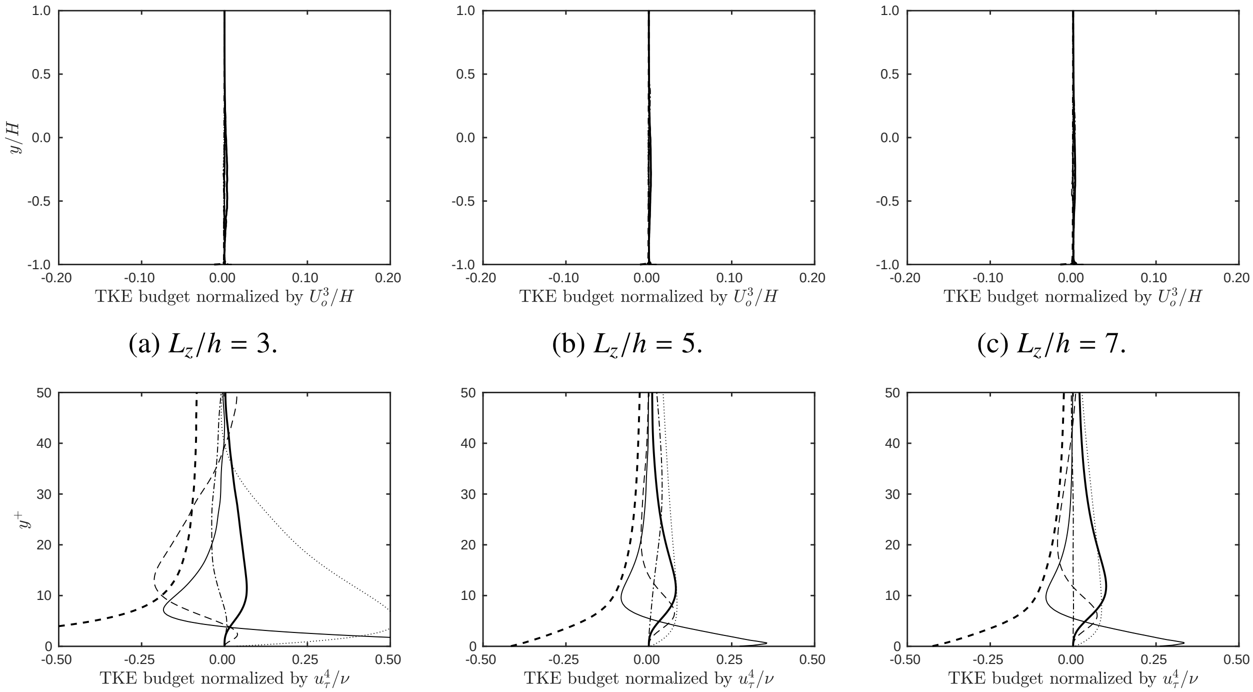
(c) $L_z/h = 7$.

(d) $L_z/h = 3$.

(e) $L_z/h = 5$.

(f) $L_z/h = 7$.

Figure 3.11: Normalized span-averaged TKE contributions in the expansion section at $x/H = 6$ for different spanwise spacings: −·−·, convection; ——, production; − − −, transport; ——, diffusion; - - - -, dissipation; ⋯ ⋯, velocity-pressure gradient.



(a) $L_z/h = 3$.

(b) $L_z/h = 5$.

(c) $L_z/h = 7$.

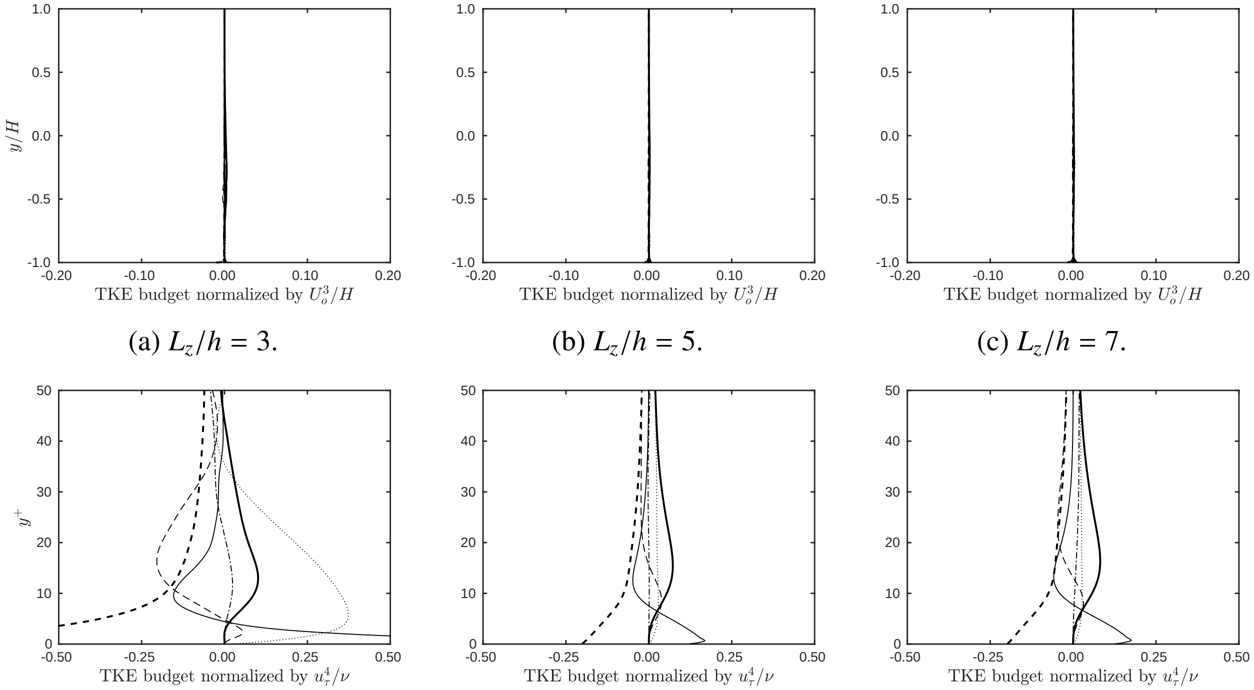
(d) $L_z/h = 3$.

(e) $L_z/h = 5$.

(f) $L_z/h = 7$.

Figure 3.12: Normalized span-averaged TKE contributions in the expansion section at $x/H = 10$ for different spanwise spacings: —·, convection; ——, production; – – –, transport; ——, diffusion; - - - -, dissipation; ··· ···, velocity-pressure gradient.

plains the overall drop in the magnitude of the TKE. The TKE production, transport, and diffusion are of a similar order in both the outer-flow and inner-wall regions for the cases with $L_z/h = 5$ and 7. For $L_z/h = 3$, the late reattachment of flow results in TBL with low momentum and large velocity gradients, which contribute to the TKE production as observed between $-0.5 < y/H < 0$. Moreover, the attached flow results in a favorable pressure gradient with $\overline{p'_{,x}} < 0$, such that the velocity-pressure gradient has a positive peak within $y^+ < 5$. Positive convection indicates that the attached flow convects the TKE in the streamwise direction towards the near-wall regions. Similar to the single VG study (Chapter 2), at $x/H = 15$ the TBL is in a state of non-equilibrium turbulence as the TKE production is not balanced by its dissipation, and does not recover by the end of the domain for all the spanwise spacings considered in this study.



(a) $L_z/h = 3$.

(b) $L_z/h = 5$.

(c) $L_z/h = 7$.

(d) $L_z/h = 3$.

(e) $L_z/h = 5$.

(f) $L_z/h = 7$.

Figure 3.13: Normalized span-averaged TKE contributions in the expansion section at $x/H = 15$ for different spanwise spacings: —·, convection; ——, production; −−−, transport; ——, diffusion; - - - -, dissipation; · · · · · ·, velocity-pressure gradient.

3.4 Conclusions

Wall-resolved large-eddy simulations are conducted at a Reynolds number of 19,600, based on the inlet boundary layer thickness and freestream velocity, to study flow over a backward-facing ramp modulated by an array of equally-spaced, submerged, wall-mounted cubes used as canonical vortex generators (VGs). In particular, the turbulent transport that results in the modulation of the separated flow over the ramp is investigated by varying the spanwise spacing between the VGs, which in turn is shown to modify the interactions between the VG-induced flow structures and those of the separated region. Although the flow separation and reattachment around the VGs show little dependence on the spanwise spacing as identical horseshoe vortex systems are formed around the individual cubes of the array, which extend in the near-wake to form multiple pairs of counter-rotating vortices. Moreover, these localized regions of high vorticity are associated with the production of turbulent kinetic energy, thus illustrating an effective mechanism of energy transfer from the freestream to the near-wall regions. Modulation of the separated flow depends on the relative intensity of the interaction between neighboring vortex pairs and the shear layer. The size of the counter-rotating vortex pairs depends on the height of the cubic array; however, their lateral spreading is governed by the spanwise spacing between the VGs. As a result, the intensity of the vortex pairs interacting with the shear layer is modified, compared to a problem with a single VG. For $L_z/h = 3$, the small spanwise spacing increases the blockage, thus resulting in more compact counter-rotating vortices, which succumb to high strain and large velocity gradients in the expansion section, and are therefore are unable to modulate the separated flow over the ramp. However, for $L_z/h = 7$, the large spanwise spacing results in negligible interaction between the neighboring VG-induced flow structures, and the resulting flow modulation over the ramp is similar to that of an isolated single VG (Chapter 2). On the other hand, the drag originating from the skin friction forces and pressure forces increases with the spanwise spacing. Based on the parameters considered in the present study, optimal modulation of the separated region is achieved with a cubic array with $h/\delta_0 = 0.6$, $x_{vg}/h = 3$, and $L_z/h = 5$ as evidenced by the significant reductions in the volume of separation over the ramp, while maintaining low coefficient of drag.

The present study is limited to a single line array of cubes of given geometry. The VG geometry and interaction between VGs on different rows may affect the horseshoe vortices and thus affect the flow separation region in different ways. Future studies of these phenomena would improve our understanding of separation modulation using passive vortex generators.

Part II:

Optimization of High-Order Discontinuous Galerkin Method for Next-Generation HPC Platforms

CHAPTER 4

Improving the Parallel Efficiency of Recovery-Assisted Discontinuous Galerkin Methods on Modern HPC Systems

This chapter is adapted from Tandon & Johnsen (2020). An accurate representation of certain flow phenomena can only be achieved if the discretization error is small, which can be achieved by employing high-order methods. A class of recovery-assisted discontinuous Galerkin (RADG) methods has been shown to provide arbitrarily high orders of accuracy by increasing the solution polynomial order p. The RADG methods explored here have compact stencils with dependence only on the nearest neighbors, thus reducing complications with communication in large-scale computations. Although, an increase in p increases the number of degrees of freedom, thereby significantly increasing the arithmetic operations performed, the increase in floating-point operations will be more than offset the reduction in data transfers. The arithmetic intensity of a class of RADG methods for hyperbolic systems of conservation laws is theoretically analyzed for polynomial order one through six in arbitrary dimensions. Different data cache models are considered and validated numerically on an Intel-Knights-Landing-based XSEDE Stampede2 node. Theory and numerical experiments demonstrate that RADG methods are able to provide increases in arithmetic intensity that will be necessary to make better utilization of on-node floating-point capability.

4.1 Introduction

High-fidelity numerical simulations of complex turbulent flows often demand high orders of accuracy, which can be achieved either by increasing the mesh resolution or by employing high-order methods. High-order methods are usually defined as having an order of accuracy greater or equal to two, which reduces the error E from the numerical discretization as $E \sim \mathcal{O}(h^n)$, where h is the characteristic grid spacing and $n \geq 2$. Tan *et al.* (2005), Desjardins *et al.* (2008), Bermejo-Moreno *et al.* (2013), Colella *et al.* (2011), Loffeld & Hittinger (2019), King & Kirby (2013), and others have explored the high-order variants of the traditional finite difference (FD) and finite volume (FV) methods, generally used in spatial discretization of the governing equations. However, it is well known that high-order accuracy may give rise to aliasing errors in FD schemes (Rogallo & Moin, 1984), which can lead to violation of the invariance of the governing equations with erroneous results, and therefore require special care. Furthermore, high-order accuracy in FD and FV discretizations is achieved by expanding the numerical stencil, i.e., more information from neighboring cells is needed to compute the solution at a give cell.

The dependency of high-order FD and FV methods on large stencils introduces several complications. In parallel computation, the increased stencil leads to more data movement and increased communication time. Boundary conditions can pose challenges due to the necessity for ghost cells or a modification of the spatial discretization. For implicit time solvers, these high-order FV methods with larger stencils require more memory and adversely impact the stability of iterative algorithms (Fidkowski, 2004; Mavriplis, 2002). Such issues become more prominent when conducting massively parallel simulations such as the flow simulations of Bermejo-Moreno *et al.* (2013) and Godenschwager *et al.* (2013) with over a trillion cells on more than a million cores.

The discontinuous Galerkin (DG) method combines the desirable properties of finite element and the FV method (Cockburn *et al.*, 2000). Arbitrarily high orders of accuracy can be achieved by adding degrees of freedom to each element to represent the solution as a high-order polynomial. Since the solution is allowed to be discontinuous across elements, borrowing from the FV method, the flux between immediately adjacent elements is used to exchange information, which preserves a compact stencil (Henry de Frahan, 2016). Thus, as demonstrated by Bey *et al.* (1995), Heinecke *et al.* (2014), Houba *et al.* (2019), and others, the DG method exhibits good parallel scaling on modern HPC platforms. The recovery-assisted DG method (RADG, Johnson & Johnsen, 2019) uses the information from a pair of neighboring elements to recover a high-order DG approximation of the flux at the interface of the two neighboring elements. The recovery procedure results in higher orders of accuracy than conventional DG methods. While the DG methods present many advantages, the large number of degrees of freedom inside each element, particularly for RADG methods, poses a challenge for on-node data motion and memory usage.

As high-performance computing (HPC) moves towards exascale computing and beyond, the gains in performance are coming from using heterogeneous architectures with reduced clock speeds and memory per processor. The traditional strategy of adding more computational units (CPUs) to computing clusters to achieve greater FLOP counts, motivated by Moore's law, has shifted in recent years, in large part due to the resulting power requirements (200 MW) and corresponding costs, as well as reaching Dennard's scaling (Dennard *et al.*, 1974) namely, limitations in memory bandwidth and capacity. On these heterogeneous architectures, the decreasing power cost of FLOP has exposed the power cost of data motion. Thus, FLOP is no longer the primary (on-node) cost factor for numerical simulation. A new trade-off must be made between data motion, memory usage, and operations (Brown *et al.*, 2010; Ashby *et al.*, 2010; Lucas *et al.*, 2014; Heroux *et al.*, 2020).

The objective of this work is to systematically address the trade-off between FLOP and bytes transferred for RADG discretizations of hyperbolic systems of conservation laws. Building on the formalism of Johnson (2019), the flux discretization is presented and the bounds for solution polynomial orders one through six, for both the operation count and the arithmetic intensity, are evaluated. These two quantities are used to determine the on-node performance based on the roofline model of Williams *et al.* (2009), which is described in section 4.2. The recovery-assisted discontinuous Galerkin formulation is presented in section 4.3. Section 4.4 discusses the theoret-

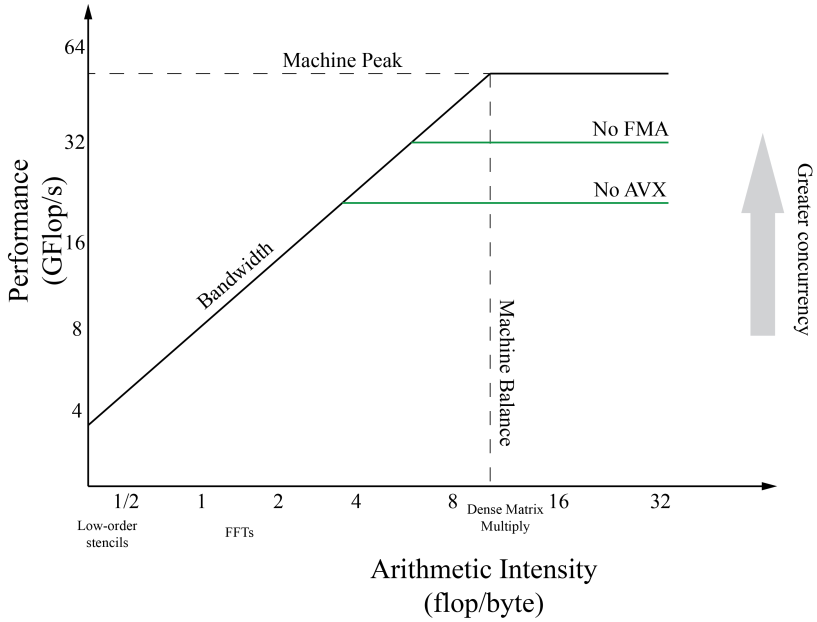


Figure 4.1: Example roofline model for a fictitious node architecture.

ical bounds for different data cache models, and the empirical verification results are presented in section 4.5. The chapter ends with concluding remarks in section 4.6.

4.2 Roofline Model

Figure 4.1 shows a roofline model for a fictitious node architecture. The vertical axis is the number of floating-point operations per second, which is ultimately bounded by the performance of the processor. The horizontal axis is the arithmetic intensity, expressed in terms of the number of floating-point operations per byte transferred. The diagonal line is the upper bound on the bandwidth, namely the maximum rate at which data can be provided to the processor. The optimal location on this plot is at the balance point (Williams *et al.*, 2009), where the bandwidth limit and the processing limit intersect. The arithmetic intensity of several representative algorithms studied by Williams *et al.* (2009) are shown, where the computation for low-order stencils is bandwidth-limited and is unable to exercise the full capability of the node.

Improvements and alternatives to the roofline model have been proposed (Sim *et al.*, 2012; Zhang *et al.*, 2014; Stengel *et al.*, 2015), introducing additional details such as overlapping communication and computation. While such details can be included in the roofline model, the basic roofline model will suffice for this work in order to theoretically understand the trade-offs in implementing recovery-assisted discontinuous Galerkin (RADG) discretizations. The roofline model has been used to guide the performance optimizations of a wide variety of algorithms (Williams *et al.*, 2009; Rossinelli *et al.*, 2011; Bermejo-Moreno *et al.*, 2013; Godenschwager *et al.*, 2013; Rossinelli *et al.*, 2013; Basu *et al.*, 2015; Modave *et al.*, 2016; Karakus *et al.*, 2019). Moving vertically on the roofline model requires one to leverage all of the performance-enhancing features of a processor, which is mostly an implementation question that will not be addressed in this work. Instead, here the roofline model is used to motivate algorithm design. Specifically, an attempt is made to derive best and worst-case estimates of the algorithmic intensity of RADG methods to understand the effectiveness of such higher-order methods as a strategy for moving beyond the bandwidth-limited region of the roofline model.

4.3 Recovery-assisted discontinuous Galerkin methods for hyperbolic systems

Consider the hyperbolic system

$$\frac{\partial u}{\partial t} + \nabla.\mathcal{F}(u) = 0, \tag{4.1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the D-dimensional spatial coordinate, $t \in \mathbb{R}^+$ depicts time, $u(\mathbf{x}, t) : \mathbb{R}^D \times \mathbb{R}^+ \to \mathbb{R}^n$ is a vector of n conserved variables, and $\mathcal{F}(u) : \mathbb{R}^n \to \mathbb{R}^{n \times D}$ are the vector-valued flux functions. In recovery-assisted discontinuous Galerkin (RADG) discretization, the computational domain $\mathbf{\Omega}$ is partitioned into M non-overlapping elements Ω_m, such that $\cap_{m=1}^{M} \Omega_m = \emptyset$ and $\cup_{m=1}^{M} \Omega_m = \mathbf{\Omega}$, and the elemental boundary is denoted by $\partial\Omega_m$. Each element's set of basis functions ϕ_m is a polynomial basis of at most degree p in a given direction and contains K members. In this work, the DG method is applied with uniform polynomial order for all elements. This constraint discards the possible benefits of an hp-adaptive mesh refinement procedure. For uniform grids, a full tensor product basis of degree p in each direction is employed, such that, $K = (p+1)^D$.

The numerical solution U^h within each element is a K-dimensional polynomial expansion within the solution space ϕ_m:

$$U^h(\mathbf{x} \in \Omega_m) = U^h_m(\xi(\mathbf{x})) = \sum_{k=1}^{K} \phi^k_m(\xi)\hat{U}^k_m, \tag{4.2}$$

where $\xi(\mathbf{x})$ is a mapping from the physical coordinate $\mathbf{x}$ to the reference coordinate ξ on the reference element Ω_{ref}. The governing equation 4.1 is satisfied in the weak form for element by integrating against the basis function as,

$$\int\limits_{\Omega_m} \phi^k_m \frac{\partial}{\partial t} U^h_m d\mathbf{x} + \int\limits_{\Omega_m} \phi^k_m \nabla.\mathcal{F}(U^h) d\mathbf{x} = 0, \qquad \forall k \in \{1, 2, \ldots, K\}. \tag{4.3}$$

Applying the divergence theorem and integrating by parts gives,

$$\int\limits_{\Omega_m} \phi^k_m \frac{\partial}{\partial t} U^h_m d\mathbf{x} = \int\limits_{\Omega_m} (\nabla\phi^k_m).\mathcal{F}(U^h_m) d\mathbf{x} \;-\; \int\limits_{\partial\Omega_m} \phi^k_m (\widetilde{\widetilde{\mathcal{F}}}.\mathbf{n}^-) ds. \tag{4.4}$$

The common flux values at the element interface are denoted as $\widetilde{\widetilde{\mathcal{F}}}$ and must be multiplied with element's outward normal $\mathbf{n}^-$ to complete the weak form of governing equation 4.1. In order to explain the evaluation of the flux terms, first a description of the recovery concept is provided.

4.3.1 Recovery

The RADG schemes for advection problems proposed by Khieu & Johnsen (2014) and extended by Johnson (2019), employ the recovery operator as a tool approximate the solution U along element-element interfaces and results in orders of accuracy of $2p + 2$ in the cell-average norm. The recovery concept originates from the observation that the discontinuous polynomial form of U^h is an attempt to replicate some globally smooth, underlying exact solution U (van Leer & Nomura, 2005). The guiding principle is to recover this underlying solution over a subdomain of the global spatial domain $\mathbf{\Omega}$. To keep the stencil compact, this subdomain is always chosen to be the union of two adjacent elements.

Consider a union of two adjacent elements, $\mathcal{U} = \Omega_A \ \cup \ \Omega_B$ with same polynomial order p. For each union, a recovery coordinate $\mathbf{r}$ must be defined; the origin of this coordinate is the interface centroid. A recovery basis ψ, supported over U, is defined with the recovery coordinate $\mathbf{r}$. For this work, Legendre basis for ψ are used. The recovered solution over the union is defined as a $2K$-dimensional polynomial expansion in the recovery basis

$$f(\mathbf{r}) = \sum_{n=1}^{2K} \Psi^n(\mathbf{r})\hat{f}^n, \tag{4.5}$$

where the $2K$ coefficients in the recovered solution require $2K$ constraints. Now, let W be some arbitrary variable approximated in the DG solution space over Ω_A and Ω_B (for example $W = U^h$). The recovered solution Johnson & Johnsen (2019) is required to be weakly equivalent to W_A^h and W_B^h over $\mathcal{U}$ with respect to the K DG basis functions of Ω_A and the K DG basis functions of Ω_B.

$$\int\limits_{\Omega_A} \phi_A^k f \, d\mathbf{x} = \int\limits_{\Omega_A} \phi_A^k W_A^h \, d\mathbf{x} \quad \forall k \in \{1, 2, \ldots, K\}, \tag{4.6a}$$

$$\int\limits_{\Omega_B} \phi_B^k f \, d\mathbf{x} = \int\limits_{\Omega_B} \phi_B^k W_B^h \, d\mathbf{x} \quad \forall k \in \{1, 2, \ldots, K\}. \tag{4.6b}$$

Discrete recovery operator

The constraints of equation 4.6 can be recast in matrix-vector form for a given variable, where the coefficients $\hat{\mathbf{f}}$ are calculated directly from a combined coefficient vector, $[\hat{\mathbf{W}}_\mathbf{A}; \hat{\mathbf{W}}_\mathbf{B};]$, using a single matrix-vector multiplication. The recovery procedure is used exclusively to calculate the interface quantity $f(0)$ given the coefficients $\hat{\mathbf{W}}$ of Ω_A and Ω_B.

$$\underbrace{\begin{bmatrix} \int\limits_{\Omega_A}\psi\phi_A^0 dx \\ \vdots \\ \int\limits_{\Omega_A}\psi\phi_A^{K-1}dx \\ --- \\ \int\limits_{\Omega_B}\psi\phi_B^0 dx \\ \vdots \\ \int\limits_{\Omega_B}\psi\phi_B^{K-1}dx \end{bmatrix}}_{2K\times 2K} \begin{bmatrix} \hat{f}^0 \\ \vdots \\ \hat{f}^{K-1} \end{bmatrix} = \underbrace{\begin{bmatrix} \int\limits_{\Omega_A}\phi_A\phi_A^0 dx & \mathbf{0}_{1\times K} \\ \vdots & \vdots \\ \int\limits_{\Omega_A}\phi_A\phi_A^{K-1}dx & \mathbf{0}_{1\times K} \\ --- & --- \\ \mathbf{0}_{1\times K} & \int\limits_{\Omega_B}\phi_B\phi_B^0 dx \\ \vdots & \vdots \\ \mathbf{0}_{1\times K} & \int\limits_{\Omega_B}\phi_B\phi_B^{K-1}dx \end{bmatrix}}_{2K\times 2K} \begin{bmatrix} \hat{\mathbf{U}}_A \\ \hat{\mathbf{U}}_B \end{bmatrix} \tag{4.7}$$

In practice, this entire process is stored in the discrete recovery operator, $\mathcal{R}$, for a given interface

$$f(r=0) = \mathcal{R}\begin{bmatrix} \hat{\mathbf{U}}_A \\ \hat{\mathbf{U}}_B \end{bmatrix} \tag{4.8}$$

where $\mathcal{R}$ has $2K$ columns and as many rows as the the number of quadrature points along the given interface. In the implementation used in this work, $\mathcal{R}$ is precomputed for each interface and can be used whenever a recovery operation is required to be performed.

Derivative-based recovery

The discrete recovery operator implementation detailed above relies on the formation of the recovery coordinate, the recovery basis, and the inversion of a linear system for each element-element interface in the domain. These operations are simpler to implement in 1D, however, in multi-dimensional case the recovery operator can become ill-conditioned (Johnson & Johnsen, 2019) on non-cartesian meshes, which threatens the scheme stability and accuracy. Johnson (2019) proposed a new derivative-based recovery implementation, which is free of recovery basis and the inversion of the associated linear system in equation 4.8, and replicates the accuracy of the typical recovery system on 1D meshes and 2D Cartesian meshes. The derivative-based recovery operation depends on recovery weights **C** and takes the following form for arbitrary solution order p

$$f(r=0) = \left(C_0 U_A^h + (1-C_0) U_B^h\right)|_{x=x_\mathcal{I}} + \sum_{j=1}^{p} C_j \left(\frac{\partial^j}{\partial r^j} U_A^h - \frac{\partial^j}{\partial r^j} U_B^h\right)|_{x=x_\mathcal{I}} \tag{4.9}$$

where the derivatives are evaluated by direct differentiation of the DG basis functions and all quantities are evaluated at the element-element interface ($x = x_\mathcal{I}$). The derivatives are written in terms of the recovery coordinate r, which points out of Ω_A into Ω_B

$$r(x) = \frac{(x - x_\mathcal{I}).n_A^-}{h}. \tag{4.10}$$

Here h is the uniform element width for a given uniform structured grid. As the difference in the derivatives vanishes, the interface approximation tends towards a linear combination of U_A^h and U_B^h. In cases where the derivative differences are nonzero, the recovery operation uses the jumps in the derivatives to form a correction to the interface approximation. In practice, the derivatives with respect to r in equation 4.9 are populated via the derivatives of the DG basis functions of an element, given as

$$\frac{\partial^j}{\partial r^j} U_m^h = \sum_{k=0}^{K-1} \hat{U}_m \frac{\partial^j}{\partial r^j} \phi_m^k. \tag{4.11}$$

Since the transformation from x to r is linear, each $\left(\partial^j \phi / \partial r^j\right)$ is obtained as follows:

$$\begin{aligned} \frac{\partial}{\partial r}\phi &= h \sum_{a=1}^{D} n_a \frac{\partial}{\partial x_a}\phi, \\ \frac{\partial^2}{\partial r^2}\phi &= h^2 \sum_{a=1}^{D}\sum_{b=1}^{D} n_a n_b \frac{\partial^2}{\partial x_a \partial x_b}\phi, \\ \frac{\partial^3}{\partial r^3}\phi &= h^3 \sum_{a=1}^{D}\sum_{b=1}^{D}\sum_{c=1}^{D} n_a n_b n_c \frac{\partial^3}{\partial x_a \partial x_b \partial x_c}\phi, \quad \text{etc.} \end{aligned} \tag{4.12}$$

The recovery weights **C** are functions of the type of recovery and the mesh uniformity index $Q = (h_A - h_B)/(h_A + h_B)$, which goes to zero for a uniform structured grid where $h_A = h_B = h$. For details regarding the evaluation of **C** and application of derivative-based recovery, readers are directed to Johnson (2019).

4.3.2 The steps in residual update

Equation 4.4 can be conveniently re-written, where the term on the left-hand side is expanded by using the definition in equation 4.2 which gives a $K \times K$ mass matrix M_m of the element Ω_m that depends only on the element's geometry and set of basis functions ϕ_m. Additionally, the spatial residual terms for a given element can be collected in a single residual vector R_m. The mass matrix can be inverted to directly calculate the temporal derivatives of the degrees of freedom $\hat{\mathbf{U}}_m$,

$$M_m^{row,col} = \int\limits_{\Omega_m} \phi_m^{row} \phi_m^{col} d\mathbf{x}, \tag{4.13a}$$

$$R_m^{row}(U^h) = \int\limits_{\Omega_m} (\nabla \phi_m^k).\mathcal{F}(U_m^h) d\mathbf{x} \; - \int\limits_{\partial\Omega_m} \phi_m^k (\widetilde{\mathcal{F}}.\mathbf{n}^-) ds, \tag{4.13b}$$

$$\frac{\partial}{\partial t}\hat{U}_m = M_m^{-1} R_m(U^h). \tag{4.13c}$$

The evaluation of the residual term at every time-step can be further broken down into the following steps:

1. Collocate U^h on the interface points and use recovery to get high-order solution approximations. Consider a quadrature point $\mathbf{x}_g$ along the interface $\mathcal{I} = \partial\Omega_A \cap \partial\Omega_B$ shared by Ω_A and Ω_B. Let U_L and U_R be the competing limits of U^h from inside Ω_A and Ω_B, respectively, at $\mathbf{x}_g$:

$$U_L = \lim_{x \to x_g} \tilde{U}_A, \qquad U_R = \lim_{x \to x_g} \tilde{U}_B. \tag{4.14}$$

   where $\tilde{U}_A = \mathcal{R}[\hat{\mathbf{U}}_A; \hat{\mathbf{U}}_B]$ is the recovered approximation of U_A^h in Ω_A and similar for Ω_B.

2. Evaluate the common flux $\widetilde{\mathcal{F}}$ at each quadrature point $\mathbf{x}_g$ along the interface $\mathcal{I} = \partial\Omega_A \cap \partial\Omega_B$,

by passing the U_L and U_R to a Reimann solver

$$\widetilde{\mathcal{F}} = Rie\left(U_L|_{x_g}, U_R|_{x_g}, \mathbf{n}_A^-\right) \tag{4.15}$$

where $\mathbf{n}_A^-$ is the outward normal from Ω_A at each $\mathbf{x}_g$.

3. Evaluate the solution approximations U_m^h at each interior quadrature point within each element Ω_m.

4. Over the interior of each Ω_m, the flux function $\mathcal{F}$ is directly calculated using the DG approximation U_m^h.

4.3.3 Temporal discretization

The DG spatial discretization is typically paired with an explicit time integration method to integrate forward in time from some initial condition $U(\mathbf{x}, 0)$. For example, if applying the forward Euler method, then $\hat{U}_m(t+\Delta t) = \hat{U}_m(t) + \Delta t.M_m^{-1} R_m(U^h(t))$ for each element Ω_m, where the timestep size Δt is determined based on the Courant-Friedrichs-Lewy (CFL) constraint. A popular choice with DG methods is the explicit, four-stage, fourth-order Runge-Kutta method (Cockburn *et al.*, 1989; Gottlieb & Shu, 1998; Schwartzkopff *et al.*, 2004; Johnson & Johnsen, 2019).

From the perspective of arithmetic intensity for the RADG discretizations of hyperbolic problems, it is most informative to consider the evaluation of the residual term (effectively, the right-hand side (RHS) of the equation 4.13. The majority of floating-point operations occur in the evaluation of this RHS vector. For multi-stage methods like Runge-Kutta, the residual will be computed at least once for each stage, and the predicted states resulting from the stage evaluations are weighted and summed. Given that the RHS evaluation is dominant, the details of the time-stepping are not considered and the focus of this work will be the approximation of the RHS of equation 4.13.

4.4 Analysis of arithmetic intensity

This section derives the equations for the floating-point operations (FLOP) and data transfer costs of the methods as functions of the box size, the number of dimensions of the problem, and the order of accuracy. A box-structured grid is employed for the present analysis. For large-scale problems, it is common to subdivide the domain into smaller rectangular boxes that can be computed in parallel. At each step, boxes are surrounded by a layer of ghost cells containing the neighboring data upon which the box is dependent, allowing computation in boxes to be done independently. At the edge of the problem domain, boundary data fill the ghost cells. Boxes are distributed over processors, with each box assigned to one processor. A processor may possess multiple boxes, which can be computed in parallel on a multi-core processor, either by assigning a thread to each box or by assigning multiple threads within a box. At the beginning of each step (and as necessary during a step), processors communicate to exchange ghost data.

Let, W be the number of ghost cells needed on each side, in each direction. Thus the total number of ghost cells surrounding a box of length N per side in D dimensions is

$$N_g = (N + 2W)^D - N^D. \tag{4.16}$$

Figure 4.2 shows a graphical illustration of a 2D box-structured grid employed for RADG discretization with $W = 1$. Note that the corner ghost cells may not be required, but to keep the analysis simple the corresponding costs are included. Furthermore, it is assumed that only one box is loaded per node and all ghost-filling operations are performed off-node. Except for the cost of the Riemann solver and the cost of the flux function, the FLOP and data transfer costs increase proportionally to the number of system components. Therefore, without loss of generality, the single-component case is considered. The costs of the Riemann solver and flux function are problem-specific, so we leave their costs as parameters in the expressions, denoting their FLOP costs per face as f_R and f_F respectively.

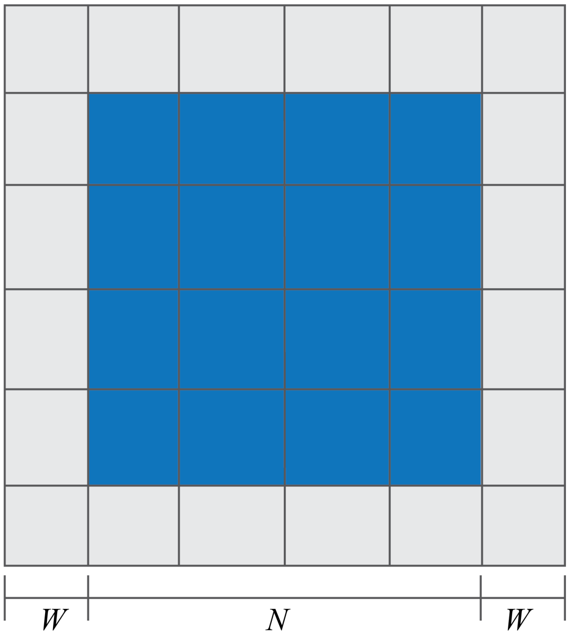


Figure 4.2: Graphical illustration of a $2D$ box-structured grid employed for RADG methods. Blue elements constitute the domain of interest and the outer grey elements form the ghost cell region. For RADG, the ghost cell width is $W = 1$, giving a total of $(N + 2W)$ elements in each direction.

To derive the bounds on the arithmetic intensity for RADG discretization applied to hyperbolic systems, first a degenerate case of no-cache is considered, which is analogous to poor cache utilization (Olschanowsky *et al.*, 2014). This provides an upper bound on the number of data transfers that must be performed, and thus providing a lower bound on the arithmetic intensity. Then a fictional infinite-size cache is considered, which provides a lower bound on the number of data transfers and a limit to performance in the ideal scenario. For a more practical implementation, a finite-size cache is considered and number of operations are counted for a simple tiling strategy.

4.4.1 Machine model and operation costs

For the purposes of calculating arithmetic intensity, the following abstract machine model is assumed. The machine is composed of a processor and two levels of memory: an unlimited amount of slow memory and a limited amount of fast memory. Data can be transferred in both directions between slow and fast memory. Data transferred to a location in either memory overwrites the previous value held there. At the start of processing, all data are stored in the slow memory and the fast memory is empty. Computation is only performed by the processor on data residing in the fast memory, so any data to be operated on must first be transferred from slow memory. Computation takes the form of arithmetic operations that load one or more operands from fast memory and return the result of the computation back to fast memory, never to slow memory. Computation is not considered complete until the final results are stored back in the slow memory.

The two costs in this model are the data transfers between the slow and fast memory, and the cost of arithmetic operations. All data transfers are assumed to have the same cost whether to or from the fast memory and all arithmetic operations are considered to have the same costs. In particular, the method requires only addition and multiplication operations. Fractional terms, such as the derivative-based recovery coefficients, can be assumed to be pre-computed by the compiler, so we assume the method to be free of division operations. To count the number of operations, the definition of a FLOP is adopted from Golub & Van Loan (2012), where the common operations such as the vectorized and fused-multiply-add (FMA) operations on modern central processors (CPUs) and accelerators, are decomposed into their respective stand-alone operations. Therefore, if vectorized addition of two vectors $\overrightarrow{\mathbf{a}}$ and $\overrightarrow{\mathbf{b}}$ of length four each results in a new vector $\overrightarrow{\mathbf{c}} = \overrightarrow{\mathbf{a}} + \overrightarrow{\mathbf{b}}$ of length four, then the number of operations is taken to be four.

Clearly the computation and data transfer costs for the algorithm are both minimized if the size of fast memory is large enough to hold all intermediate results. In other words, if the only data transfers that occur are at the beginning of the problem, when loading the entire problem data into fast memory, and at the end, when transferring the final result to slow memory. In general, the size of fast memory is too small (~ 2.5 MB per core on modern HPC platforms, Loffeld & Hittinger, 2019; top, 2019), so additional memory transfers of intermediate results are necessary. Note that the case of unlimited fast memory is not necessarily an upper bound on arithmetic intensity. To minimize the amount of data that must be kept in fast memory, an implementation might redundantly compute some intermediate results such as ghost cell data, and this can actually increase the arithmetic intensity over the optimal case. However, the number of operations would increase as well, possibly to a degree resulting in a net loss in performance. Therefore, a tiling strategy

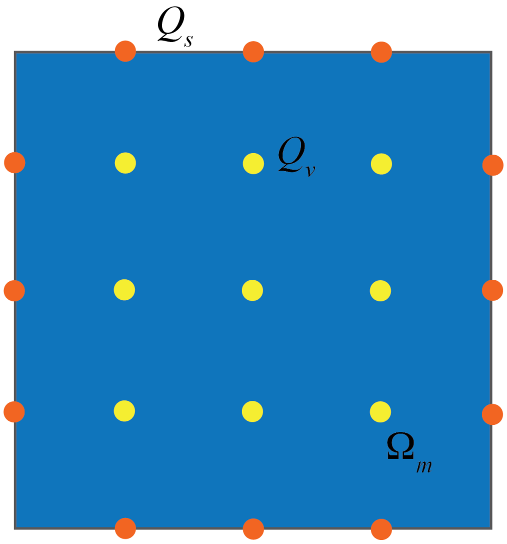


Figure 4.3: Illustration of a 2D element showing the interior (Q_v in yellow) and interface (Q_s in orange) quadrature points for $p = 2$. Here $Q_s = (p{+}1) = 3$ for each interface, and $Q_v = (p{+}1)^2 = 9$.

with an arithmetic intensity that is higher than in the infinite-cache case would not necessarily be a higher-performing strategy. The infinite-cache case is simply the case that minimizes the total number of operations of the algorithm.

4.4.2 No-cache case

The no-cache case provides a lower bound on the arithmetic intensity in the case when the cache is poorly utilized. It is assumed that there is just enough local storage (fast memory) to hold all data needed to perform computations at each cell or face, but when moving to a new cell or face, the required data must be loaded from slow memory. In addition, it is assumed there is no reuse of data between steps. Since there is no use of cache and because each computation must load its operands anew, multi-threading does not affect the total number of data transfers or flop computations.

Figure 4.3 shows an illustration of a 2D element. For a DG order p, the number of degrees of freedom $K = (p + 1)^D$, for a D-dimensional problem. The interior quadrature points $Q_v = K$, and the quadrature points on each face is $Q_s = (p + 1)^{D-1}$. Table 4.1 lists the steps in residual update (section 4.3) and the formula for FLOP count of each step, while table 4.2 lists the cost of data transfer for each step. For illustration, the cost breakdown for the first step is explained.

#	Calculation	FLOP
1	U_L or U_R, at interface quad. points	$2(Q_s \times 4K) \times N_v$
2	$\tilde{\mathcal{F}}(U_L, U_R)$, at interface quad. points	$(f_R \times Q_s) \times N_v$
3	U^h at interior quad. points	$(Q_v \times 2K) \times N^D$
4	$\mathcal{F}(U^h)$ at interior quad. points	$(f_F \times D \times Q_v) \times N^D$

Table 4.1: Floating-point operations (FLOP) per step for no-cache case.

#	Calculation	data transfer
1	U_L or U_R, at interface quad. points	$2(\mathcal{R} + 2Q_v + Q_s) \times N_v$
2	$\tilde{\mathcal{F}}(U_L, U_R)$, at interface quad. points	$(3Q_s) \times N_v$
3	U^h at interior quad. points	$(2Q_v) \times N^D$
4	$\mathcal{F}(U^h)$ at interior quad. points	$(2D + 1) \times Q_v \times N^D$

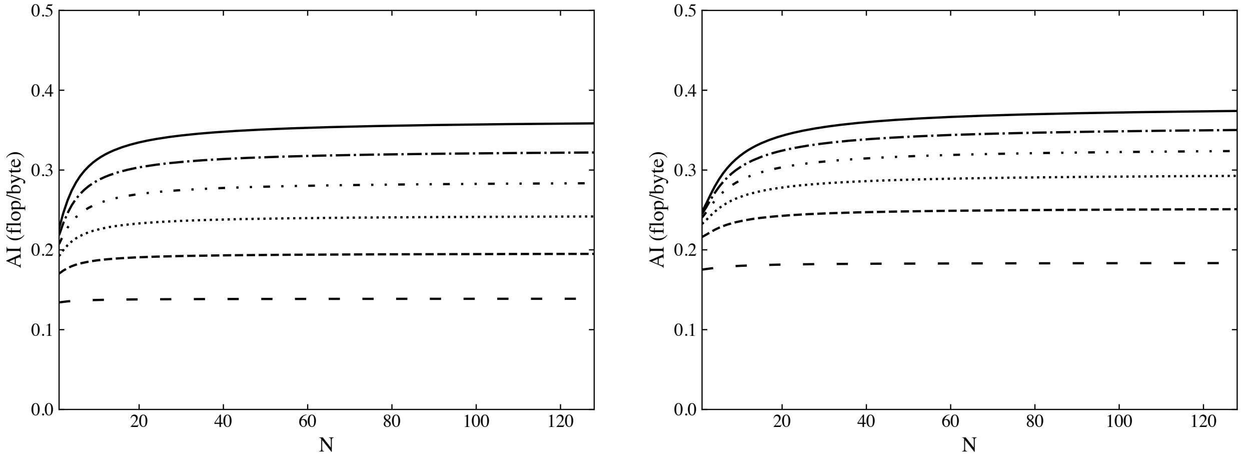
Table 4.2: Data transfers per step for no-cache case.

Floating-point operations: Step 1. Evaluate U_L and U_R at the interface between neighboring elements. For each face of the element,

(i) Evaluate a recovered approximation $\tilde{U}$ using the discrete recovery operator $\mathcal{R}$ which is a matrix of size $Q_s \times 2K$, resulting in a total of $Q_s \times 4K$ operations per interface.

To determine the number of faces, for a box containing $(N + 2W)$ elements along any direction D, there are $(N + 2W + 1)$ planes of faces that touch the interior $(N + 2W)^D$ cells of a box. There are $(N + 2W)$ cells in each of the $(D - 1)$ directions transverse to each plane; recall that the additional $2W$ cells per direction are the ghost cells needed. Thus, the number of faces in any plane is $(N + 2W)^{D-1}$, there are $(N + 2W + 1)$ planes per direction, and there are D directions for a total of $N_v = D(N + 2W + 1)(N + 2W)^{D-1}$ faces that must be computed in this step.

For each interface, step 1 requires U_L and U_R. Thus the recovery is performed twice, one for each of the two neighboring elements. This gives the total FLOP count for step 1: $2(Q_s \times 4K) \times N_v$. Here and in subsequent estimates, the count is approximate as recovery is performed at all faces in the ghost regions, even if they are not used in the final update. This is a small number of additional faces (relative to the used faces) because they occur near the outer edges of the $(N + 2W)^D$ box.
Data transfers: In Step 1 of the residual update, for each interface of each element



(a) 2*D*.

(b) 3*D*.

Figure 4.4: Arithmetic intensity for the no-cache case for RADG method with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ········, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

(i) Load the precomputed recovery operator. *One $\mathcal{R}$ load.*

(ii) Load the degrees of freedoms of both the neighboring elements. $2Q_v$ *loads.*

(iii) Store the recovered value ($\tilde{U}$) at each quadrature along the interface. Q_s *stores.*

As explained in the FLOP case, the total number of faces computed is $N_v = D(N + 2W + 1)(N + 2W)^{D-1}$. Step 1 requires the evaluation of U_L and U_R at each interface, and thus total data transfer needed are $2(\mathcal{R} + 2Q_v + Q_s) \times N_v$. Note that when using double-precision floating-point values, each transfer is 8 Bytes.

Figure 4.4 shows the arithmetic intensity for RADG orders $p = 1$ through 6 for the no-cache case. For box length $N \leq 128$, the arithmetic intensity values are below 0.4. In the asymptotic limit, the maximum arithmetic intensity is achieved by $p = 6$, with an intensity of 0.37 for a 2D problem and 0.39 for a 3D problem. Current machines have a flops-to-byte ratio of five or greater (Williams *et al.*, 2009). As expected, without a cache, a significant arithmetic intensity is not achievable. Additionally, at large box sizes, the arithmetic intensity is independent of N as with no-cache, the amount of data transferred is often proportional to the number of operations (Loffeld & Hittinger, 2019).

#	Calculation	data transfer
1	U_L or U_R, at interface quad. points	$Q_v \times (N + 2W)^D + (\mathcal{R} + 2Q_s) \times N_v$
2	$\tilde{\mathcal{F}}(U_L, U_R)$, at interface quad. points	$(3Q_s) \times N_v$
3	U^h at interior quad. points	$(2Q_v) \times N^D$
4	$\mathcal{F}(U^h)$ at interior quad. points	$(2D + 1) \times Q_v \times N^D$

Table 4.3: Data transfers per step for infinite-size cache case.

4.4.3 Infinite-size cache case

Consider an idealized machine with an infinite cache. The number of data values that must be loaded in each step consists of the entire box's worth of cells or faces that must be used to compute each step. Each value must be transferred exactly once. Likewise, the entire box's worth of results must be written at the end of the step. Multi-threading may allow the values to be loaded or stored more quickly, but does not change the total number of values that must be transferred, and therefore the formulas are agnostic to multi-threading in this case.

The number of floating-point operations computed is identical to the no-cache case, so the expressions for FLOP counts are those in table 4.1. The data transfers for the infinite-size cache case is listed in table 4.3. Since the cache is infinite in size, the number of loads in each step is simply the number of cells or faces that must be touched for input, and the number of stores is the number of cells or faces written to for output. The breakdown for the first step is described as an example.

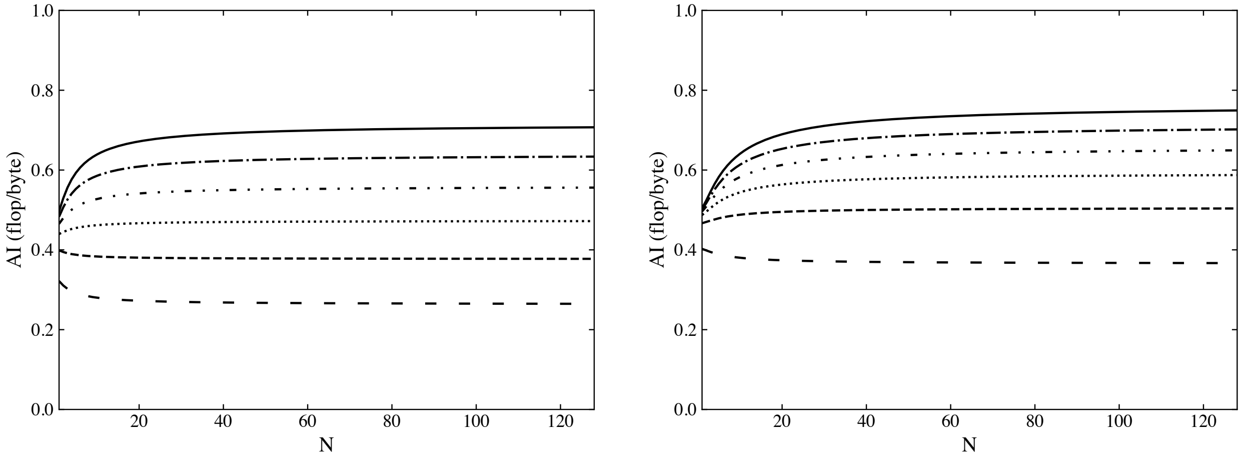
Data transfers: In Step 1 of the residual update, for the entire domain

(i) Load the precomputed recovery operator for all the interfaces. $\mathcal{R} \times N_v$ *loads*.

(ii) Load the degrees of freedoms of all the elements. $Q_v \times (N + 2W)^D$ *loads*.

(iii) Store the recovered values ($\tilde{U}$) at quadrature points along all the interface. $2Q_s \times N_v$ *stores*.

The total data transfers for step 1 are $Q_v \times (N + 2W)^D + (\mathcal{R} + 2Q_s) \times N_v$. Similar to the no-cache case, the corner ghost cell data may not be needed, but for simplicity it is included in this analysis.



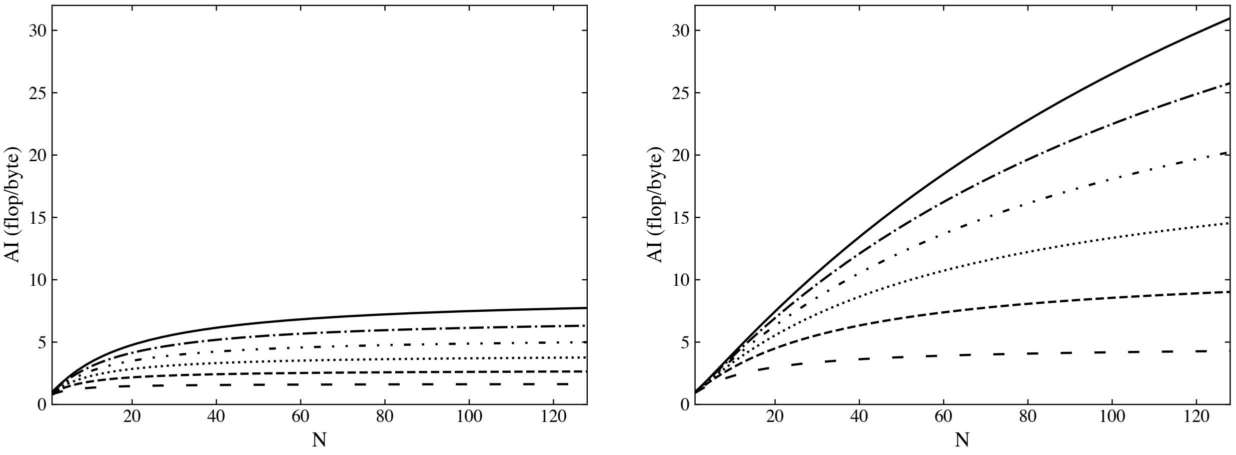
(a) $2D$.

(b) $3D$.

Figure 4.5: Arithmetic intensity for the infinite-size cache case for RADG method with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ·······, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

Figure 4.5 shows the arithmetic intensity for RADG orders $p = 1$ through 6 for the infinite-size cache case. For $2D$ and $3D$ problems, the arithmetic intensity is less than 1.0. Most current machines have a flops-to-byte ratio of five or greater (Williams *et al.*, 2009). For lower order p in $2D$, the large amount of data transfers per step increases with the box size and overshadows the increase in FLOP, which results in performance degradation. Therefore, in practice, a high arithmetic intensity is not achievable without keeping data in cache between steps.

In the idealized case of an infinite-size cache, all data between steps is kept in the cache and only the input data for the first step and the final result at the last step is transferred. The total number of flops is still the sum over all steps. Figure 4.6 shows the arithmetic intensity in such a case for $2D$ and $3D$ problems. The aggregate arithmetic intensity is much higher than the per-step arithmetic intensity. For $2D$ problems, when the box size has length $N = 128$, the arithmetic intensity of the RADG method with order $p = 6$ is slightly under 8, which is approximately the machine balance for most current mid-life production HPC machines such as Intel-Knights-Landing-based machines (XSEDE Stamede2, Koskela *et al.*, 2018). The corresponding arithmetic intensity for the $p = 3$ method is around 4. While not high enough to reach the machine balance of most machines, it is still a large improvement over low-order methods. For $p = 1$, the arithmetic intensity is approximately 2 only. For 3D problems, when the box size is of length 128, the $p = 3$



(a) $2D$.

(b) $3D$.

Figure 4.6: Arithmetic intensity for RADG method when using cubicle tiles of length N with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ·······, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

method achieves an arithmetic intensity of about 15, which is on par with state-of-the-art machines (OLCF's Summit, YarKhan *et al.*, 2019). For the same box size, $p = 6$ achieves an arithmetic intensity over 30.

The arithmetic intensities for the idealized case suggest that the methods have the potential to improve machine utilization, provided the cache can retain most intermediate data between steps. Of course, the caches on physical machines do not have an infinite size, so a tiling strategy is required to make good use of the cache. The arithmetic intensities under a simple tiling strategy is considered in the next subsection.

4.4.4 Finite-size cache case

A simple approach to using a finite-size cache is to divide the box into cubical tiles. Figure 4.7 illustrates the subdivision of a cubicle box and the data that is kept in the cache in this case. The starting box is depicted as the outer box with dashed lines. The box is partitioned further into cubical tiles of length T. Table 4.4 lists the data transfer for the case with a finite-size cache. Three types of data are used when computing the residual over a tile.

(i) A T^D cube of elements that hold the accumulating residual, depicted in figure 4.6 as the inner

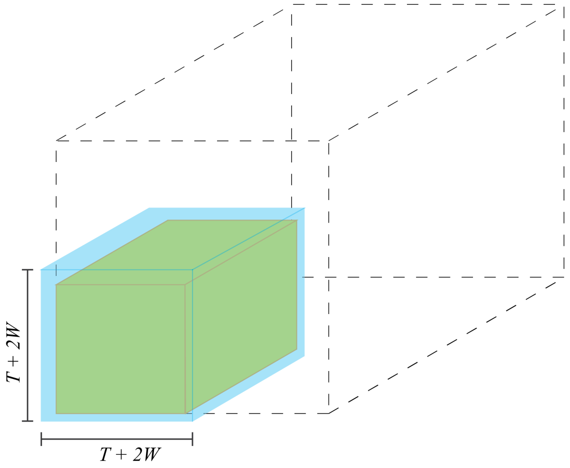


Figure 4.7: Illustration of a tiling strategy to subdivide a domain into cubical tiles.

#	**Calculation**	**data transfer**
1	U_L or U_R, at interface quad. points	$Q_v \times (T^D + (T+2W)^D) + \mathcal{R} \times 2D(T+2W)^{D-1}$
2	$\tilde{\mathcal{F}}(U_L, U_R)$, at interface quad. points	-
3	U^h at interior quad. points	-
4	$\mathcal{F}(U^h)$ at interior quad. points	$Q_v \times T^D$

Table 4.4: Data transfers per step for finite-size cache case with cubical tiles of length T.

cube of the tile. The flux divergence data must be re-accessed for each change in direction in order to accumulate the contribution of the flux divergence from that direction. As such, it is ideally kept in cache to avoid an additional transfer from slow memory for each change in direction.

(ii) An extended cube of $(T+2W)^D$, shown as the outer surrounding cube of the tile, holds the starting elemental data, including the ghost cell dependencies of width W. This data also must be re- accessed for each change of direction, so ideally is kept in cache to avoid multiple transfers from slow memory.

(iii) Finally import the discrete recovery operator precomputed at all interfaces.

Thus the total amount of data that must be kept in cache per tile is $Q_v \times (T^D + (T+2W)^D) + \mathcal{R} \times 2D(T+2W)^{D-1}$.

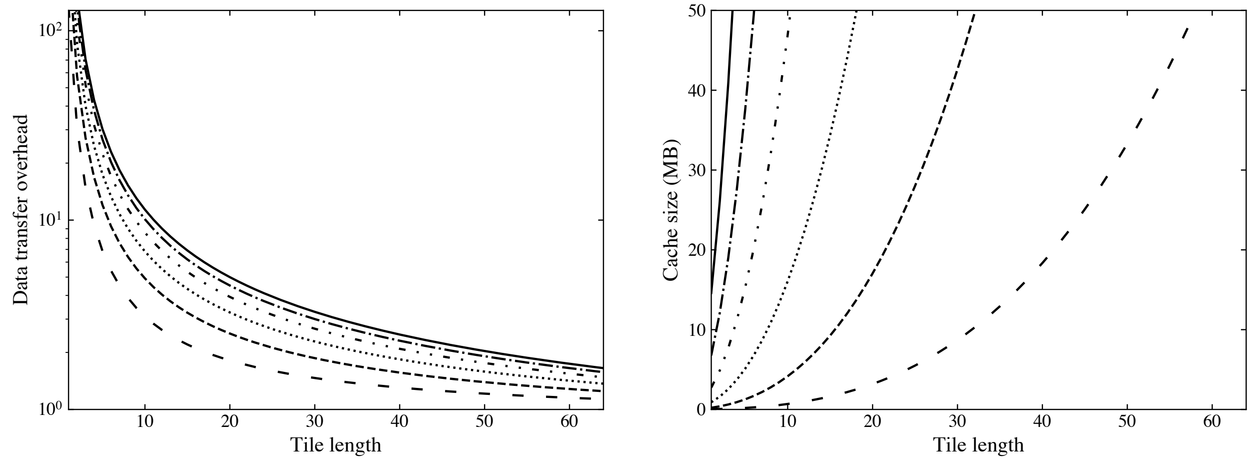
Tiling in this manner is equivalent to subdividing the problem domain into boxes, except that data for the ghost cells around the box is not duplicated through explicit ghost cell exchange, but rather is taken from the shared array of input values. For now, it is assumed that none of the ghost cell data overlapping with neighboring tiles carries over within the cache when moving between tiles. This suggests that the expressions for the FLOP in table 4.1 as well as the arithmetic intensity from figure 4.6, are directly applicable to this case. The size N in the formulas is reinterpreted as the length of the tile, T, such that the tile size is restricted to fit within the cache.

As discussed earlier, multi-threading would not affect the total number of operations within a box, and that holds even for a tile. However, if the tiles are computed concurrently, the ordering of when tiles are computed could be affected. Since the analysis presented here conservatively assumes that the ghost cell data are never already in the cache, multi-threading over tiles does not change the expressions for the number of operations that must be performed. Multi-threading can change the rate (bandwidth and FLOP rate) at which the operations are performed, but does not affect the balance of operations (arithmetic intensity).

While using cubicle tiles for finite-size cache shows great improvements in balance of operations, it must be noted that the arithmetic intensity is smaller when partitioning into smaller regions than large ones, as seen in figure 4.6, so the constraint on the size of the tile limits the maximum achievable arithmetic intensity. In addition, the ghost cell data overlapping with a neighboring tile need to be reloaded when moving to that tile, which increases the total communication volume due to some values being reloaded multiple times. The estimate of data transfer overhead of tiling, as the increase in data transfers that must be enacted from computing the residual for a box through tiling versus the data transfers for computing without tiling using an infinite-size cache, is given as

$$\frac{m^D \times (\text{cost of data transfer per tile})}{(\text{cost of data transfer per box with infinite cache})}. \tag{4.17}$$

Here it is assumed that the length of each tile divides the box length evenly, that is, $N = m \times T$. Figure 4.8a shows the overhead on 3D problems when subdividing a box of size $N = 128^3$ into sub-tiles. The data transfer overhead becomes exorbitant if the tile length becomes much smaller



(a) Data transfer overhead of tiling on a 128^3 as a function of tile size.

(b) Required cache size for 3D tiles of length up to 64 on each side.

Figure 4.8: Data transfer overhead and cache size requirement for RADG method with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ·······, $p = 3$; - · · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

than $T = 32$ in length. In particular, the overhead when $T = 32$ for $p = 6$ method is three times the cost without tiling and that of $p = 3$ method is approximately two times the cost without tiling.

Figure 4.8b shows the cache space requirements per component for 3D problems. For $p = 6$, the tile of length $T = 1$ requires 15 MB of cache space, whereas $p = 3$ requires approximately 1.5 MB for a $T = 1$ tile. Last-level caches on current HPC-grade machines are sized to approximately 2 to 2.5 MB per core. Therefore, the space requirements for a single-component 3D problem at higher p cannot fit into the last level of cache sizes of current machines. In addition, the governing equations of many problems of interest are multi-component. The Euler equations, for example, have five components in 3D. Therefore, a better implementation of RADG discretization along with a more economical caching strategy would be needed to realize the full arithmetic intensity on 3D problems.

4.4.5 Lowering the cache space requirement for 3D problems

Up until now, the Step 1 of residual update was considered to use the discrete recovery operator $\mathcal{R}$, which was precomputed for each interface and stored. Thus, in the evaluation of Step 1, $\mathcal{R}$ of each interface his loaded from the slow to the fast memory. As an alternative, the derivative-based

#	Calculation	data transfer
1	U_L or U_R, at interface quad. points	$Q_v \times (T^D + (T+2W)^D) + Q_s \times D(T+2W+1)(T+2W)^{D-1}$
2	$\tilde{\mathcal{F}}(U_L, U_R)$, at interface quad. points	-
3	U^h at interior quad. points	-
4	$\mathcal{F}(U^h)$ at interior quad. points	$Q_v \times T^D$

Table 4.5: Data transfers per step for finite-size cache case with cubical tiles of length T for derivative-based recovery.

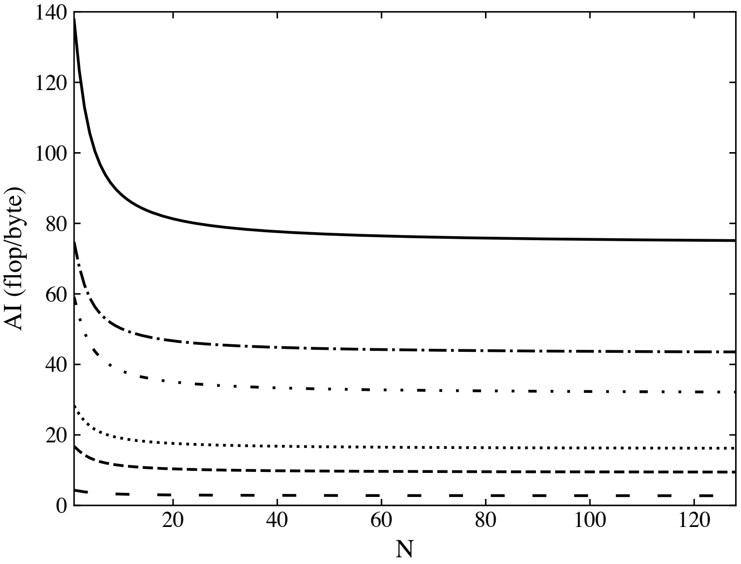
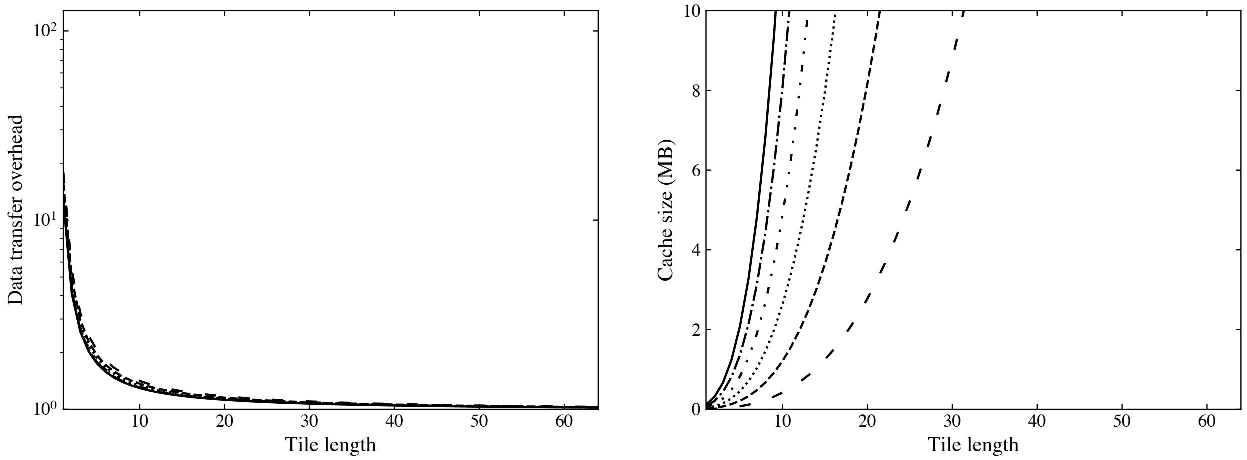


Figure 4.9: Arithmetic intensity for the finite-size cache case with derivative-based recovery for RADG method with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ········, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

recovery operation is implemented. Johnson (2019) formulated the derivative-based approach and verified the accuracy to be same as the discrete operator on 2D and 3D cartesian meshes. Using the definition of derivative-based recovery from equation 4.9, the total number of operations performed at each quadrature point is on the order $4K$, and thus the FLOP count is same as that described in table 4.1. Since no discrete operator is saved, the number of loads for step 1 are reduced. The expected data transfers for this approach are listed in table 4.5.

Figure 4.9 shows the arithmetic intensity for the derivative-based recovery for the case of finite-size cache with cubical tiling. Due to increased FLOP the arithmetic intensity sees significant improvement. The efficiency at larger tile sizes decreases because the data residing in the memory



(a) Data transfer overhead of tiling on a 128^3 as a function of tile size.

(b) Required cache size for 3D tiles of length up to 64 on each side.

Figure 4.10: Data transfer overhead and cache size requirement for RADG method with derivative-based recovery with varying polynomial order:− −, $p = 1$; - - - -, $p = 2$; ·······, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

increases faster than the FLOP count, thus resulting in a decrease in arithmetic intensity. At tile length $T \leq 8$, the arithmetic intensity for $p = 6$ is approximately 90, over 20 for $p = 3$, and about 3.5 for $p = 1$. Furthermore, the derivative-based implementation relaxes the cache requirements. Figure 4.10 shows the estimate of data transfer overhead due to tiling, similar to figure 4.8, and the cache requirements. For $p = 6$, the data transfer overhead for tile length $T \leq 32$ is reduced. In particular, for $T = 8$, the data transfer overhead is 1.2 times the cost without tiling. Compared to the discrete recovery operator implementation, the derivative-based recovery implementation of reduces the cache size required such that for $p = 6$, tile of length that can be loaded in 1 MB of cache has increased to $T = 4$. Reduction of cache size is valid for lower order p as well, where the tile length requiring 1 MB of cache storage has increased to $T = 8$ for $p = 3$, and $T = 16$ for $p = 1$.

Additional reduction in the cache size requirements can be made by exploring more economical caching strategies. A simple modification to the cubical tiling was proposed by Loffeld & Hittinger (2019), where flattened rectangular tiles of size 32 × 32 × 8 were used for an eighth-order finite-volume scheme, and reported that the cache requirement reduced by half. A similar strategy can be

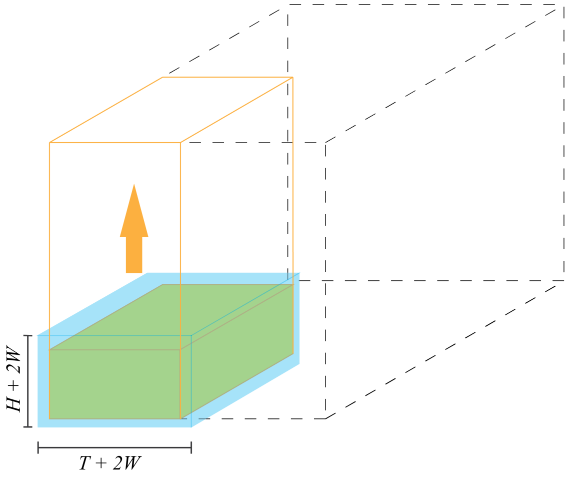


Figure 4.11: Vertical flattened rectangular tile iteration.

employed here. While the reduced volume of tile would require less space in the cache, the trade-off would be that the surface-to-volume ratio of the tile would increase, resulting in higher overhead for reloading of the ghost cell data. This can be avoided by moving from tile to tile in a vertically iterated manner. Figure 4.11 illustrates this tiling strategy. For every tile-wide column in the box, the first tile is computed at the bottom of the column, which is followed by the tile immediately above it and so on, until the column is completely computed and then a new column is chosen. In a multi-threading situation, threads would be assigned to columns. It is conservatively assumed that neighboring columns do not share ghost cell data through the cache. By iterating vertically in this manner, the ghost cell data at the bottom of a tile are still in cache from the previous tile. Part of the non-ghost cell data is as well, but for simplicity it is assumed that the data is re-fetched.

For a $3D$ problem with tile sizes of $T \times T \times H$, where H is the height of the tile, the amount of cache space taken up becomes

$$Q_v(H + 2W)(T + 2W)^2 + Q_v(HT^2) + Q_s N_F,$$

where $N_F = 2(T + 2W + 1)(T + 2W)(H + 2W) + (H + 2W + 1)(T + 2W)^2$ is the total number of interfaces in each rectangular tile. From figure 4.10 we observe that for $p = 6$, a cubical tile $T = 8$ requires 5.5 MB of cache. Therefore, a flattened rectangular tile of size $8 \times 8 \times 2$ is selected, which

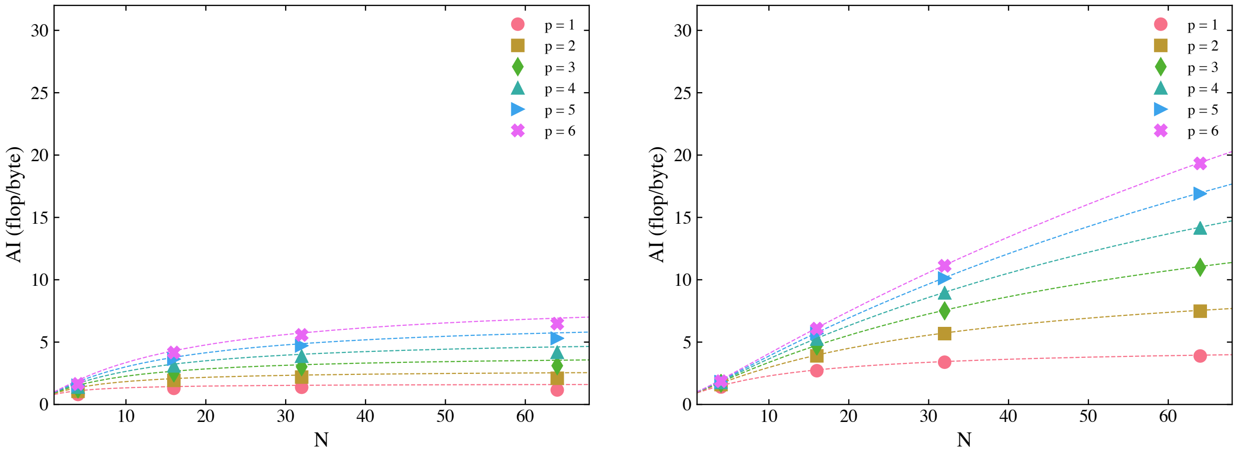
p	**Tile dim.**	**cache size** (MB)	**Arith. Intensity**
1	32^3	8.5	3.0
1	$32\times32\times4$	1.4*	3.3*
3	32^3	52.4	17.2
3	$32\times32\times4$	8.5	19.5
3	16^3	17.5	18.4
3	$16\times16\times4$	2.3*	20.2*
6	32^3	245.3	75.8
6	$32\times32\times4$	39	87.8
6	16^3	34.4	82
6	$16\times16\times4$	10.6	91.7
6	8^3	5.5	93
6	$8\times8\times4$	3.1*	98.6*
6	$8\times8\times2$	2*	106.3*

Table 4.6: The difference in costs of the cubical versus vertically iterated caching strategy. (*) marked entries suggest an optimal tile size.

reduces the cache space requirement to 2 MB. Similarly for $p = 3$, the rectangular tile of size $16\times16\times4$ reduces the cache requirement from 8 MB to 2.3 MB. Table 4.6 lists the values of cache size and arithmetic intensity for different DG orders with different cache tiling. Note that for the flattened rectangular tile, the arithmetic intensity is enhanced compared to its cubical counterpart.

4.5 Numerical experiments

To verify the arithmetic intensity predictions, the RADG methods from orders one through six, with discrete recovery operator, were implemented in C++, and hardware performance counters on a node of an Intel-Knights-Landing-based XSEDE Stampede2 system were used to measure the floating-point operations (FLOP) and data transfers. The counters were read through the Intel Advisor XE (O'Leary *et al.*, 2017), which provides the related statistic of flops/byte or the arithmetic intensity. Only the steps of the method as described in section 4.3.2 were tested. Each measurement was for a single evaluation of the right-hand side of the equation 4.13, and no time integration was performed. A node on Intel-Knights-Landing-based Stampede2(Towns *et al.*, 2014) has 68 64-bit



(a) 2*D*.

(b) 3*D*.

Figure 4.12: Measured versus predicted arithmetic intensity on cubical sub-tiles.

cores. The L2 cache is 1 MB per two-core tile, with a Multi-Channel Dynamic Random Access Memory (MCDRAM) operating as 16*GB* direct-mapped L3. Each core has 4 hardware threads, and slow DDR4 memory of 96*GB* per node.

A 3D single-component test problem is conducted, where the data are initialized with random numbers (Loffeld & Hittinger, 2019). A single thread measures the arithmetic intensities for a single tile. The sizes of the tiles are 4, 16, 32, and 64 in length on a side. Since the cost of the flux function is ignored in the theoretical models, a no-op flux function is used, which makes no changes to the inputs (a no-op function). Recall that the flux function is assumed to operate on the input data in-place, so there is no additional data transfer cost for the flux function if the data is read from the cache. A no-op function is similarly used for the Riemann solver. Since there is no flux function or Riemann solver, the cost of the method is wholly independent of the content of the data.

Figure 4.12 shows the measured arithmetic intensity of the numerical test. The measured number of bytes moved is always greater than the predicted values by a fixed constant. As a result the measured arithmetic intensity is lower than the predicted arithmetic intensity. This overhead is more evident for the 2*D* case at small tile sizes, because the amount of data moved on small problems is modest enough that the a small number of cache line overhead becomes of size similar to the predicted total data movement. However, for the $3D$ problems the data movement remains large compared to the overhead, even when the length per side is small.

4.6 Conclusions

For scientific calculations, it is challenging to efficiently utilize the full floating-point capability of a machine. On-node performance of an application needs to emphasized along with parallel scalability for optimal algorithm design. With data motion increasingly becoming the dominant cost, algorithms that fail to make efficient use of every byte transferred will likely be bandwidth-limited. The algorithmic intensity of a method should therefore be an important design criterion.

A class of recovery-assisted discontinuous Galerkin (RADG) methods, used for spatial discretization of hyperbolic conservation laws, achieves higher arithmetic intensities. Estimates for three cache models are developed—the lower bound case of no-cache, an ideal case of infinite-size cache, and a finite-sized cache. Theory and numerical experiments suggest that RADG methods achieve high arithmetic intensity, which is necessary to better utilize on-node floating-point capability of modern HPC systems.

The present study is limited to DG approximation of smooth solutions and does not consider the nonlinear treatment necessary to capture discontinuities such as in the case of shocks. Solution limiting performs in-place operations on data already loaded in the cache which is expected to contribute to increase in arithmetic intensity. In addition, applications are mapped onto number of nodes, and data transfer between nodes will add overheads. Finally in this work it was shown that high algorithmic intensity can be achieved but requires hand optimization of code. In practice, many applications are written for modularity and maintainability, which poses challenge for optimization and necessitates further investigation.

CHAPTER 5

Enabling Power-Performance Balance with Transprecision Calculations for Extreme-Scale Computations of Turbulent Flows

This chapter is adapter from Tandon *et al.* (2020*c*). In modern scientific computing, the execution of floating-point operations emerges as a major contributor to the energy consumption of a compute-intensive applications with large dynamic range. Experimental evidence shows that over 50% of the energy consumed by a core and its data memory is related to floating-point computations. The adoption of floating-point formats requiring lesser number of bits is an interesting opportunity to reduce the energy consumption as it allows simplification of the arithmetic circuitry and reduces the memory bandwidth required to transfer the data between memory and registers. In theory, the adoption of multiple floating-point types following the principle of transprecision computing allows fine-grained control of floating-point arithmetic while meeting the desired standards on the accuracy of the final result. In this paper, the power-performance trade-offs for computing at different precision levels is analyzed for a parallel and distributed framework based on recovery-assisted discontinuous Galerkin (RADG) methods. The recovery operator of the RADG, operates on a compact support from neighboring elements and allows high-order approximation of the solution, with potential for massive parallelism. Using PoLiMEr – a power monitoring and management tool for HPC applications – fine-grained insights into the power characteristics of the RADG code on the supercomputer Theta at Argonne National Laboratory are presented. $3D$ benchmark tests indicate a savings of approximately 5 W per node with single precision computing. A mixed precision approach where all computations except recovery operation is performed in single precision shows promising results, however, an automated approach for tuning floating-point types and analyzing the floating-point sensitivity of variables and operations is desirable.

5.1 Introduction

In scientific computing today, most applications involving numerical computations with large dynamic range are typically performed using the double-precision (binary64) floating-point type, described by the IEEE 754 standard (Zuras *et al.*, 2008). In these applications, the execution of floating-point operations (FLOP) constitutes a major contribution to the energy consumption. Experimental investigation (Gautschi *et al.*, 2017) shows that more than 50% of the energy consumption for a floating-point-intensive application comes from the FLOP and moving the floating-point operands from data memory to registers and vice versa. Therefore, when such intensive calculations are performed at large-scale on current high-performance computing (HPC) centers, the power consumption and temperature control pose a developmental bottleneck (Deng *et al.*, 2013; Dongarra *et al.*, 2014); thus, power efficiency has garnered considerable concern in the supercomputing platform design and usage.

To advance science in the areas of climate science, ocean flow behavior, space sciences, biology, complex materials and others, the compute power of supercomputers must continue to grow (Brown *et al.*, 2010). The end of Dennard's scaling Dennard *et al.* (1974) has arrested increases in clock speed of processors and the increase in compute power was achieved by adding more computational units (CPUs). The CPU power consumption can be constrained by power capping (David *et al.*, 2010; Marincic *et al.*, 2017) to a value below the CPU Thermal Design Power (TDP) value, where TDP is the maximum amount of power that a node can draw. However, the increasing component count per node drives the energy consumption (Langer *et al.*, 2015), which suggests that scaling petascale systems require exorbitantly high power consumption (200 MW, Brown *et al.*, 2010). In addition, the size of memory on-chip has failed to keep up with the rising number of components, which has increased memory latency and the cost of data motion (Ashby *et al.*, 2010). Therefore, driven by these challenges, the next-generation exascale systems will operate under strict power budgets (20 MW, Lucas *et al.*, 2014), and are expected to feature heterogenous architectures. These architectural developments require new method development and redesign of existing algorithms.

Research on high-order methods (Tan *et al.*, 2005; Desjardins *et al.*, 2008; Bermejo-Moreno *et al.*, 2013; Colella *et al.*, 2011; Loffeld & Hittinger, 2019), which can achieve orders of accuracy > 2, shows that these methods have low discretization errors and can achieve desired accuracy with less number of data points, and therefore exhibit advantages over their low-order counterparts. This attribute of high-order methods makes them more desirable, especially for complex flow problems. High-order methods such as the discontinuous Galerkin (DG) method offer arbitrarily high orders of accuracy by adding degrees of freedom to each element to represent the solution as a high-order polynomial. Since the solution is allowed to be discontinuous across elements, the flux between immediately adjacent elements is used to exchange information, which preserves a compact stencil (Henry de Frahan, 2016). Thus, as demonstrated by Bey *et al.* (1995), Heinecke *et al.* (2014), Houba *et al.* (2019), and others, the DG method exhibits good parallel scaling on modern HPC platforms. The recovery-assisted DG method (RADG, Johnson & Johnsen, 2019) uses the information from a pair of neighboring elements to recover a high-order DG approximation of the flux at the interface of the two neighboring elements. The recovery procedure results in higher orders of accuracy than conventional DG methods. While the DG methods present many advantages, the large number of degrees of freedom inside each element, particularly for RADG methods, makes them floating-point-intensive.

To reduce the power cost of floating-point-intensive calculations, the number of precision bits in a floating-point representation can be reduced. Modern architectures offer wide range of floating-point types, from double precision (binary64) to binary8 (Gustafson & Yonemoto, 2017). To facilitate extreme-scale computations with RADG methods on emerging HPC platforms under restricted power budgets, this work analyzes the power-performance trade-offs when computing at double and single precision. To gain valuable, fine-grained insights into power consumption characteristics of RADG methods, an energy monitoring and power limiting interface for HPC applications, called PoLiMEr (Marincic *et al.*, 2017), is used. A brief discussion of the RADG methods and PoLiMEr is presented in the next section to highlight and identify the floating-point-intensive calculations. For this work, the $3D$ Taylor-Green Vortex (TGV) problem (Taylor & Green, 1937) is used as the benchmark test case. The test problem description and results of the numerical tests will be explained in section 5.3 and the conclusions are stated in section 5.4.

5.2 RADG framework and PoLiMEr interface

This section introduces the recovery-assisted discontinuous Galerkin (RADG) discretization for advection-diffusion systems of conservation laws and determines the floating-point intensity to estimate the code performance for single and double precision. In addition, the details of the power monitoring and management tool, PoLiMEr, is also provided.

5.2.1 Recovery-assisted discontinuous Galerkin (RADG) method

Consider the partial differential equation that describes an advection-diffusion system as:

$$\frac{\partial U}{\partial t} + \nabla . \mathcal{F}(U) - \nabla . \mathcal{G}(U, \nabla U) = 0, \tag{5.1}$$

where U is the vector of conservative variables, $\mathcal{F}$ is the convective flux and $\mathcal{G}$ is the viscous flux. The initial conditions for the system are defined as $U(\mathbf{x}, 0)$. The computational domain $\mathbf{\Omega}$, is partitioned into N_e non-overlapping elements Ω_m, such that $\mathbf{\Omega} = \cup_{m=1}^{N_e} \Omega_m$. The numerical solution U^h within each element is a $K-$dimensional polynomial expansion within the solution space ϕ_m:

$$U^h(\mathbf{x} \in \Omega_m) = U^h_m(\xi(\mathbf{x})) = \sum_{k=1}^{K} \phi^k_m(\xi) \hat{U}^k_m \tag{5.2}$$

where $\xi(\mathbf{x})$ is a mapping from the physical coordinate $\mathbf{x}$ to the reference coordinate ξ on the reference element, Ω_{ref}. The DG solution is defined by the K degrees of freedom (DOF) in each element, denoted as $\hat{U}_e$ in vector form. Each basis function has compact support, meaning that each member of ϕ_m is nonzero only on Ω_m. The governing equation (5.1) is satisfied in the weak form over for every element element by integrating against the basis functions:

$$\int\limits_{\Omega_m} \phi_m^k \frac{\partial}{\partial t} U_m^h d\mathbf{x} = \int\limits_{\Omega_m} \phi_m^k \nabla.\mathcal{G}(U^h, \nabla U^h) d\mathbf{x} - \int\limits_{\Omega_m} \phi_m^k \nabla.\mathcal{F}(U^h) d\mathbf{x}, \quad \forall k \in \{1, 2, \ldots, K\}. \tag{5.3}$$

Then, to allow communication between the neighboring elements and share common flux values along the interfaces of an element, integration by parts is performed. These common fluxes are denoted $\widetilde{\mathcal{F}}$ and $\widetilde{\mathcal{G}}$ and must be multiplied with the element's outward normal, n^-, to complete the DG weak form

$$\begin{aligned}\int\limits_{\Omega_m} \phi_m^k \frac{\partial}{\partial t} U_m^h d\mathbf{x} = \int\limits_{\partial\Omega_m} \phi_m^k (\widetilde{\mathcal{G}}.n^-) ds - \int\limits_{\partial\Omega_m} \phi_m^k (\widetilde{\mathcal{F}}.n^-) ds \\ - \int\limits_{\Omega_m} (\nabla\phi_m^k).(\mathcal{G}(U_m^h, \nabla U_m^h) - \mathcal{F}(U_m^h)) d\mathbf{x}, \quad \forall k \in \{1, 2, \ldots, K\}.\end{aligned} \tag{5.4}$$

Over the interior of each Ω_m, the DG solution U_m^h is applied directly to calculate the advective flux $\mathcal{F}$. For the diffusive flux $\mathcal{G}$, the gradient is approximated by an auxiliary variable σ (Johnson & Johnsen, 2019).

Equation (5.4) can be conveniently re-written, where the term on the left-hand side is expanded by using the definition in equation (5.2) which gives a $K \times K$ mass matrix (M_m) of the element Ω_m that depends only on the element's geometry and set of basis functions ϕ_m. Additionally, the spatial residual terms for a given element can be collected in a single residual vector (R_m). The mass matrix can be inverted to directly calculate the temporal derivatives of the DOF vector:

$$M_m^{row,col} = \int\limits_{\Omega_m} \phi_m^{row}\phi_m^{col} d\mathbf{x}, \tag{5.5a}$$

$$\begin{aligned} R_m^{row}(U^h) = \int\limits_{\partial\Omega_m} \phi_m^k(\widetilde{\mathcal{G}}.n^-)ds - \int\limits_{\partial\Omega_m} \phi_m^k(\widetilde{\mathcal{F}}.n^-)ds \\ - \int\limits_{\Omega_m} (\nabla\phi_m^k).(\mathcal{G}(U_m^h, \nabla U_m^h) - \mathcal{F}(U_m^h))d\mathbf{x}, \end{aligned} \tag{5.5b}$$

$$\frac{\partial}{\partial t}\hat{U}_m = M_m^{-1}R_m(U^h). \tag{5.5c}$$

In practice, the DG spatial discretization is typically paired with an explicit time integration method (such as four-stage, fourth-order Runge-Kutta) to integrate forward in time from some initial condition $U(\mathbf{x}, 0)$. For example, if applying the forward Euler method, then $\hat{U}_m(t + \Delta t) = \hat{U}_m(t) + \Delta t.M_m^{-1}R_m(U^h(t))$ for each element Ω_m, where the timestep size Δt must be small enough to maintain numerical stability.

Now, consider a union of two adjacent elements, $\mathcal{U} =\Omega_A \ \cup \ \Omega_B$ with same polynomial order p. For each union, a recovery coordinate $\mathbf{r}$ must be defined; the origin of this coordinate is the interface centroid. A recovery basis ψ, supported over U, is defined with the recovery coordinate $\mathbf{r}$. For this work, Legendre basis for ψ are used. The recovered solution over the union is defined as a $2K-$dimensional polynomial expansion in the recovery basis:

$$f(\mathbf{r}) = \sum_{n=1}^{2K} \Psi^n(\mathbf{r})\hat{f}^n \tag{5.6}$$

where the $2K$ coefficients in the recovered solution require $2K$ constraints. Now, let W be some arbitrary variable approximated in the DG solution space over Ω_A and Ω_B (for example $W = U^h$). The recovered solution is required to be weakly equivalent to W_A^h and W_B^h over $\mathcal{U}$ with respect to the K DG basis functions of Ω_A and the K DG basis functions of Ω_B.

$$\int\limits_{\Omega_A} \phi_A^k f \, d\mathbf{x} = \int\limits_{\Omega_A} \phi_A^k W_A^h \, d\mathbf{x} \quad \forall k \in \{1, 2, \ldots, K\}, \tag{5.7a}$$

$$\int\limits_{\Omega_B} \phi_B^k f \, d\mathbf{x} = \int\limits_{\Omega_B} \phi_B^k W_B^h \, d\mathbf{x} \quad \forall k \in \{1, 2, \ldots, K\}. \tag{5.7b}$$

This system can be recast in matrix-vector form (Johnson & Johnsen, 2019), where the coefficients $\hat{\mathbf{f}}$ are calculated directly from a combined coefficient vector, $[\hat{\mathbf{W}}_{\mathbf{A}}; \hat{\mathbf{W}}_{\mathbf{B}};]$, using a single matrix-vector multiplication. The recovery procedure is used exclusively to calculate the interface quantity $f(0)$ given the coefficients $\hat{\mathbf{W}}$ of Ω_A and Ω_B. The entire process is encapsulated in a single operator: $\mathcal{R}(W_A^h, W_B^h) = f(0)$.

5.2.2 PoLiMEr: Power monitoring and management for HPC applications

PoLiMEr is a C/C++ library for monitoring and managing power consumption of HPC applications (Marincic *et al.*, 2017), enabling scientific computing developers to assess the power and energy costs with minimal code instrumentation and tie these measurements to application-specific events. Its power monitoring capability enables users to obtain: 1) time-series data for power and energy over application runtime as well as user-specified code blocks, and 2) aggregate power and energy consumption summaries for the whole application and specific code sections. For the time-series data, power and energy are measured according to a user-configurable interval with 200 ms being the default setting. PoLiMEr's power management capability enables users to limit power consumption to achieve power or energy savings. Power can be capped on user-specified application phases, which is especially useful during memory or IO-heavy phases. Power caps can be applied on any code block, and can be set to any hardware-supported value.

Power and energy measurements are collected per each hardware-supported power domain a distributed MPI application occupies, and the readings are available per domain. For instance, on the Theta supercomputer, a CPU has one power domain, and one compute node has one CPU. If RADG is run on 128 nodes, the aforementioned power consumption data are reported per each node, regardless of the number of MPI ranks per node. Similarly, when power caps are set, they are set per power domain. Users can also specify if power should be capped on specific domains (CPUs) only. PoLiMEr assigns one rank from the application per power domain for the monitoring and power capping tasks.

On HPC systems with Intel CPUs, such as Theta, on which the power consumption of RADG is characterized, energy consumption is exposed through the file system as Model Specific Registers (MSR). Reading and writing to MSRs requires elevated user privileges, but supercomputing facilities can allow for safe access via the msr-safe module (Shoga *et al.*, 2014). Intel's Running Average Power Limit (RAPL) interface ensures power caps set through PoLiMEr are respected. RAPL maintains a moving average of the long-term power cap over a long-term window of time, allowing for brief violations up to the short-term power cap over a short-term time window depending on CPU utilization (David *et al.*, 2010). PoLiMEr allows users to set the long- and short-term power caps and time windows. From empirical evidence, setting both long- and short-term power limit to a value P results in measured power consumption being below P. This guarantees that the power budget is strictly obeyed. However, the system will not be enabled to utilize all of P power. On the other hand, setting the long-term power cap only sets power consumption to P with occasional brief violations.

Using PoLiMEr's fine-grained power and energy monitoring capabilities, the impact of mixed precision in the RADG code on its power and energy consumption is evaluated.

5.2.3 Floating-point operation (FLOP) count

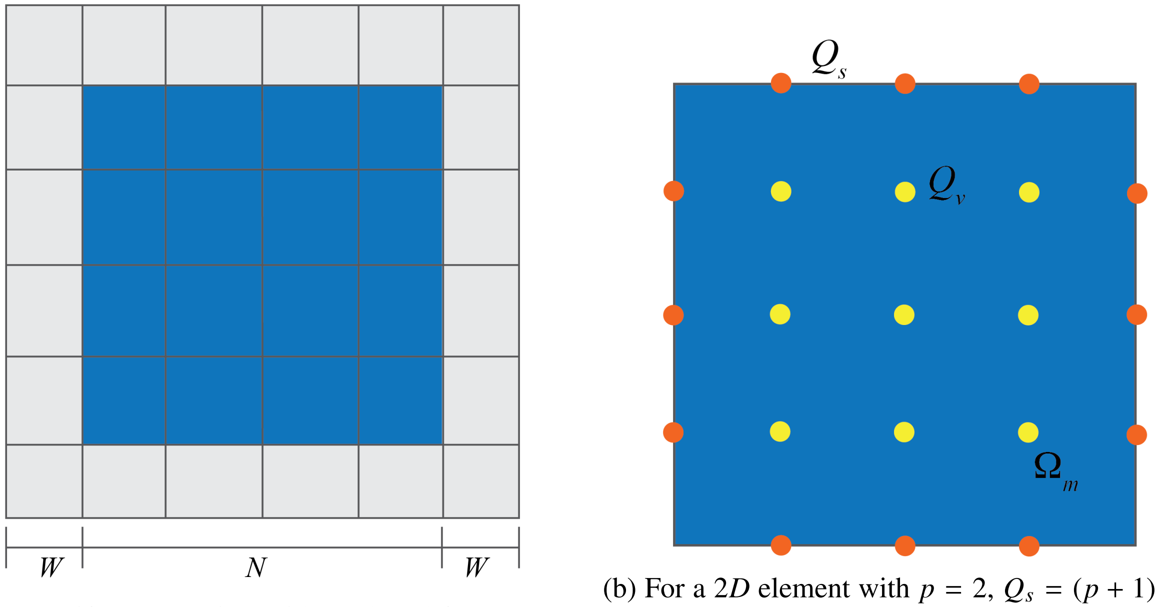
From the discussion in chapter 4, it is understood that the RADG discretization is floating-point-intensive. Following a similar approach, the FLOP count estimate for an advection-diffusion system can be obtained from the equations 5.2, 5.5, 5.6 and evaluation of the discrete recovery operator $\mathcal{R}$ (see section 4.3). Consider a regular, uniform cartesian grid, where the $1D$ element is a straight

#	Calculation	FLOP
1	$\widetilde{U}$, at interface quad. points	$2(Q_s \times 4K) \times N_v$
2	$\widetilde{\mathcal{F}}$, at interface quad. points	$(f_{R_f} \times Q_s) \times N_v$
3	U^h, at interior quad. points	$(Q_V \times 2K) \times N^D$
4	$\mathcal{F}(U^h)$, at interior quad. points	$(f_F \times D \times Q_v) \times N^D$
5	σ, at interior quad. points	$((D \times Q_v \times 2K) + (2D^2 \times Q_v \times Q_s)) \times N^D$
6	$\mathcal{G}(\sigma)$, at interior quad. points	$(f_G \times D \times Q_v) \times N^D$
7	$\widetilde{\sigma}$, at interface quad. points	$(D \times Q_s \times 4K) \times N_v$
8	$\mathcal{G}(\widetilde{\sigma})$, at interface quad. points	$(f_{R_g} \times Q_s) \times N_v$

Table 5.1: Floating-point operations for each step of the residual for advection-diffusion systems. Here, $K = (p+1)^D$, $Q_V = K$ is number of volume quadrature points, $Q_s = (p+1)^{D-1}$ is number of quadrature points on each interface, $N_v = D(N+2W+1)(N+2W)^{D-1}$ is the total number of interfaces, D is number of spatial dimensions.

edge; a $2D$ element is a quadrilateral, or square; and a $3D$ element is a hexahedral, or cube. Figure 5.1 shows an illustration of a $2D$ structured grid with quadrature points on a $2D$ element. The recovery operator in RADG requires information from nearest-neighbor elements, which ensures a compact stencil for RADG methods. Therefore, for a structured mesh with N elements in each direction, RADG methods need only one ghost cell on either side ($W = 1$) resulting in a total of $N + 2W = N + 2$ elements in each direction. Additionally, there are $(N + 2W + 1)$ planes per direction, and each plane has $(N + 2W)^{D-1}$ faces, which gives the total number of interfaces in D directions as $N_v = D(N + 2W + 1)(N + 2W)^{D-1}$. For a given polynomial order p, the number of degrees of freedom within each element is $K = (p + 1)^D$, the number of interior quadrature points is $Q_v = K$, and the number of quadrature points on each interface is $Q_s = (p + 1)^{D-1}$. Table 5.1 provides a rough estimation of the FLOP count for each step of the residual evaluation. Note that the evaluation of surface and volume fluxes in step 2, 4, 6, and 8, is problem dependent. Therefore, their costs are labelled f_{R_f}, f_F, f_G and f_{R_g}.

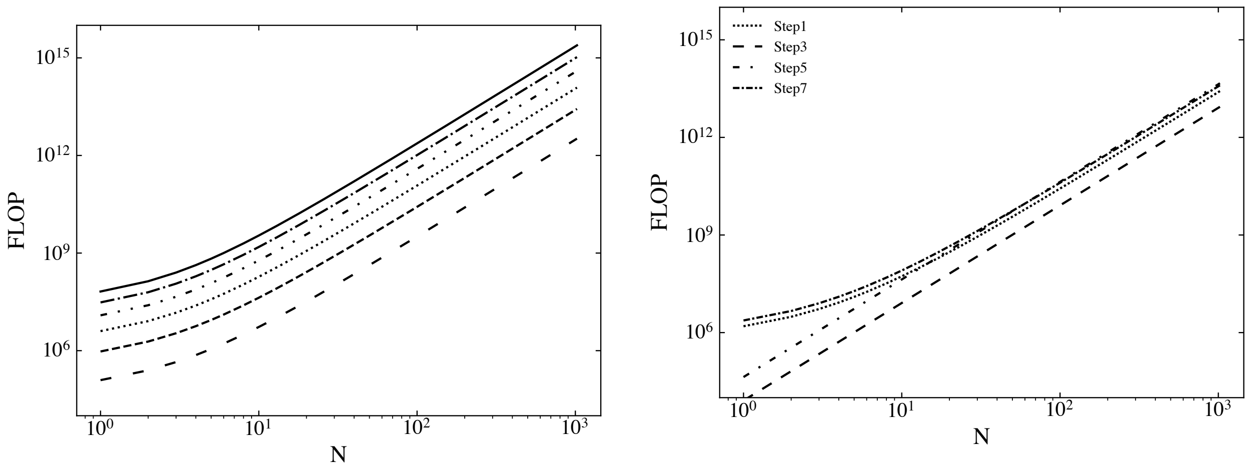
Figure 5.2 shows that the FLOP count for $3D$ advection-diffusion systems of conservation laws, discretized by RADG methods, increases with increasing number of elements, which increases the total number of degrees of freedom and the associated arithmetic operations. For a given element count, a higher polynomial order p has a higher FLOP count, attaining as much as a



(a) $2D$ uniform cartesian grid employed for RADG methods.

(b) For a $2D$ element with $p = 2$, $Q_s = (p + 1) = 3$ for each interface, and $Q_v = (p + 1)^2 = 9$.

Figure 5.1: Illustration of a $2D$ uniform cartesian grid with ghost cells and a $2D$ element: (a) Blue elements constitute the domain of interest and the outer grey elements form the ghost cell region. For RADG, the ghost cell width is $W = 1$, giving a total of $(N + 2W)$ elements in each direction. (b) Illustration of a $2D$ element showing the interior (Q_v in yellow) and interface (Q_s in orange) quadrature points for $p = 2$.



(a) Increase in the FLOP count with problem size

(b) FLOP count for different steps of the residual evaluation for $p = 3$.

Figure 5.2: Floating-point operation count for RADG discretization of advection-diffusion systems with varying polynomial order:– –, $p = 1$; - - - -, $p = 2$; ········, $p = 3$; - · · -, $p = 4$; -·-·-, $p = 5$; ——, $p = 6$.

petaflop (10^{15}), as the degrees of freedom in each element depend on p. Major contributions to the FLOP count comes from Steps 1 and 7, as shown for $p = 3$, which indicates that the recovery procedure is floating-point intensive. Therefore, it is essential to investigate and understand the energy consumption of a RADG-based application and the implications of operating with different floating-point types.

5.3 Numerical tests

5.3.1 Preliminaries

Two numerical tests are considered to evaluate the floating-point operations (FLOP) count and the energy consumption of RADG schemes. The first test is scalar advection-diffusion in $1D$, which verifies the convergence rates of RADG schemes and analyses the FLOP count required to reach a desired error. The second problem is the Taylor-Green vortex (TGV) on a uniform mesh, simulated through discretization of the $3D$ compressible Navier-Stokes equations. The TGV flow tests the method's energy consumption when computed with different floating-point types. For all test cases, the explicit four-stage, fourth-order Runge-Kutta (RK) method for time marching is applied as benchmark to keep the order of accuracy of advection, diffusion and time marching are the same. Although other approaches, such as (Prince & Dormand, 1981) may provide higher accuracy, such higher-order explicit RK schemes are impractical due to memory requirements. The time step is determined based on the Courant-Friedrichs-Lewy (CFL) and Von Neumann number (VNN) constraints as described by Johnson & Johnsen (2019). Advective interface fluxes in $1D$ scalar advection-diffusion are implemented via the upwind flux and the SLAU2 Riemann solver (Kitamura & Shima, 2013) for the TGV case. Parallelization is achieved via MPI. The software Gmsh (Geuzaine & Remacle, 2009) is used to generate all meshes. The larger simulations in TGV case were conducted on Stampede2 of XSEDE (Towns *et al.*, 2014) and the power measurements are reported on the Theta supercomputer at Argonne National Laboratory —a Cray XC40 system ranked #28 by the Top500 project (top, 2019). Theta has 4392 single-socket compute nodes with second-generation 64-core Intel Xeon Phi 7230 CPUs. The base frequency of each node is 1.3 GHz with turbo frequency up to 1.5 GHz.

5.3.2 Test 1: Scalar advection-diffusion in 1D

The $1D$ scalar advection-diffusion equation is

$$\frac{\partial U}{\partial t} + \frac{\partial(\mathcal{F} - \mathcal{G})}{\partial x} = 0, \qquad \text{where } \mathcal{F} = aU \quad \text{and} \quad \mathcal{G} = \mu \frac{\partial U}{\partial x}. \tag{5.8}$$

This equation is discretized in $1D$ to form the linear system described in equation 5.7. Here $a = \pi$, $\mu = \pi/100$, the spatial domain is $x \in [0, 8\pi]$ and periodicity is enforced along the left and the right boundaries. Initial condition is $U(x, 0) = \sin(x)$, which gives the exact analytical solution $U(x, t) = \sin(x - at)e^{-\mu t}$. The solution is marched forward in time or two translational periods such that, $t_f = 16$.

Three polynomial orders p are considered with varying number of elements N. Table 5.2 lists the grid resolution and the number of degrees of freedom, which for $1D$ case is given as

p	**Number of elements** (N)	**Degrees of freedom (nDOF)**
1	{16, 32, 64, 128, 256, 512}	{32, 64, 128, 256, 512, 1024}
2	{10, 22, 42, 86, 170, 342}	{30, 66, 126, 258, 510, 1026}
3	{8, 16, 32, 64, 128, 256}	{32, 64, 128, 256, 512, 1024}

Table 5.2: Mesh resolutions for test case 1.

$nDOF = (p+1)N = KN$ (see, section 5.2). At t_f, the error is quantified in terms of the L_2 norm of the cell-average given as,

$$E_{CA} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\overline{U}_i^h - \overline{U}_i)^2} \tag{5.9}$$

where the DG and exact solutions in each element Ω_i are denoted as $\overline{U}_i^h$ and $\overline{U}_i$, respectively. Figure 5.3 shows the cell-average L_2 error. The rate of convergence of RADG schemes for advection-diffusion systems is $2p+2$ for odd p and $2p+1$ for even values of p. Therefore for $p = 1$ and $p = 2$ both, RADG exhibits 4^{th} order convergence (Johnson, 2019). For high p, the desired error can be reached with a lower mesh resolution. Consider the desired error tolerance to be on the order 10^{-6}, shown as a dotted line in figure 5.3. RADG scheme with $p = 1$ requires approximately 10^3 degrees of freedom to achieve the desired accuracy. However, for $p = 3$, the same error is achieved with approximately 10^2 degrees of freedom, which is an order of magnitude less than the $p = 1$ case.

Figure 5.4 shows the FLOP required to achieve the desired error tolerance for this $1D$ problem. The FLOP count is evaluated using the expressions given in table 5.1 for one residual update per time step. The vertical dotted line corresponds to the error tolerance of 10^{-6}. While the FLOP count per time step for this trivial $1D$ test case is not as high as the predictions for $3D$ problems in figure 5.2, higher order RADG schemes with $p = 3$ are predicted to require an order of magnitude fewer FLOP than $p = 1$ case to achieve the same level of accuracy.

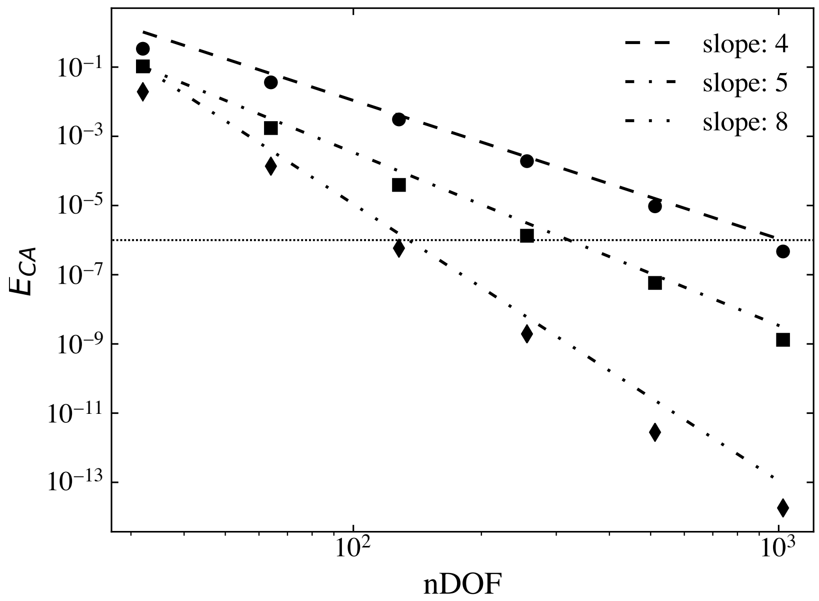


Figure 5.3: Convergence study in cell-average error for test case 1. Polynomial order:●, $p = 1$; ■, $p = 2$; ♦, $p = 3$.

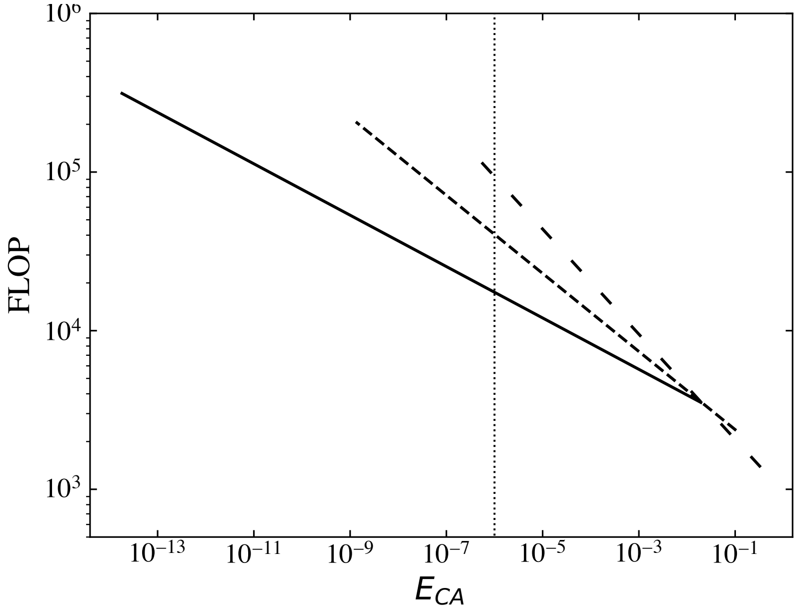


Figure 5.4: FLOP count required to achieve a particular accuracy on the 1D test problem with varying polynomial order:− −, $p = 1$; - - - -, $p = 2$;——, $p = 3$.

5.3.3 Test 2: Taylor-Green Vortex

The Taylor-Green Vortex (TGV, Taylor & Green, 1937) is a standard $3D$ benchmark problem (Chapelier *et al.*, 2014), where the flow exhibits transition from deterministic initial condition to anisotropic turbulence. The three-dimensional compressible Navier-Stokes equations are solved using the RADG method described in section 5.2.1. The initial conditions are characterized by velocity V_0, pressure p_0, and density ρ_0:

$$u = V_0 \sin(x/L)\cos(y/L)\cos(z/L), \tag{5.10a}$$

$$v = -V_0 cos(x/L)\sin(y/L)\cos(z/L), \tag{5.10b}$$

$$w = 0, \tag{5.10c}$$

$$p = p_0 + (\rho_0 V_0^2/16)[\cos(2x/L) + \cos(2y/L)][\cos(2z/L) + 2]. \tag{5.10d}$$

The specific flow parameters in the study are $L = 1$ m, $V_0 = 1$ m/s and $\rho_0 = 1$ kg/m^3 with constant viscosity set to $\mu = 0.000625$ kg/ms such that Reynolds number with respect to the characteristic length L is $Re = \rho_0 V_0 L/\mu = 1600$. The computational domain is $-\pi L \leq x, y, z \leq \pi L$ and the boundary conditions are all spatially periodic. The remaining fluid parameters are $\gamma = 1.4$, $R_g = 273.15$, and $Pr = 0.71$. The reference pressure p_0 is set such that V_0 corresponds to a Mach number of 0.1, thus $p_0 = \frac{\rho_0}{\gamma}(10V_0)^2$. Time is non-dimensionalized by $t_c = L/V_0$.

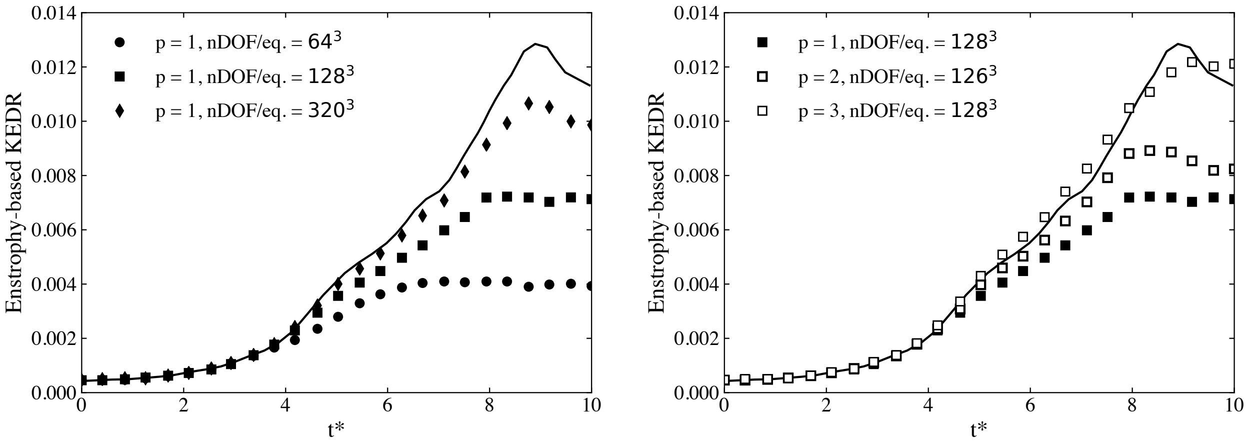
The enstrophy-based kinetic energy dissipation rate (KDER)

$$\epsilon = 2\varepsilon\frac{\mu}{\rho_0} \qquad \text{where} \quad \varepsilon = \frac{1}{\rho_0 V}\int_\Omega \frac{\rho}{2}(\omega.\omega)d\mathbf{x}. \tag{5.11}$$

is the metric considered to assess the method. The reference solution is taken from a pseudospectral code (Carton de Wiart *et al.*, 2014) using 512^3 degrees of freedom per equation. Figure 5.5 shows the enstrophy-based KEDR versus time for different grid resolutions considered (see table 5.3). As the number of computational elements are increases (holding the order constant with $p = 1$, the

p	**Number of elements per direction** (N)	**Degrees of freedom (nDOF)**
1	{32, 64, 128}	{64^3, 128^3, 256^3}
2	{21, 42, 85}	{63^3, 126^3, 255^3}
3	{16, 32, 64}	{64^3, 128^3, 256^3}

Table 5.3: Mesh resolutions for test case 2.



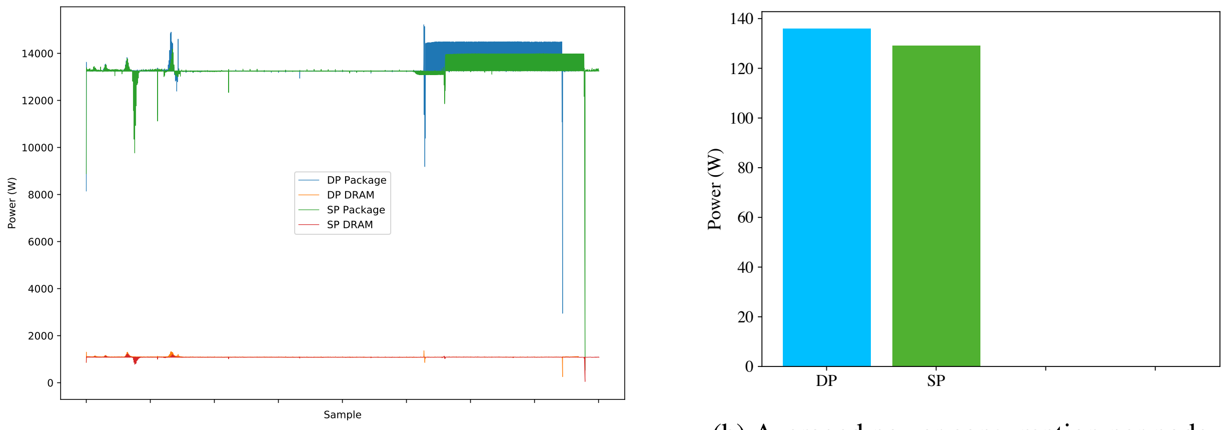
(a) Improving the solution accuracy by refining the mesh, $t* = (t/t_c)$.

(b) Improving the solution accuracy by increasing the polynomial order p, $t* = (t/t_c)$.

Figure 5.5: Enstrophy-based KEDR versus time for test case 2. ——, reference solution.

numerical solution converges to the reference. Similarly, increasing the order p while keeping the number of degrees of freedom constant also reduces the error. For $p = 3$ case, the use SLAU2 flux results in unreasonable results at later times, due to polynomial aliasing errors associated with the nonlinearity of the compressible Navier-Stokes equations. To improve stability a more dissipative flux can be used or the number of quadrature points can be increased for overintegration. These techniques were not explored in the present study. However, the increase in accuracy with increasing p comes at the cost of large FLOP. Referring back to figure 5.2, a spatial resolution of 128^3 for $p = 3$ accounts for FLOP count on the orders of teraflop (10^{12}) per residual evaluation.

Benchmark tests 1 and 2, both verify the floating-point intensive nature of RADG, which is more prominent in higher dimensions even at moderately high orders of p. Therefore, it can be expected that large-scale simulations of complex turbulent flows that are of practical importance will be more computationally demanding. This necessitates investigations to understand energy



(b) Averaged power consumption per node.

(a) A sample power trace for all 128 nodes.

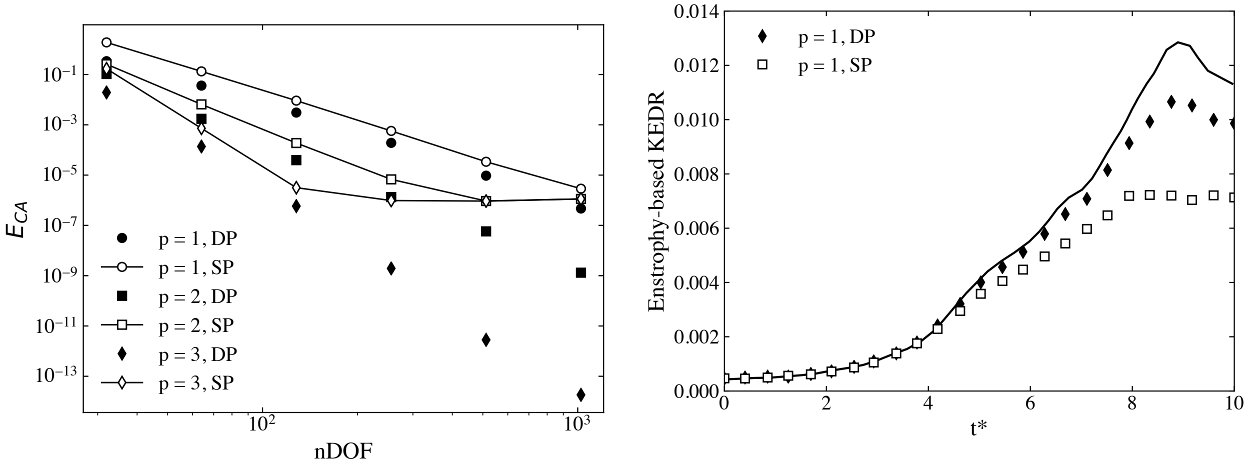
Figure 5.6: Overall power consumption measured with PoLiMEr for test case 2 (TGV) with single precision (SP) and double precision (DP) calculations using a RADG-based application.

consumption of a RADG-based application.

5.3.4 Power Management with PoLiMEr

To get an understanding of the energy consumption by a RADG-based application on an HPC platform, this study uses PoLiMEr (Marincic *et al.*, 2017), a power monitoring and management tool. An in-house C++ code implements the RADG framework described in section 5.2, which features a parallel and distributed framework through the use of MPI and CUDA programming environments. The details of parallel scaling of this application are provided in Appendix E.

Benchmark test case 2 (TGV) is simulated on ALCF's Theta supercomputer (top, 2019), with a conservative selection of $p = 1$ and $N = 160$ in each direction, resulting in total number of degrees of freedom to be 320^3. The benchmark test uses a total of 128 nodes with 64 MPI ranks on each node and the power trace is collected with PoLiMEr. Figure 5.6 shows the power trace sampled by PoLiMEr accounting the long initialization process with few peaks, followed by the time-marching iterations, where most of the work is done. The standard configuration with double precision (DP) draws approximately 136 W per node. The power consumed is less than the maximum allowed thermal design power (TDP) of 215 W on Theta, which is because the problem size is sufficiently



(a) Convergence of cell-average error for test case 1.

(b) Enstrophy-based KEDR vs time for test case 2, $t* = (t/t_c)$.

Figure 5.7: Comparing the performance with single precision (SP) calculations of RADG methods.

big enough to utilize all the resources on the 128 nodes allocated for the run. Modern HPC systems such as Theta have a machine balance of around 10 and generally require applications to have high arithmetic intensity to utilize the floating-point capability of the node (see, section 4.4). When the problem size 320^3 is decomposed on to 128 nodes, cores on a given node have approximately 16 degrees of freedom (8 elements), which is not sufficient to completely utilize the machine. The underutilization of resources results in low FLOP count, which cannot hide communication latency, thereby resulting in large overheads. Therefore, it can be expected that the increase in the problem size will increase the power consumption on the node.

In the scenario where power consumption is high, a natural way to lower the workload and the associated energy consumption is by computing at lesser precision levels. The power trace shown in figure 5.6 also sampled a single precision (SP) configuration, which reduces the overall power consumption with savings of approximately 5 W per node. However, this is indicative of the fact that changing the precision level affects the energy consumption.

Figure 5.7 shows the implications of computing at lower precision levels for both test cases. For $1D$ scalar advection-diffusion test, computations at single precision (SP) show a slower convergence compared to their higher precision counterpart. At higher orders of p the error saturates at approximately 10^{-7} which is the minimum possible achievable for single precision floating-point type (Kaneko & Liu, 1973; Haidar *et al.*, 2017). For $p = 3$, at larger $nDOF$ the error appears to increase. This is typical of low-precision computations (Lindstrom, 2014), where the small errors accumulate over the iterations and degrade the performance eventually causing the application to fail. In the TGV case, the single precision run, evaluated on the same configuration ($p = 1$, $nDOF = 320^3$) as the double precision setup, has a lower accuracy. Therefore, a power-performance balance is desired to maintain the desired levels of accuracy while keeping the energy consumption in control.

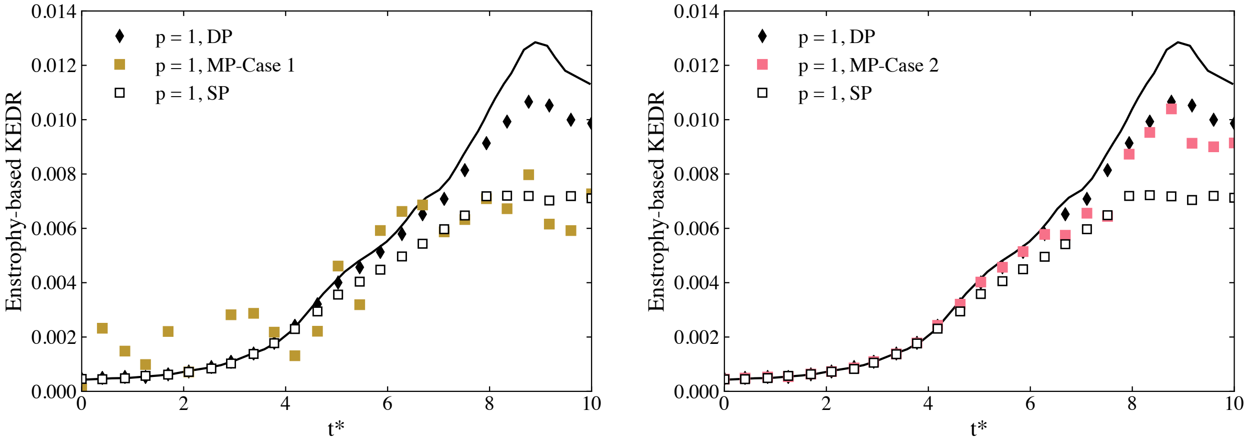
5.3.5 Transprecision compute framework for RADG methods

From the formulation of the RADG methods described in section 5.2, two optimization strategies are considered to assess mixed precision computing: (i) Case 1: Since the DG solution within each element Ω_e is defined as a polynomial expansion of the coefficients $\hat{U}_e$, store and compute $\hat{U}_e$ at higher precision and all other computations at lower precision, (ii) Case 2: the recovery operations in RADG build a high-order approximation of the solution, such that all computations except recovery are performed at low precision.

Figure 5.8 shows the performance of the two mixed precision models. For case 1, the DG coefficients are stored and computed in double precision. The solution accuracy is suboptimal to the single precision configuration and produces erroneous results. This is attributed to the fact that multiple casting of coefficients to a lower precision level incurs higher truncation errors which reduces the performance. On the other hand, the performance of case 2 shows significant improvements over the single precision configuration. Since, case 2 employs recovery operation at higher precision levels, the approximations near interfaces have increased accuracy that in turn enables more accurate approximations of volumetric fluxes than case 1.

5.3.5.1 Power Management with PoLiMEr

The effect of mixed precision computation on the overall power consumption is shown in figure 5.9. As before, PoLiMEr is used to sample the power trace of case 1 and 2 configurations of mixed



(a) Solution accuracy for test case 2, with case 1 mixed precision configuration, $t* = (t/t_c)$.

(b) Solution accuracy for test case 2, with case 2 mixed precision configuration, $t* = (t/t_c)$.

Figure 5.8: Comparing the performance with mixed precision (MP) calculations of RADG methods.

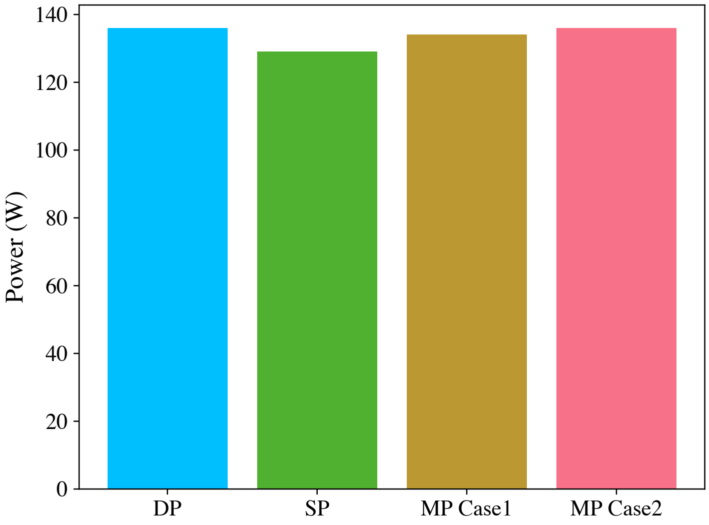


Figure 5.9: Overall power consumption measured with PoLiMEr for mixed precision (MP) calculations on RADG.

precision on ALCF's Theta supercomputer using the same benchmark of TGV (see section 5.3.4). The power consumption by the two mixed precision models is on par with the double precision configuration of the application. Possible reasons for this may include the larger overheads of computing at lower precision levels on a 64-bit architecture machine (Purkayastha *et al.*, 2004). In addition, the recovery operation is floating-point intensive and as discussed in chapter 4, has high cache memory requirements, which increases the number of data transfers needed, thus consuming more energy. Note, that the power trace analysis for all the configurations was averaged over 10 runs. In each run the single and double precision configurations were executed on the same resources (same nodes of the HPC). Since the mixed precision analysis was performed afterwards, the execution on a separate node may also add to the discrepancies observed.

The power-performance evaluation in case 1 and 2, suggests that manual optimization of applications, which is tedious, requires fine-grained tuning. Since emerging architectures are expected to have heterogeneous nodes, optimization for one HPC platform may not always be portable to other environments. To exploit the available floating-point types offered by the underlying hardware, many studies have started adopting a more automated optimization strategy (Lam *et al.*, 2013; Rubio-González *et al.*, 2013; Panchekha *et al.*, 2015; Bao & Zhang, 2013). A promising approach is the use of algorithmic differentiation (Naumann, 2012), which numerically computes a derivative of a computer program and can be used to study the floating-point sensitivity of variables and operations in a program. Algorithmic Differentiation Applied to Precision Tuning (ADAPT, Menon *et al.*, 2018) is C++ library, which enables precision tuning for scientific applications, thus enabling transprecision computing. The integration of ADAPT with the RADG-based C++ application will be considered in the future studies, and it is expected that a power-aware compute framework can be designed for RADG methods that can enable extreme-scale simulations of complex turbulent flow problems on the next-generation HPC platforms.

5.4 Conclusions

Recovery-assisted discontinuous Galerkin (RADG) methods are highly scalable, and the compact support of the discontinuous solution on nearest neighbors, supports large-scale simulations on modern HPC platforms. However, RADG discretizations of advection-diffusion problems are floating-point intensive, achieving up to a petaflop (10^{15}) for a $3D$ benchmark test of Taylor-Green Vortex. The energy footprint of RADG methods is assessed by integrating with a power measurement and management library, PoLiMEr. Switching from double to single precision evaluation of the benchmark test case on 128 nodes on Cray XC40, Theta, results in savings of approximately 5 W per node at the cost of loss of solution accuracy. A mixed precision configuration, where all operations except recovery were performed at single precision, shows significant improvement in solution accuracy on the $3D$ benchmark test in comparison to the single precision configuration. However, manual optimization for floating-point types is tedious and achieving power-performance balance is not trivial. Therefore, an automated approach for tuning floating-point types and analyzing the floating-point sensitivity of variables and operations is desirable.

CHAPTER 6

Conclusions

6.1 Summary

This dissertation is focussed on large-scale simulations of complex turbulent flows in the context of modulation of turbulent boundary layer separation and optimization of a new class of recovery-assisted discontinuous Galerkin methods for next-generation HPC platforms. Turbulent boundary layer flow separation occurs in regions of adverse pressure gradient, e.g., flow over an aircraft wing at high angle-of-attack, flow near the rear end of a road vehicle at highways speeds, atmospheric flow around ground structures, and in many other engineering applications. Flow separation can give rise to undesirable effects such as loss of lift on aircraft wings, or increase in drag of road vehicles. Passive flow control strategies, such as vortex generators (VGs), have been used in many engineering applications to delay and reduce the size of the separation region. However, an understanding of the interactions between the VG-induced flow structures and those of the separated region is missing.

In part I of this thesis, wall-resolved large-eddy simulation of a model problem of flow over a backward-facing ramp is studied with a submerged, wall-mounted cube used as a canonical VG. Numerical simulations are conducted using the open source OpenFOAM (Weller *et al.*, 1998) libraries, whose numerical framework and discretization schemes are validated and verified (Tandon *et al.*, 2017) against the canonical problem of turbulent flow over a backward-facing step (Le *et al.*, 1997). The focus of this study is to elucidate the effects of the VG configuration, namely its height, location, and the spacing between neighboring VGs in a line array, on the interaction between the

VG-induced flow structures and the separated region, and the resulting turbulent transport which reduces flow separation. For this purpose, three sets of studies are considered at a Reynolds number of 19,600 based on the boundary layer thickness, and classified under two categories:

- Single VG studies (Tandon *et al.*, 2017, 2018, 2020*a*) to understand the effect of VG height and its location on the flow modulation. A single wall-mounted cube is placed upstream of the leading ramp edge along the plane of symmetry. For a fixed cube position, the cube height is varied as h/δ_0 = 0.2, 0.6 and 1.0, and for a fixed cube height, the upstream location of the cube is varied as x_{vg}/h = 0, 3 and 6.
- Multiple VG study (Tandon *et al.*, 2019, 2020*b*), with an array of equally-spaced, wall-mounted cubes, to understand the dependence of flow modulation on the spanwise spacing between neighboring VGs. For a fixed height $h/\delta_0 = 0.6$ and upstream location $x_{vg}/h = 3$ the spanwise spacing between neighboring cubes in an array is varied as L_z/h = 3, 5 and 7.

The numerical results demonstrate the dependence of the turbulence transport mechanism on the intensity of interaction between the horseshoe vortex system of the wall-mounted cube with the separated region, which in turn depends on the VG configuration. The evolution of turbulent kinetic energy in the expansion section is analyzed to understand the contributions to the production and transfer of energy for different VG configurations to derive a better understanding of the flow modulation mechanism.

Numerical simulations of complex turbulent flows, e.g., wall-bounded turbulent boundary layer flow, require high spatial and temporal resolution to capture the unsteady flow dynamics accurately. The cost of computation grows with the Reynolds number, which characterizes range of relevant scales in the flow problem. Since many practical flow problems of interest operate at high Reynolds number regime, numerical simulations of such complex systems are infeasible, even on the largest supercomputers, due to the large range of scales that need to be resolved. The need for high accuracy with low discretization errors and the evolving heterogeneous architecture of the next-generation high-performance computing centers has impelled interest in the development of high-order methods. While the new class of recovery-assisted discontinuous Galerkin (RADG) methods (Johnson, 2019) can provide arbitrarily high-orders of accuracy, the large number of degrees of freedom increases the costs associated with the arithmetic operations performed and the amount of data transferred on-node. The focus of the part II of this thesis is to improve the parallel efficiency of RADG methods for modern high-performance computing (HPC) architectures. First, the on-node performance of RADG methods is assessed by analyzing different cache memory models. With a finite-sized cache, use of a derivative-based recovery (Johnson, 2019), and an optimized data-tiling strategy, RADG methods show significant improvements to the arithmetic intensity (Tandon & Johnsen, 2020), which is necessary to make better utilization of on-node floating-point capability. Second, the power-performance balance of RADG methods is evaluated for computing floating-point values with double and single precision. Analysis of power-performance trade-offs suggests savings in power consumption when operating at lower precision, indicating that a transprecision framework will likely offer better power-performance balance on modern HPC platforms (Tandon *et al.*, 2020*c*).

6.2 Key findings and contributions

6.2.1 Part I: Modulation of turbulent boundary layer flow

Separation of turbulent boundary layer flow was reduced over the ramp surface when a wall-mounted cube was used as a passive vortex generator (Shinde *et al.*, 2016; Tandon *et al.*, 2017). Consistent with the previous studies on the turbulent flow around a cubic bluff body (Martinuzzi & Tropea, 1993; Krajnovic & Davidson, 2002; Shinde *et al.*, 2017; Shinde, 2018), a horseshoe vortex system was observed upstream of the wall-mounted cube. The horseshoe vortex extends in the near-wake of the cube to form a pair of streamwise vortices, which are weaker compared to the more aerodynamic vortex generators studied by Lin (2002), and are unable to modulate the flow at large streamwise distances. However, the simpler design of a wall-mounted cube ensures that the canonical flow study considered in this work has a reduced parameter space, such that the dependence of flow modulation on these parameters is studied with relative ease.

In Chapter 2, the single VG study demonstrates that the horseshoe vortex system interacts and entrains the hairpin vortices in the separated region to form a counter-rotating vortex pair. The vortex pair entrains high momentum fluid from freestream towards the near-wall region, thereby energizing the boundary layer and reducing the separation of flow (Tandon *et al.*, 2018). The evolution of the turbulent kinetic energy (TKE) is studied in the expansion section of the backward-facing ramp, which indicates that the high vorticity centers of the counter-rotating vortex pair correspond to regions of TKE production. The analysis of the Reynolds stress distribution along the spanwise flow direction confirms the transport of momentum by the counter-rotating flow towards the plane of symmetry and the near-wall regions, thereby reducing flow separation. The size and location of the wall-mounted cube affects the turbulent transport mechanism and, thus, the modulation of flow separation (Tandon *et al.*, 2020*a*). The size of the horseshoe vortex system increases with the size of the cube, which enhances the production and transfer of TKE to the near-wall region. When the cube height increases from $h/\delta_0 = 0.6$ to 1.0, the volume of separation reduces by 40%, however, the coefficient of drag for the cube increases by 25%. Change in the upstream location of the cube also influences the behavior of the horseshoe vortex system such that, when the cube is placed far upstream, the horseshoe vortex must traverse larger streamwise distance under the influence of near-wall diffusion. The TKE produced by the horseshoe vortex is diffuses and dissipates close to the wall, which leads to a dispersed core with low efficiency to modulate the separated region. However, when the cube is placed too close to the leading ramp edge, the high energy-producing structures formed around the cube are not fully developed and are subjected to high strain the expansion section. The turbulent structures decay, and the transfer of TKE is inadequate to modulate the separated region. For a cube of height $h/\delta_0 = 0.6$, the upstream location of $x_{vg}/h = 3$ offers the most optimal configuration with high reductions in the volume of separation, and the separation length along the plane of symmetry.

Deriving from the findings in the single VG case, Chapter 3 studies the modulation of flow separation over the backward-facing ramp by an array of equally-spaced, wall-mounted cubes

(Tandon *et al.*, 2019). The array of cubes of height $h/\delta_0 = 0.6$, is placed $x_{vg}/h = 3$ upstream of the leading ramp edge. Analysis of the flow behavior shows that the size horseshoe vortex system depends only on the height of the cube, however, its lateral spreading is dictated by the spanwise spacing between the neighboring cubes of the array, which is consistent with the previous studies of flow around an array of wall-mounted cubes by Shinde (2018), or multiple cylindrical VGs studied by Pujals *et al.* (2010). When the spanwise spacing is too low, the interaction of horseshoe vortices of adjacent cubes produces larger TKE with a wall-normal ejection of low-momentum fluid that produces secondary turbulent flow (Yang *et al.*, 2019). However, the compact turbulent structures are subjected to high strain in the expansion section and an increased diffusion due to near-wall effects, which reduces their efficiency of flow modulation. When the spanwise spacing is larger than $L_z/h \geq 5$, the interaction between adjacent horseshoe vortices drastically diminishes, and the flow behavior is similar to that of the single, isolated VG (Tandon *et al.*, 2020*b*). This behavior occurs because the counter-rotating flow imparts spanwise motion that brings the two legs of the horseshoe vortex system closer to one another and thus, restricts it lateral spreading. Previous studies by Krajnovic & Davidson (2002) and Hwang & Yang (2004) have shown that $5h$ downstream of the cube, the lateral spreading of the horseshoe vortex system is restricted to $7h$. Therefore, $L_z/h = 7$ offers the most optimal reduction in the size of the separated region over the ramp—if the spanwise spacing is reduced, the interaction of vortical structures reduces the efficiency, and if the spacing is increased, the area under the influence is not maximized .

The canonical study of modulation of turbulent boundary layer flow in part I is limited to the investigation of a single cube and a single array of cubes. The VG geometry and interaction between VGs in different rows may affect the horseshoe vortices and thus affect the flow separation region in different ways. Future studies of these phenomena would improve our understanding of separation modulation using passive vortex generators.

6.2.2 Part II: Optimization of high-order discontinuous Galerkin method for next-generation HPC platforms

The parallel efficiency of the new class of recovery-assisted discontinuous Galerkin (RADG) methods recently proposed by Johnson (2019) is enhanced by analyzing data locality and cache tiling strategy, and the power-performance trade-offs when computing with different floating-point types.

In Chapter 4, RADG discretizations of hyperbolic systems of conservation laws are analyzed, and the steps in the residual update involve floating-point intensive computations. The intensity of arithmetic operations increases with the polynomial order p, which is expected as the number of degrees of freedom in each element increases with p (Johnson, 2019). Bounds on arithmetic intensity (Williams *et al.*, 2009), the ratio of total work done per data transferred, are theoretically obtained by considering three cache memory models. The no-cache and idealized infinite-size cache models provided the lower bound and ideal performance limits of RADG methods, respectively. The finite-size cache, with a cubical tiling strategy, achieves high arithmetic intensity of > 10, which is on par with the state-of-the-art machines (OLCF's Summit, YarKhan *et al.*, 2019). However, the cache space requirement and the data transfer overheads associated with decomposing data into cubic tiles are exorbitantly high due to the use of a discrete recovery operator. An alternative implementation of derivative-based recovery (Johnson, 2019) drastically relaxes the cache requirements (< 5 MB) and allows cubic tile lengths of up to $T = 16$ for $p = 3$ to be computed with relative high efficiency. In addition, a vertical tiling strategy, originally proposed by Loffeld & Hittinger (2019), is evaluated to show further improvements in achievable arithmetic intensity and reduction of cache size requirements. Theoretical estimates and supporting numerical tests demonstrate that RADG methods can achieve high arithmetic intensity and make better use of floating-point capabilities available on modern HPC platforms (Tandon & Johnsen, 2020).

Chapter 5 analyzes the power-performance balance of RADG methods for advection-diffusion systems of conservation laws. Following the approach of Chapter 4, the steps in the residual update for a RADG discretization are inspected to show that the steps involving the use of recovery operator are floating-point intensive. For a $3D$ problem on a uniform structured grid, the RADG methods attain up to a petaflop (10^{15}) per residual evaluation. To ensure operation within the prescribed power budgets, the associated energy footprint of RADG methods is assessed by integrating with a power measurement and management library, PoLiMEr Marincic *et al.* (2017). Switching from double to single precision evaluation of a benchmark test case on 128 nodes on Cray XC40, Theta (top, 2019), results in savings of 5 W per node. However, for single precision computations, the error convergence is slower and for higher polynomial order p, the error saturates at approximately 10^{-7}, which is the minimum possible value achievable for single precision floating-point type (Kaneko & Liu, 1973; Haidar *et al.*, 2017). A mixed precision configuration, where all operations except recovery were performed at single precision, shows significant improvement in solution accuracy on the $3D$ benchmark test in comparison to the single precision configuration. However, manual optimization for floating-point types is tedious and achieving power-performance balance is not trivial. Therefore, an automated way to tune for floating-point types and analyze the floating-point sensitivity of variables and operations is desired, which is where an automated transprecision compute framework will be beneficial.

Previous studies on optimization of DG-based methods for parallel computing include studies on mixed-precision algorithms (Chapelier *et al.*, 2014; Renac *et al.*, 2015), scalable implementation on accelerators (Chan *et al.*, 2016; Modave *et al.*, 2016; Henry de Frahan, 2016), many-core processors (Heinecke *et al.*, 2014; Müller *et al.*, 2019), improving parallel I/O routines (Rettenberger & Bader, 2015), and modifying the implementation of schemes (Fidkowski, 2019; Faghih-Naini *et al.*, 2020). Breuer *et al.* (2015) presented a framework that minimizes energy and time-to-solution by increasing the DG order from 2 to 7 while maintaining double-precision accuracy. The present studies are the first to approach on-node performance optimization and a power-aware transprecision compute framework for high-order recovery-assisted discontinuous Galerkin methods. The analyses is limited to DG for smooth solutions and does not consider the nonlinear treatment necessary to capture discontinuities, such as in the case of shocks. Solution limiting performs in-place operations on data already loaded in the cache, which is expected to contribute to an increase in the arithmetic intensity and energy consumption due to increased arithmetic operations. Solution limiting is known to impact the convergence rate, and therefore the implementation of a transprecision framework will require special care. In addition, applications in practice are mapped onto a number of nodes, and data transfer between nodes will add overheads not considered in chapter 4. Finally, in this work, it was shown that high algorithmic intensity and power-performance balance can be achieved but requires hand optimization of code. In practice, many applications are written for modularity and maintainability, which poses a challenge for optimization and necessitates further investigation.

6.3 Recommendations for future work

Large-scale simulations of complex turbulent flows is an active area of research and calls for investigations in model development, computer hardware, application, and system software, and other related areas. The work presented here can be extended in a number of directions and the following topics are suggested for future studies.

6.3.1 Modulation of flow separation on backward-facing ramp by multiple cube arrays

A direct extension of the canonical study presented in part I is the modulation of flow over a backward-facing ramp by multiple arrays of cubes placed either inline or in a staggered arrangement. Yang *et al.* (2019) studied turbulent flow over a sparse arrangement of cubes and found that for lower surface coverage densities, with appropriate streamwise spacing, the secondary turbulent vortices formed above spanwise-heterogenous roughness redistributes the fluid momentum in the outer layer, leading to high-momentum pathways above the wall-mounted cubes and low-momentum pathways at the two sides of the wall-mounted cubes, which increases the coefficient of drag of the cubes. Similar work on turbulent boundary layer flow over cube roughened walls by Lee *et al.* (2011) showed the dominance of hairpin vortices in the outer boundary layer. Therefore, the arrangement of the multiple rows of cube arrays will affect the behavior of the turbulent boundary layer, thereby affecting the modulation of flow separation over the ramp. The proposed study can be also thought of as an optimization problem, with the goal of reducing flow separation by considering different arrangements of multiple rows of VGs. Furthermore, uncertainty quantification and machine learning techniques can also be employed in the optimization problem.

6.3.2 Modulation of flow separation at high Reynolds number

In many applications of interest, the Reynolds number of the flow is much higher than the direct numerical studies of canonical problems considered (Smits *et al.*, 2011). For example, the friction Reynolds number of a Boeing 747 aircraft is roughly estimated to be $Re\tau = 10^5$ under a typical cruising condition (Iwamoto *et al.*, 2005). For flows of such high Reynolds numbers, where highly complex turbulent structures exist with a very wide range of turbulent spectra, quantitative knowledge of flow separation and its modulation is required. While the advances in computer hardware and distributed computing continue to provide the compute power needed to approximate complex physical systems, the development of high-order numerical methods and algorithms is also desired (Pirozzoli, 2011). High-order methods based on discontinuous Galerkin approach offer the advantage of arbitrary high-orders of accuracy and high-scalability, which makes them a better candidate than the high-order finite difference and finite volume schemes, both of which rely on large stencil size. The development of DG-based schemes for advection-diffusion systems is an ongoing area of research (Henry de Frahan, 2016; Johnson & Johnsen, 2019; Halila *et al.*, 2019), and can facilitate simulations of complex turbulent flows.

6.3.3 High-fidelity large-eddy simulations with recovery-assisted discontinuous Galerkin methods

Developing large-eddy simulations (LES) algorithms based of discontinuous Galerkin (DG) methods is an active area of research. To achieve high accuracy and low discretization errors, LES approaches demand high resolution. Since DG methods offer the advantage of arbitrarily high accuracy on relatively coarser meshes, they are interesting avenues for LES. Some notable work in the area revolves around implicit LES (ILES, Uranga *et al.*, 2011; Frere *et al.*, 2015; Renac *et al.*, 2015; de Wiart & Hillewaert, 2015; Fernandez *et al.*, 2017), where the numerical dissipation of the discretization scheme to account for the dissipation that takes place in the unresolved scales. ILES benefits from its easy implementation without a subgrid stress model and currently gains considerable attention from researchers in the computational fluid dynamics community. Other areas of research are directed towards new model development for explicit LES. Recent study by Parish (2018) uses the Mori-Zwanzig approach to decompose the discrete unknowns into a coarse-scale resolved set and a fine-scale unresolved set for DG methods and facilitates modeling of under-resolved simulations of turbulent flow. Recovery-assisted DG methods (RADG, Johnson & Johnsen, 2019) have higher accuracy for advection-diffusion systems than traditional DG methods, and therefore, can enable high-fidelity LES of complex turbulent flows.

6.3.4 Performance portability of applications

The next-generation high-performance computing (HPC) systems (at exascale and beyond), are expected to have heterogeneous architectures (Ashby *et al.*, 2010; Brown *et al.*, 2010; Lucas *et al.*, 2014). This poses a challenge for scientific applications to maintain and demonstrate similar levels of efficiency across different architectures. The fine-grain optimization of applications comes at a cost of loss in generality and modularity. Therefore, current trends in computing algorithms is attracting research in programming models that can maintain performance portability while incurring minimal overheads (Heroux *et al.*, 2020). Performance portable programming libraries such as Kokkos (Edwards *et al.*, 2014) and RAJA (Beckingsale *et al.*, 2019) offer a high-level abstraction with multiple backend support for thread-level parallelism on host (processors/CPU) and device (accelerators/GPU). Figure 6.1 shows implementation of thread-level parallelism using Kokkos for a model problem of scalar advection, discretized with the recovery-assisted discontinuous Galerkin (RADG) method. As observed, Kokkos allows building an application with multiple thread-level

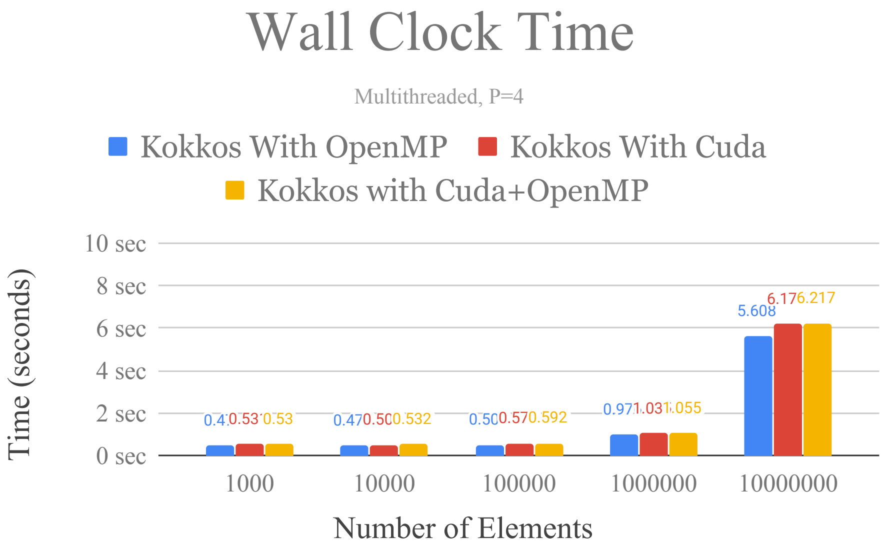


Figure 6.1: Implementation of thread-level parallelism with Kokkos showing performance portability on three different architectures with minimal overheads.

support—OpenMP, CUDA, etc. Depending on where the application is launched (host or device), Kokkos abstraction takes care of the associated data coalescing and results in improved performance portability, with minimal overheads. It is believed that the on-node performance of RADG methods can be further improved by invoking thread level parallelism.

6.3.5 Recovery for fault-resilience

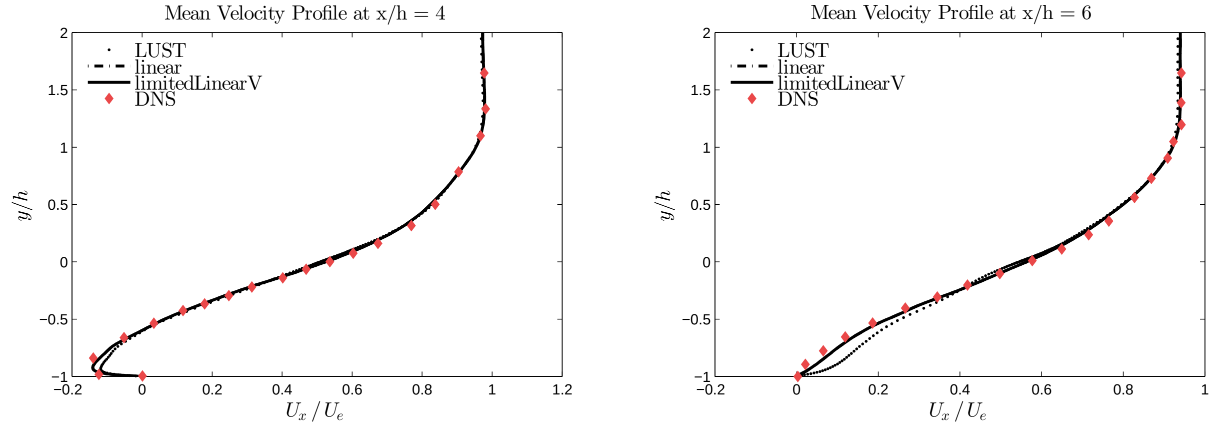
Fault resilience is a major roadblock for High-Performance Computing (HPC) executions on exascale machines. The increased component count requires dramatic architectural changes and new programming practices. The occurrence of random faults from the failure of hardware or software malfunction raises the concern that simulations have to either stopped or that the data is missing in certain regions (subdomains), which will render the results erroneous (Schroeder & Gibson, 2007; Cappello *et al.*, 2014). While many advancements are being made to redesign the hardware capability, new algorithmic solutions need to be employed in user applications that enable fault-tolerance Brown *et al.* (2010); Lucas *et al.* (2014). Traditional checkpointing methods El-Sayed & Schroeder (2013); Gainaru *et al.* (2013); Das *et al.* (2017), that capture a redundant image of the solution and roll back all processors to a previously saved state and restart computations to roll forward, will carry a lot of overhead in terms of memory buffer required and the restart times of all processors at exascale. Other approaches based on estimation theories and interpolation methods Lee *et al.* (2017) need auxiliary data in the form of a low-resolution solution state or some analytical result available for the problem. Access to auxiliary data may not be possible for complex flows, and the reconstruction using low-resolution results can itself return polluted final results.

With these issues in mind, in theory the recovery operation in recovery-assisted discontinuous Galerkin (RADG) methods, can be thought of as a tool for diagnosis. In the scenario of a node failure, the data from the elements that interfaces the failed node can be used to recover solution at the interface of failed node, as the recovery technique uses the data from the neighboring elements to recover the flux by matching moments of the approximate solution over both the adjacent cells. This two-element union procedure for recovery can then be iteratively applied to the interior elements of the failed node. In the scenario of missing data, recovery can be used as a "detector", such that if the the DG polynomials and the recovered solution do not agree in a pointwise fashion over a two-element union, it would be apparent that the numerical approximation is erroneous. Hence, it may be plausible to build a fault-resilient framework with RADG methods.

Appendices

Appendix A

Validation of LES Model And Discretization Schemes



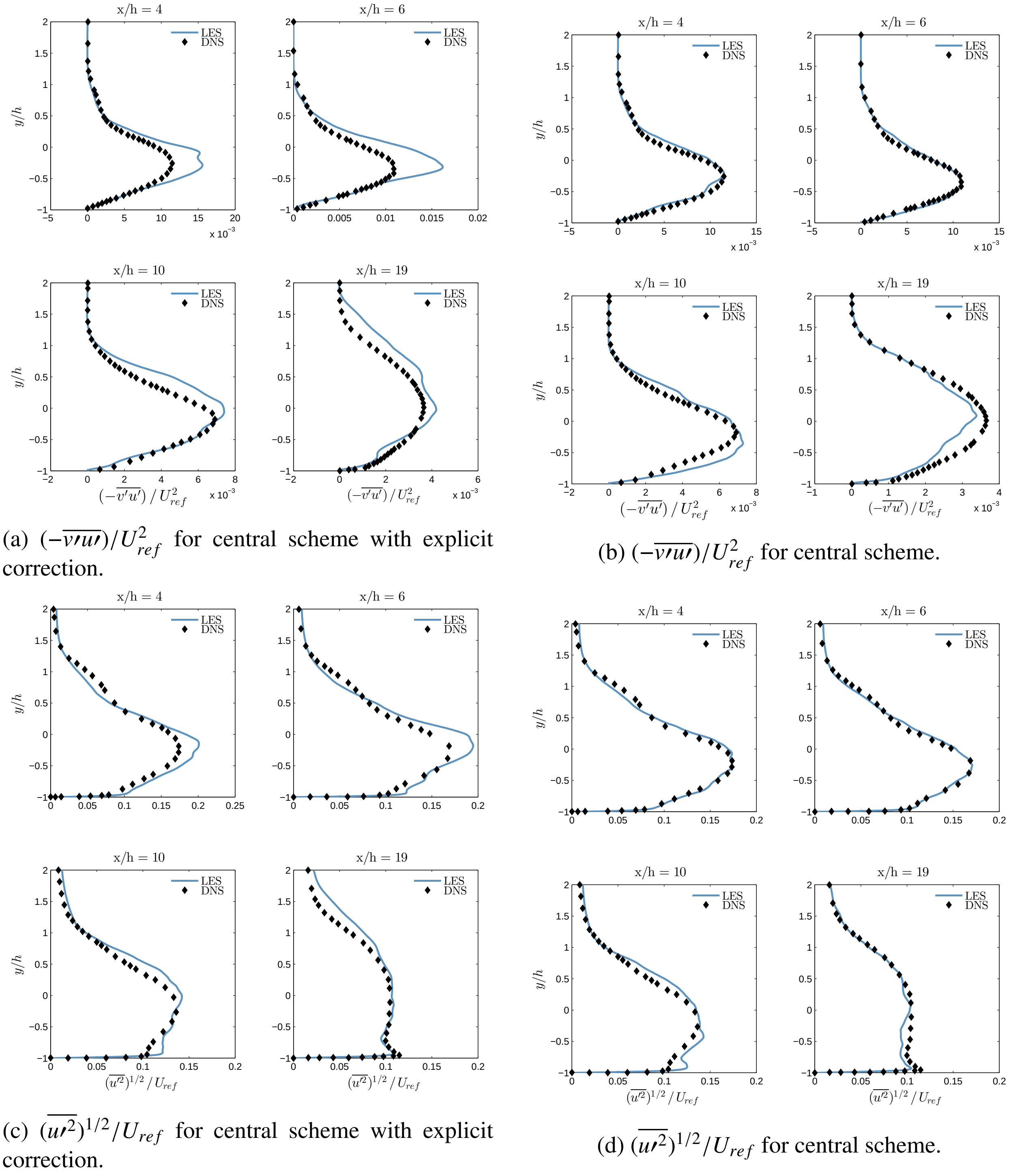
(a) $4H_{step}$ after the step.

(b) $6H_{step}$ after the step.

Figure A.1: Streamwise component of the mean velocity profile sampled at different stream wise locations for backward-facing step simulations with One-Equation Eddy Viscosity LES model and compared with DNS data of Le *et al.* (1997).

In order to choose a suitable large eddy simulation (LES) model in OpenFOAM® that predicts the separation with less numerical dissipation, a validation study is performed. A backward-facing step with similar set-up as the DNS study of Le *et al.* (1997) is simulated with one-equation eddy-viscosity LES model and tested with three different discretization schemes - a linear upwind stabilized scheme, a second-order central scheme and second-order central scheme with explicit correction.

The mean streamwise velocity profiles at different streamwise locations are shown in figure A.1. The mean flow behavior is similar in two of the three discretization schemes as the velocity profiles collapse on each other. In order to differentiate, turbulent statistics are shown in



(a) $(-\overline{v\prime u\prime})/U_{ref}^2$ for central scheme with explicit correction.

(b) $(-\overline{v\prime u\prime})/U_{ref}^2$ for central scheme.

(c) $(\overline{u\prime^2})^{1/2}/U_{ref}$ for central scheme with explicit correction.

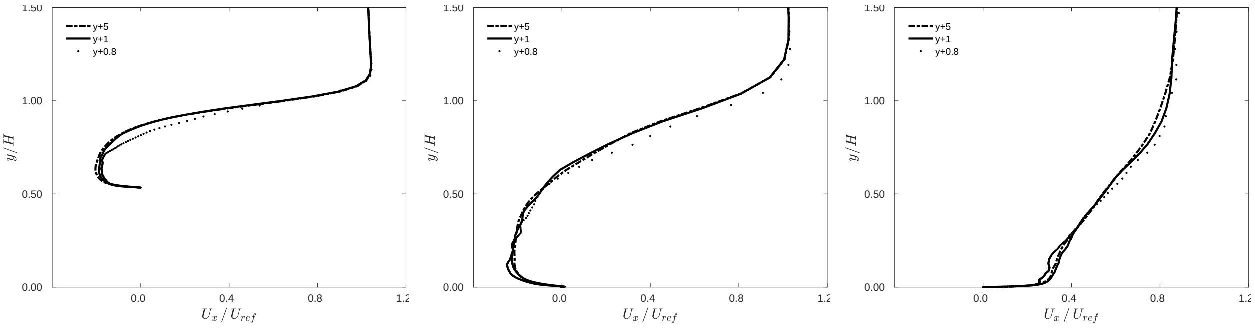
(d) $(\overline{u\prime^2})^{1/2}/U_{ref}$ for central scheme.

Figure A.2: Comparison of Reynolds stress profiles of backward-facing step simulations with One-Equation Eddy Viscosity LES model and various discretization schemes with DNS of Le *et al.* (1997).

figure A.2 for these schemes. We see that the central scheme with one-equation eddy viscosity LES model replicates the flow features and matches DNS data very well.

Appendix B

Grid refinement study of turbulent boundary layer flow over a backward-facing ramp



(a) At $x/H = 1$.

(b) At $x/H = 2$.

(c) At $x/H = 6$.

Figure B.1: Streamwise component of the mean velocity profile sampled at different streamwise locations along the plane of symmetry for turbulent flow over a backward-facing ramp.

Property	Coarse mesh	Medium mesh	Fine mesh
Total number of cells ($\times 10^6$)	19.29	23.21	27.16
Number of cells/h	60	100	120
Δ_y^+ in the refined region	5.0	1.0	0.8

Table B.1: Meshing information for turbulent flow over a backward-facing ramp.

To study grid refinement for turbulent flow over a backward-facing ramp (see Chapter 2), three grid resolutions are designed with different near-wall resolution of $\Delta_y^+ = 5.0$, 1.0 and 0.8 as listed in table B.1. The "+" indicates the dimensionless grid spacing in wall coordinates. $\Delta^+ = \frac{\Delta u_\tau}{\nu}$ where

Δ is the grid spacing in physical dimensions, u_τ is the friction velocity at the wall and ν is the kinematic viscosity of the fluid.

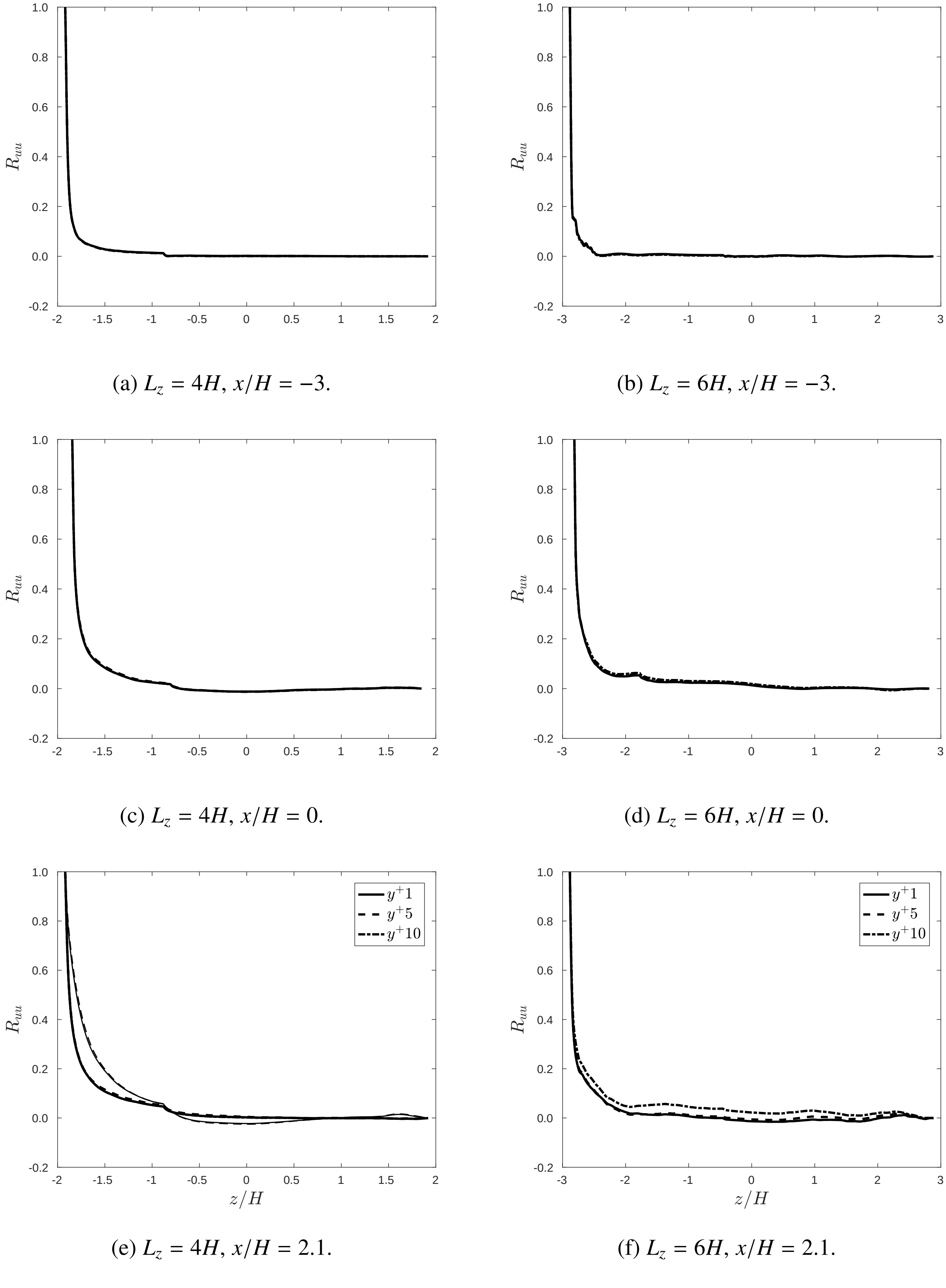
It is evident from figure B.1 that the results with the coarse mesh differ slightly from those obtained from the medium and fine meshes. The reason for this disagreement is the inadequate mesh resolution in the coarse mesh simulations near the wall. Especially after the flow separates at the ramp edge the generation of turbulent kinetic energy produces small scale flow structures which cannot be adequately captured by the coarse mesh. Therefore, to obtain mesh independent results it is necessary to have a grid refinement. We observe that the separation region over the ramp surface and the subsequent flow reattachment on the bottom wall is adequately captured by medium and fine grids. Especially the flow behavior near the wall surface is similar. Thus the medium mesh with wall-normal grid spacing of $\Delta_y^+ = 1$ is sufficient of our analysis of flow control.

Appendix C

Spatial two-point correlation study for domain size analysis of turbulent boundary layer flow over a backward-facing ramp with a wall-mounted cube

To determine the spanwise extent of the computational domain, we show the streamwise component of the spatial two-point correlations for two different doamin widths, $L_z = 6H$ and $L_z = 4H$, where H is height of the ramp. The spatial two-point correlations, R_{uu}, in Figure C.1 show the extent of the interaction of the streamwise fluctuations between the side lateral wall, located at $z/H = -3$ and $z/H = -2$ respectively, and the plane of symmetry which is fixed at $z/H = 0$ for both cases. Figure C.1 shows variation in R_{uu} along different streamwise stations depicted by x/H. In the upstream region of the cube at $x/H = -3$, Figures C.1a and C.1b, we see that the interaction regions near the lateral side walls for both the cases is smaller as compared to a ramp height. As one moves closer to the upper ramp edge at $x/H = 0$, the region of generation of the shear layer and separation bubble, the interaction region near the lateral side walls grows to the order of a ramp height for both domain sizes, see Figures C.1c and C.1d. This lateral side wall interaction, that occurs due to the slip boundary condition imposed on the lateral walls (Chapter 2), is constrained to a region of the order of a ramp height in the expansion region as well. Figures C.1c and C.1d show the interaction length at the bottom ramp edge located at $x/H = 2.1$.

Therefore, based on the spatial two-point correlations, as shown in Figure C.1, we conclude that the for domain size greater than $L_z = 4H$, the lateral side wall interactions are constrained to a zone that is of the order of the ramp height, H. This confirms that our domain is wide enough, so that the flow in the wake of the single VG is not affected by the flow behavior along the side walls. Also, note, that we cannot decrease the size of the domain to less than $4H$ and this can be attributed to the flow studies around a cube by Krajnovic & Davidson (2002). For a flow around a cube, the spanwise extent of the horse-shoe vortex is visible until $z/h = 7$, where h is height of the cube. For our largest cube height of $h/\delta_0 = 1.0$ in this study that corresponds to roughly $z/H = 2.8$ in terms of ramp height. Thus, having a domain width $L_z < 4H$ will pollute the results due to blockage effect and flow interactions from the lateral side walls.



(a) $L_z = 4H$, $x/H = -3$.

(b) $L_z = 6H$, $x/H = -3$.

(c) $L_z = 4H$, $x/H = 0$.

(d) $L_z = 6H$, $x/H = 0$.

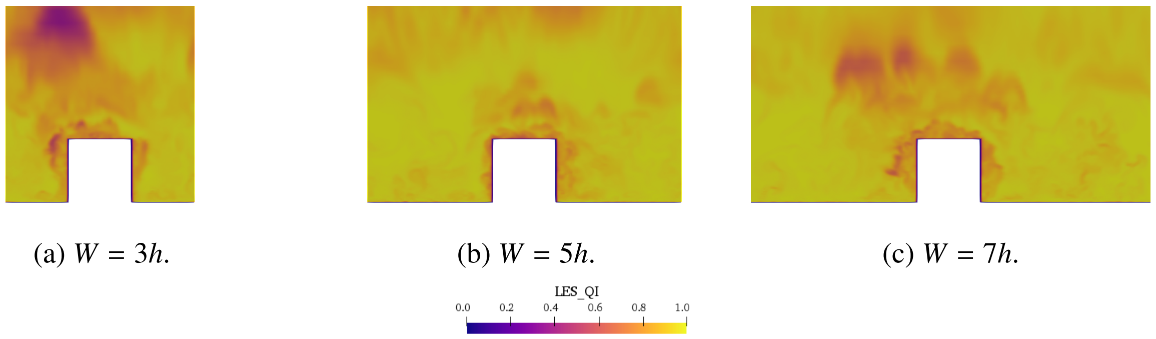
(e) $L_z = 4H$, $x/H = 2.1$.

(f) $L_z = 6H$, $x/H = 2.1$.

Figure C.1: Streamwise component of the spatial two-point correlations, R_{uu}, at different streamwise stations given by x/H for spanwise domain size analysis

Appendix D

The quality index of wall-resolved large eddy simulations


(a) $W = 3h$.

(b) $W = 5h$.

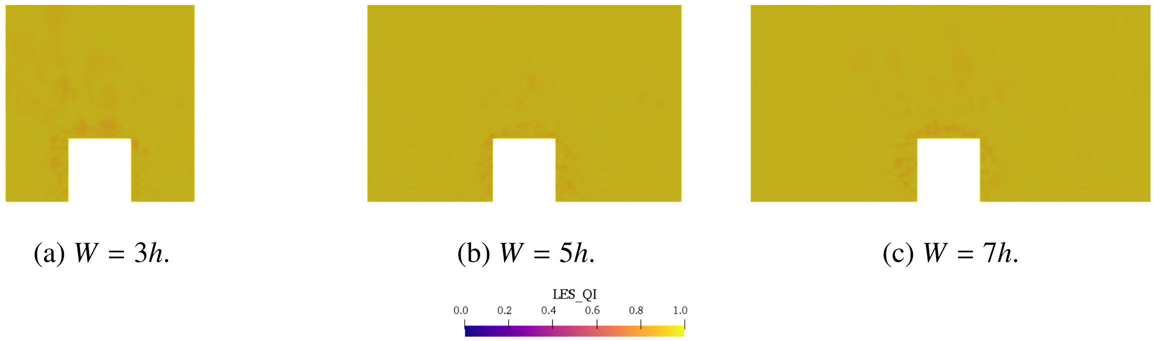
(c) $W = 7h$.

Figure D.1: LES quality index based on resolved kinetic energy (LES_QI_k)for the different inter-cube spacings.

Quality assessment in the LES is not easy similar to traditional RANS models because both the discretization errors and sub grid scale contribution to the model are proportional to the grid size (Celik *et al.*, 2005). Good LES simulation tends to DNS as finer grids are implemented, therefore there is not a grid independent result in traditional LES theory and it is necessary to have some quality assessments for every LES simulation. Pope (2000) suggests that 80% of the energy be resolved everywhere for LES with near-wall resolution. Therefore, the quality index for LES (LES_QI_k) based on the kinetic energy spectrum can be written as,

$$LES\_QI_k = \frac{k^{res}}{k^{tot}} = \frac{k^{res}}{k^{res} + k^{sgs}} \tag{D.1}$$

In this study the wall-resolved LES shows more than 80% of the TKE is resolved in the near-wall regions especially around the surface-mounted cube, see figure D.1. Another metric in the


(a) $W = 3h$.

(b) $W = 5h$.

(c) $W = 7h$.

Figure D.2: LES quality index based on the subgrid activity parameter (LES_QI_ν) for the different inter-cube spacings.

assessment of LES is the sub-grid activity parameter (Geurts & Fröhlich, 2002), which is defined as

$$s = \frac{<\varepsilon_t>}{<\varepsilon_t> + <\varepsilon_\mu>} \tag{D.2}$$

Here, <> denotes an averaged (or filtered) quantity, ε_t is the turbulent dissipation, and ε_μ is the molecular dissipation. It is stated that by definition $s = 1$ corresponds to LES, whereas $s = 0$ corresponds to DNS at infinite Reynolds number (Re). However, evaluation of s can be sometimes be tedious as it involves calculation of the volume-averaged turbulent dissipation rate, which inherently includes both the modeled dissipation and the numerical dissipation; segregation of the two is necessary but not easy. A modified parameter s^* (Celik *et al.*, 2005), can be evaluated with relative when incorporating many assumptions including the relation between turbulent dissipation rate and turbulent viscosity. Using the modified parameter, quality index for LES can be defined as,

$$LES\_QI_\nu = \frac{1}{1 + 0.05(\frac{\nu_t+\nu}{\nu})^{0.53}} \tag{D.3}$$

Here, nu is the molecular viscosity and ν_t is the turbulent viscosity. $LES\_QI_\nu > 0.8$ is considered good LES, while value of 0.95 and higher is considered as DNS. In figure D.2, the LES_QI_ν in this study remains between 0.84 and 0.92, which suggests a good wall-resolved LES study.

Appendix E

Scalability of RADG on Summit

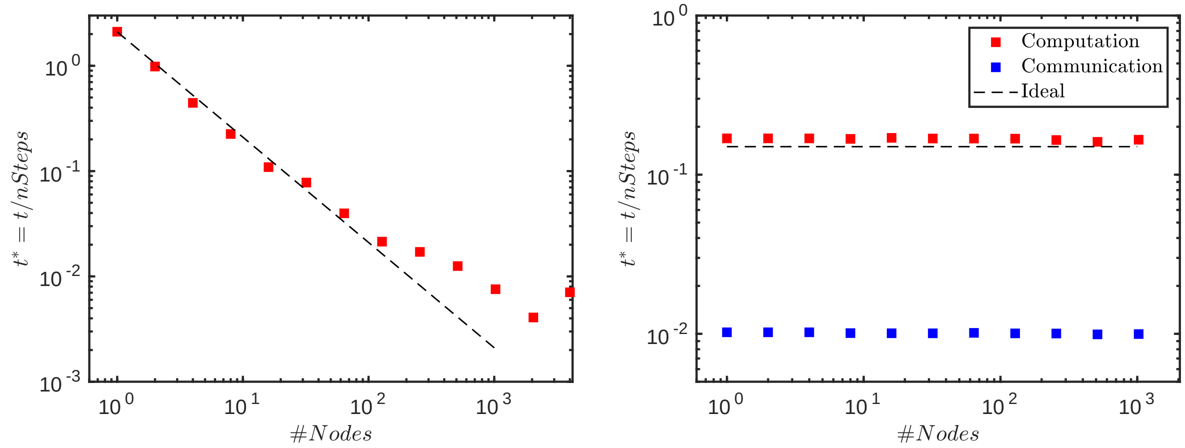


Figure E.1: Strong (left) and weak (right) scaling of RADG code on Summit, ORNL.

To perform extreme-scale computations, it is essential to assess the scalability of recovery-assisted discontinuous Galerkin (RADG) code on HPC platforms. The discontinuous solution approximation inside each element Ω_m and dependence on nearest neighbor (section 5.1) makes RADG methods highly scalable and easily parallelizable. The in-house RADG-based C++ code features many-core and multi-GPU compute framework to achieve massive concurrency by using MPI and NVIDIA's CUDA programming environment.

Figure (E.1) shows the scaling of the RADG code on Summit supercomputer on the NVIDIA Volta V100 accelerators. The scaling runs use a benchmark 3D problem of TGV with 160 elements in each direction and polynomial order of $p = 1$ for solution approximation, which gives the total number of degree of freedom (DOFs) above 32 million. The method scales up to 2K nodes on Summit, each node with 4 GPUs. Additionally, the weak scaling with 16K DOFs per node shows good performance up to 2K nodes. Hence, RADG has a high potential for massively-parallel simulations of turbulent flow problems on machines with an architecture similar to that of Summit.

BIBLIOGRAPHY

2019 Top500 list - november 2019. Accessed: 2019-07-20.

Adams, E. W. 1984 Experiments on the structure of turbulent reattaching flow. *PhDT* .

Adrian, R. J., Meinhart, C. D. & Tomkins, C. D. 2000 Vortex organization in the outer region of the turbulent boundary layer. *J. Fluid Mech.* **422**, 1–54.

Ahmed, S. R., Ramm, G. & Faltin, G. 1984 Some Salient Features Of The Time-Averaged Ground Vehicle Wake. In *International Congress & Exposition Detroit, Michigan*.

Angele, K. P. & Muhammad-Klingmann, B. 2005 The effect of streamwise vortices on the turbulence structure of a separating boundary layer. *European J. Mech. Bio Fluids* **24** (5), 539–554.

Armaly, B. F., Durst, F., Pereira, J. C. F. & Schönung, B. 1983 Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.* **127**, 473–496.

Ashby, S., Beckman, P., Chen, J., Colella, P., Collins, B., Crawford, D. & others 2010 The opportunities and challenges of exascale computing. *Tech. Rep.*. U.S. Department Of Energy, Office of Science, Advanced Scientific Computing Advisory Committee.

Ashill, P., Fulker, J. & Hackett, K. 2001 Research at dera on sub boundary layer vortex generators (sbvgs). In *39th AIAA Aerospace Sciences Meeting*, p. 887.

Ashill, P, Fulker, J & Hackett, K 2002 Studies of flows induced by sub boundary layer vortex generators (sbvgs). In *40th AIAA Aerospace Sciences Meeting*, p. 968.

Asselin, D. J. & Williamson, C. H.K. 2017 Influence of a wall on the three-dimensional dynamics of a vortex pair. *J. Fluid Mech.* **817**, 339–373.

Aubertine, C. D. & Eaton, J. K. 2005 Turbulence development in a non-equilibrium turbulent boundary layer with mild adverse pressure gradient. *J. Fluid Mech.* **532**, 345–364.

Bao, T. & Zhang, X. 2013 On-the-fly detection of instability problems in floating-point program execution. In *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications*, pp. 817–832.

Barnard, R. H. 2001 *Road vehicle aerodynamic design-an introduction*.

Barros, J. M. & Christensen, K. T. 2014 Observations of turbulent secondary flows in a rough-wall boundary layer. *J. Fluid Mech.* **748**.

Basu, P., Hall, M., Williams, S., Van S., B., Oliker, L. & Colella, P. 2015 Compiler-directed transformation for higher-order stencils. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pp. 313–323. IEEE.

Beckingsale, D. A., Burmark, J., Hornung, R., Jones, H., Killian, W. & others 2019 Raja: Portable performance for large-scale scientific applications. In *2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pp. 71–81. IEEE.

Bermejo-Moreno, I., Bodart, J., Larsson, J., Barney, B. M., Nichols, J. W. & Jones, S. 2013 Solving the compressible navier-stokes equations on up to 1.97 million cores and 4.1 trillion grid points. In *SC'13: Proceedings of Int. Conf. High Perf. Comp., Networking, Storage and Analysis*, pp. 1–10. IEEE.

Berselli, L. C., T., Iliescu & J., Layton W. 2005 *Mathematics of large eddy simulation of turbulent flows*. Springer-Verlag, Berlin: Scientific Computation.

Bey, K. S., Patra, A. & Oden, J. T. 1995 hp-version discontinuous galerkin methods for hyperbolic conservation laws: A parallel adaptive strategy. *Int. J. Num. Meth. Engg.* **38** (22), 3889–3908.

Bose, S. T., Moin, P. & You, D. 2010 Grid-independent large-eddy simulation using explicit filtering. *Phys. Fluids* **22** (10), 105103.

Bose, S. T. & Park, G. I. 2018 Wall-modeled large-eddy simulation for complex turbulent flows. *Ann. Rev. Fluid Mech.* **50**, 535–561.

Breuer, A., Heinecke, A., Rannabauer, L. & Bader, M. 2015 High-order ader-dg minimizes energy-and time-to-solution of seissol. In *International Conference on High Performance Computing*, pp. 340–357. Springer.

Bross, M., Fuchs, T. & Kähler, C.J. 2019 Interaction of coherent flow structures in adverse pressure gradient turbulent boundary layers. *J. Fluid Mech.* **873**, 287–321.

Brown, A. C., Nawrocki, H. F. & Paley, P. N. 1968 Subsonic diffusers designed integrally with vortex generators. *J. Aircraft* **5** (3), 221–229.

Brown, D, Messina, Paul, Keyes, D, Morrison, J, Lucas, R, Shalf, J, Beckman, P, Brightwell, R, Geist, A, Vetter, J & others 2010 Scientific grand challenges: Crosscutting technologies for computing at the exascale. *Tech. Rep.*. U.S. Department Of Energy, Pacific Northwest National Laboratory.

Brown, G. L. & Thomas, A. S. W. 1977 Large structure in a turbulent boundary layer. *Phys. Fluids* **20** (10), S243–S252.

Buttner, G. 2019 Airbus: Innovating the future flight. In *Swiss Conf. & HPCXXL User Group*. Lugano, Switzerland.

Calarese, W., Crisler, W. & Gustafson, G. 1985 Afterbody drag reduction by vortex generators. In *23rd AIAA Aerospace Sciences Meeting*, p. 354.

Cappello, F., Geist, A., Gropp, W., Kale, S. & Kramer, B. 2014 Toward exascale resilience: 2014 update. *Supercomputing Frontiers and Innovations* **1** (1), 4–27.

Caretto, L. S., Gosman, A. D., Patankar, S. V. & Spalding, D. B. 1973 Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pp. 60–68. Springer.

Castro, I. P. & Robins, A. G. 1977 The flow around a surface-mounted cube in uniform and turbulent streams. *J. Fluid Mech.* **79** (2), 307–335.

Celik, I. B., Cehreli, Z. N. & Yavuz, I. 2005 Index of resolution quality for large eddy simulations. *J. Fluids Engg.* **127** (5), 949–958.

Chan, J., Wang, Z., Modave, A., Remacle, J.-F. & Warburton, T. 2016 Gpu-accelerated discontinuous galerkin methods on hybrid meshes. *Journal of Computational Physics* **318**, 142–168.

Chapelier, J.-B., De La Llave Plata, M., Renac, F. & Lamballais, E. 2014 Evaluation of a high-order discontinuous galerkin method for the dns of turbulent flows. *Computers & Fluids* **95**, 210–226.

Cherry, E. M., Elkins, C. J. & Eaton, J. K. 2008 Geometric sensitivity of three-dimensional separated flows. *Int. J. Heat and Fluid Flow* **29** (3), 803–811.

Cockburn, B., Karniadakis, G. E. & Shu, C.-W. 2000 The development of discontinuous Galerkin methods. *Discontinuous Galerkin Methods* pp. 77–88.

Cockburn, B., Lin, S. Y. & Shu, C. W. 1989 TVB runge-kutta local projection discontinuous galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comp. Phys.* **84** (1), 90–113.

Colella, P., Dorr, M. R., Hittinger, J. A. F. & Martin, D. F. 2011 High-order, finite-volume methods in mapped coordinates. *J. Comp. Phys.* **230** (8), 2952–2976.

Corsiglia, V. R., Rossow, V. J. & Ciffone, D. L. 1976 Experimental study of the effect of span loading on aircraft wakes. *J. Aircraft* **13** (12), 968–973.

Das, A., Mueller, F., Hargrove, P. & Roman, E. 2017 Pin-pointing node failures in hpc systems. In *CEUR Workshop Proceedings*, , vol. 1828, pp. 53–59, arXiv: arXiv:1603.07016v1.

David, H., Gorbatov, E., Hanebutte, U. R., Khanna, R. & Le, C. 2010 Rapl: memory power estimation and capping. In *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pp. 189–194. IEEE.

Davidson, P. A. 2004 *Turbulence : an introduction for scientists and engineers*.

Dehtyriov, D., Hourigan, K. & Thompson, M. C. 2020 Direct numerical simulation of a counter-rotating vortex pair interacting with a wall. *J. Fluid Mech.* **884**.

Deng, Y., Zhang, P., Marques, C., Powell, R. & Zhang, L. 2013 Analysis of linpack and power efficiencies of the world's top500 supercomputers. *Parallel Computing* **39** (6-7), 271–279.

Dennard, R. H., Gaensslen, F. H., Rideout, V. L., Bassous, E. & LeBlanc, A. R. 1974 Design of ion-implanted mosfet's with very small physical dimensions. *IEEE J. Solid-State Cir.* **9** (5), 256–268.

Desjardins, O., Blanquart, G., Balarac, G. & Pitsch, H. 2008 High order conservative finite difference scheme for variable density low mach number turbulent flows. *J. Comp. Phys.* **227** (15), 7125–7159.

Devenport, W. J. & Simpson, R. L. 1990 Time-dependent and time-averaged turbulence structure near the nose of a wing-body junction. *J. Fluid Mech.* **210**, 23–55.

Devenport, W. J., Zsoldos, J. S. & Vogel, C. M. 1997 The structure and development of a counter-rotating wing-tip vortex pair. *J. Fluid Mech.* **332**, 71–104.

Doligalski, T. 1994 Vortex Interactions with Walls. *Ann Rev. Fluid Mech.* **26** (1), 573–616.

Dongarra, J., Hittinger, J., Bell, J., Chacon, L., Falgout, R., Heroux, M. & others. 2014 Applied mathematics research for exascale computing. *Tech. Rep.*.

Durbin, P. A. & Belcher, S. E. 1992 Scaling of adverse-pressure-gradient turbulent boundary layers. *J. Fluid Mech.* **238**, 699–722.

Edwards, H. C., Trott, C. R. & Sunderland, D. 2014 Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing* **74** (12), 3202 – 3216.

El-Askary, W. A. 2009 Turbulent boundary layer structure of flow over a smooth-curved ramp. *Comput. and Fluids* **38** (9), 1718–1730.

El-Sayed, N. & Schroeder, B. 2013 Reading between the lines of failure logs: Understanding how hpc systems fail. *Proceedings of the International Conference on Dependable Systems and Networks* .

Elbing, B. R., Mäkiharju, S., Wiggins, A., Perlin, M., Dowling, D. R. & Ceccio, S. L. 2013 On the scaling of air layer drag reduction. *J. Fluid Mech.* **717**, 484–513.

Englar, R. J 2001 Advanced aerodynamic devices to improve the performance, economics, handling and safety of heavy vehicles. *SAE Technical Paper* (2001-01-2072).

Escauriaza, C. & Sotiropoulos, F. 2011 Reynolds number effects on the coherent dynamics of the turbulent horseshoe vortex system. *Flow turbulence and combust.* **86** (2), 231–262.

Faghih-Naini, S., Kuckuk, S., Aizinger, V., Zint, D., Grosso, R. & Köstler, H. 2020 Quadrature-free discontinuous galerkin method with code generation features for shallow water equations on automatically generated block-structured meshes. *Advances in Water Resources* p. 103552.

Fernandez, Pablo, Nguyen, Ngoc Cuong & Peraire, Jaime 2017 The hybridized discontinuous galerkin method for implicit large-eddy simulation of transitional turbulent flows. *Journal of Computational Physics* **336**, 308–329.

Feynman, R. P., Leighton, R. B. & Sands, M. 1964 The feynman lectures on physics. , vol. II. Addison-Wesley, Boston, MA.

Fidkowski, K. 2019 Solution-based adaptivity as a paradigm for computational fluid dynamics. *Tech. Rep.*. University of Michigan.

Fidkowski, Krzysztof J 2004 A high-order discontinuous galerkin multigrid solver for aerodynamic applications.

Fisher, S. J., Alexander, A. S. & Elbing, B. R. 2020 Computational model of flow surrounding brazilian free-tailed bat ear tubercles. In *AIAA Scitech 2020 Forum*, p. 2021.

Henry de Frahan, M. T. 2016 Numerical simulations of shock and rarefaction waves interacting with interfaces in compressible multiphase flows. PhD thesis, University of Michigan, Ann Arbor.

Frère, A., Hillewaert, K., Chatelain, P. & Winckelmans, G. 2018 High reynolds number airfoil: From wall-resolved to wall-modeled les. *Flow Turb. Comb.* **101** (2), 457–476.

Frere, Ariane, Hillewaert, Koen, Chivaee, Hamid S, Mikkelsen, Robert F & Chatelain, Philippe 2015 Cross-validation of numerical and experimental studies of transitional airfoil performance. In *33rd Wind Energy Symposium*, p. 0499.

Friedrich, R. & Arnal, M. 1990 Analysing turbulent backward-facing step flow with the lowpass-filtered navier-stokes equations. *J. Wind Engng. and Ind. Aero.* **35**, 101–128.

Gainaru, A., Cappello, F., Snir, M. & Kramer, W. 2013 Failure prediction for hpc systems and applications: Current situation and open issues. *International Journal of High Performance Computing Applications* **27** (3), 273–282.

Gautschi, M., Schiavone, P. D., Traber, A., Loi, I., Pullini, A., Rossi, D. & others 2017 Near-threshold risc-v core with dsp extensions for scalable iot endpoint devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **25** (10), 2700–2713.

Geurts, B. J. & Fröhlich, J. 2002 A framework for predicting accuracy limitations in large-eddy simulation. *Phys. Fluids* **14** (6), L41–L44.

Geuzaine, C. & Remacle, J.-F. 2009 Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering* **79** (11), 1309–1331.

Godenschwager, C., Schornbaum, F., Bauer, M., Köstler, H. & Rüde, U. 2013 A framework for hybrid parallel flow simulations with a trillion cells in complex geometries. In *SC'13: Proceedings of Int. Conf. High Perf. Comp., Networking, Storage and Analysis*, pp. 1–12.

Golub, G. H. & Van Loan, C. F. 2012 *Matrix computations*, , vol. 3. John Hopkins University Press., Baltimore, MD.

Gottlieb, S. & Shu, C. 1998 Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society* **67** (221), 73–85.

GUSTAFSON, J. L. & YONEMOTO, I. T. 2017 Beating floating point at its own game: Posit arithmetic. *Supercomputing Frontiers and Innovations* **4** (2), 71–86.

HAIDAR, AZZAM, WU, PANRUO, TOMOV, STANIMIRE & DONGARRA, JACK 2017 Investigating half precision arithmetic to accelerate dense linear system solvers pp. 1–8.

HALILA, G. L.O., CHEN, G., SHI, Y., FIDKOWSKI, K. J., MARTINS, J. R.R.A. & DE MENDONÇA, M. T. 2019 High-reynolds number transitional flow simulation via parabolized stability equations with an adaptive rans solver. *Aerospace Science and Technology* **91**, 321–336.

HARVEY, J. K. & PERRY, F. J. 1971 Flowfield produced by trailing vortices in the vicinity of the ground. *AIAA J.* **9** (8), 1659–1660.

HEINECKE, A., BREUER, A., RETTENBERGER, S., BADER, M., GABRIEL, A.-A., PELTIES, C. & OTHERS 2014 Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers. In *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 3–14. IEEE.

HERBST, A. H., SCHLATTER, P. & HENNINGSON, D. S. 2007 Simulations of turbulent flow in a plane asymmetric diffuser. *Flow, Turbulence and Combust.* **79** (3), 275–306.

HEROUX, M. A., MCINNES, L. C., BERNHOLDT, D. E., DUBEY, A., GONSIOROWSKI, E., MARQUES, O. & OTHERS 2020 Advancing scientific productivity through better scientific software : developer productivity and software sustainability report. *Tech. Rep.*. U.S. Department of Energy Advanced Scientific Computing Research, Oak Ridge National Laboratory.

HOUBA, T., DASGUPTA, A., GOPALAKRISHNAN, S., GOSSE, R. & ROY, S. 2019 Supersonic turbulent flow simulation using a scalable parallel modal discontinuous galerkin numerical method. *Scientific reports* **9** (1), 1–19.

HUNT, J. C. R., WRAY, A. A. & MOIN, P. 1988 Eddies, streams, and convergence zones in turbulent flows. In *Studying Turbulence Using Numerical Simulation Databases, 2. Proceedings of the 1988 Summer Program*.

HWANG, J. & YANG, K. 2004 Numerical study of vortical structures around a wall-mounted cubic obstacle in channel flow. *Phys. Fluids (1994-present)* **16** (7), 2382–2394.

ISSA, R. I. 1986 Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.* **62** (1), 40–65.

IWAMOTO, K., FUKAGATA, K., KASAGI, N. & SUZUKI, Y. 2005 Friction drag reduction achievable by near-wall turbulence manipulation at high reynolds numbers. *Phys. Fluids* **17** (1), 011702–011702.

IYER, P.S. & MAHESH, K. 2013 High-speed boundary-layer transition induced by a discrete roughness element. *J. Fluid Mech.* **729**, 524–562.

JENKINS, L., GORTON, S. A. & ANDERS, S. 2002 Flow control device evaluation for an internal flow with an adverse pressure gradient. In *40th AIAA Aerospace Sciences Meeting*.

Johnson, P. 2019 A recovery-assisted discontinuous galerkin method for direct numerical simulation of compressible turbulence. PhD thesis, University of Michigan, Ann Arbor.

Johnson, P. E. & Johnsen, E. 2019 The compact gradient recovery discontinuous galerkin method for diffusion problems. *J. Comp. Phys.* **398**, 108872.

Kaiktsis, L., Karniadakis, G. E. & Orszag, S. A. 1991 Onset of three-dimensionality, equilibria, and early transition in flow over a backward-facing step. *J. Fluid Mech.* **231**, 501–528.

Kaneko, T. & Liu, B. 1973 On local roundoff errors in floating-point arithmetic. *Journal of the ACM (JACM)* **20** (3), 391–398.

Karakus, A., Chalmers, N., Świrydowicz, K. & Warburton, T. 2019 A gpu accelerated discontinuous galerkin incompressible flow solver. *J. Comp. Phys.* **390**, 380–404.

Khieu, L. H. & Johnsen, E. 2014 Analysis of improved advection schemes for discontinuous galerkin methods. In *7th AIAA Theoretical Fluid Mechanics Conference*, p. 3221.

Kim, W. & Menon, S. 1995 A new dynamic one-equation subgrid-scale model for large eddy simulations. In *33rd AIAA Aerospace Sciences Meeting*, p. 356.

King, J. & Kirby, R. M. 2013 A scalable, efficient scheme for evaluation of stencil computations over unstructured meshes. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–12.

Kitamura, K. & Shima, E. 2013 Towards shock-stable and accurate hypersonic heating computations: A new pressure flux for ausm-family schemes. *J. Comp. Phys.* **245**, 62–83.

Kline, S. J., Reynolds, W. C., Schraub, F. A. & Runstadler, P. W. 1967 The structure of turbulent boundary layers. *J. Fluid Mech.* **30** (4), 741–773.

Kolmogorov, A. N. 1941 The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Cr Acad. Sci. URSS* **30**, 301–305.

Koskela, T., Matveev, Z., Yang, C., Adedoyin, A., Belenov, R., Thierry, P. & others 2018 A novel multi-level integrated roofline model approach for performance characterization. In *International Conference on High Performance Computing*, pp. 226–245. Springer.

Kourta, A., Thacker, A. & Joussot, R. 2015 Analysis and characterization of ramp flow separation. *Exp. Fluids* **56**, 104: 1–14.

Krajnovic, S. & Davidson, L. 1999 Large-eddy simulation of the flow around a surface-mounted cube using a dynamic one-equation subgrid model. In *TSFP Digitial Library Online*. Begel House Inc.

Krajnovic, S. & Davidson, L. 2002 Large-eddy simulation of the flow around a bluff body. *AIAA J.* **40** (5), 927–936.

Lam, M. O., Hollingsworth, J. K., de Supinski, B. R. & LeGendre, M. P. 2013 Automatically adapting programs for mixed-precision floating-point computation. In *Proceedings of the 27th international ACM conference on International conference on supercomputing*, pp. 369–378.

Langer, A., Dokania, H., Kalé, L. V. & Palekar, U. S. 2015 Analyzing energy-time tradeoff in power overprovisioned hpc data centers. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*, pp. 849–854. IEEE.

Le, H., Moin, P. & Kim, J. 1997 A direct numerical study of turbulent flow over a backward-facing step. *J. Fluid Mech.* **330**, 349–374.

Lee, J. H., Sung, H. J. & Krogstad, P. 2011 Direct numerical simulation of the turbulent boundary layer over a cube-roughened wall. *J.Fluid Mech.* **669**, 397–431.

Lee, S., Kevrekidis, I. G. & Karniadakis, G. E. 2017 A general cfd framework for fault-resilient simulations based on multi-resolution information fusion. *Journal of Computational Physics* **347**, 290–304.

van Leer, B. & Nomura, S. 2005 Discontinuous Galerkin for Diffusion. *Proceeding of 17th AIAA Computational Fluid Dynamics Conference* (AIAA 2005-5109).

Leweke, T., Le Dizes, S. & Williamson, C. H.K 2016 Dynamics and instabilities of vortex pairs. *Ann. Rev. Fluid Mech.* **48**, 507–541.

Lin, J., Howard, F. & Selby, G. 1991 Exploratory study of vortex-generating devices for turbulent flow separation control. In *29th AIAA Aerospace Sciences Meeting*, p. 42.

Lin, J. C. 2002 Review of research on low-profile vortex generators to control boundary-layer separation. *Progress in Aero. Sciences* **38** (4), 389–420.

Lindstrom, Peter 2014 Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics* **20** (12), 2674–2683.

Loffeld, J. & Hittinger, J. A. F. 2019 On the arithmetic intensity of high-order finite-volume discretizations for hyperbolic systems of conservation laws. *International Journal of High Performance Computing Applications* **33** (1), 25–52.

Logdberg, O. 2006 Vortex generators and turbulent boundary layer separation control (October).

Lowery, P. S. & Reynolds, W. C. 1986 Numerical simulation of a spatially-developing, forced, plane mixing layer. *Thermosciences Division, Dept. of Mechanical Engineering, Stanford University.* .

Lucas, R., Ang, J., Bergman, K., Borkar, S., Carlson, W., Carrington, L. & others 2014 Top ten exascale research challenges. *Tech. Rep.*. U.S. Department Of Energy, Office of Science, Advanced Scientific Computing Advisory Committee.

Luton, J. A. & Ragab, S. A. 1997 The three-dimensional interaction of a vortex pair with a wall. *Phys. Fluids* **9** (10), 2967–2980.

Marincic, I., Vishwanath, V. & Hoffmann, H. 2017 Polimer: An energy monitoring and power limiting interface for hpc applications. In *Proceedings of the 5th International Workshop on Energy Efficient Supercomputing*, p. 7. ACM.

Martinuzzi, R. & Tropea, C. 1993 The Flow Around Surface- Mounted, Prismatic Obstacles Placed in a Fully Developed Channel Flow. *J. Fluids Engng.* **115** (March), 85–92.

Mavriplis, D. J. 2002 An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *J. Comp. Phys.* **175** (1), 302–325.

Menon, H., Lam, M. O., Osei-kuffuor, D., Schordan, M., Lloyd, S. & others 2018 ADAPT : Algorithmic Differentiation Applied to Floating-Point Precision Tuning. *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'18)* pp. 48:1—-48:13.

Modave, A., St-Cyr, A. & Warburton, T. 2016 Gpu performance analysis of a nodal discontinuous galerkin method for acoustic and elastic models. *Computers & Geosciences* **91**, 64–76.

Mohan, J. K. M., Anoop, D., Shashank, C. & Amar, J. 2013 Effect of Vortex generators on Aerodynamics of a Car: CFD Analysis. *International J. Innovation Engng. and Tech.* **2**, 137–144.

Müller, A., Kopera, M. A., Marras, S., Wilcox, L. C., Isaac, T. & Giraldo, F. X. 2019 Strong scaling for numerical weather prediction at petascale with the atmospheric model numa. *The International Journal of High Performance Computing Applications* **33** (2), 411–426.

NASA 2010 Learning can be a drag. accessed: 2020-04-07.

Naumann, Uwe 2012 *The art of differentiating computer programs: an introduction to algorithmic differentiation*, , vol. 24. Siam.

Ohlsson, J., Schlatter, P., Fischer, P. F. & Henningson, D. S. 2010 Direct numerical simulation of separated flow in a three-dimensional diffuser. *J. Fluid Mech.* **650**, 307–318.

O'Leary, K., Gazizov, I., Shinsel, A., Belenov, R., Matveev, Z. & Petunin, D. 2017 Intel advisor roofline analysis: A new way to visualize performance optimization trade-offs. *Intel Software: The Parallel Universe* **27**, 58–73.

Olschanowsky, C., Strout, M. M., Guzik, S., Loffeld, J. & Hittinger, J. 2014 A study on balancing parallelism, data locality, and recomputation in existing pde solvers. In *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 793–804. IEEE.

Ötügen, M. V. 1991 Expansion ratio effects on the separated shear layer and reattachment downstream of a backward-facing step. *Exp. Fluids* **10** (5), 273–280.

Panchekha, P., Sanchez-Stern, A., Wilcox, J. R. & Tatlock, Z. 2015 Automatically improving accuracy for floating point expressions. *ACM SIGPLAN Notices* **50** (6), 1–11.

Parish, Eric 2018 Variational multiscale modeling and memory effects in turbulent flow simulations. PhD thesis, University of Michigan, Ann Arbor.

Pirozzoli, S. 2011 Numerical methods for high-speed flows. *Ann. Rev. Fluid Mech.* **43**, 163–194.

Pope, S. B. 2000 *Turbulent flows*. Cambridge, UK: Cambridge University Press.

Prandtl, L. 1904 Über flussigkeitsbewegung bei sehr kleiner reibung. *Verhandl. III, Internat. Math.-Kong., Heidelberg, Teubner, Leipzig, 1904* pp. 484–491.

Prince, P. J. & Dormand, J. R. 1981 High order embedded runge-kutta formulae. *J. Comp. App. Math.* **7** (1), 67–75.

Pujals, G., Depardon, S. & Cossu, C. 2010 Drag reduction of a 3d bluff body using coherent streamwise streaks. *Exp. Fluids* **49** (5), 1085–1094.

Purkayastha, A., Guiang, C. S., Schulz, K., Minyard, T., Milfeld, K., Barth, W. & others 2004 Performance characteristics of dual-processor hpc cluster nodes based on 64-bit commodity processors. In *Proceedings of the Linux Clusters Institute (LCI) International Conference: the HPC Revolution*.

Ra, S. H. & Chang, P. K. 1990 Effects of pressure gradient on reattaching flow downstream of a rearward-facing step. *J. Aircraft* **27** (1), 93–95.

Rao, D. M. & Kariya, T. T. 1988 Boundary-layer submerged vortex generators for separation control-an exploratory study. *AIAA J.* **3546**, 1988.

Renac, F, de la Llave Plata, M, Martin, E, Chapelier, J-B & Couaillier, V 2015 Aghora: a high-order dg solver for turbulent flow simulations. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pp. 315–335. Springer.

Rettenberger, S. & Bader, M. 2015 Optimizing i/o for petascale seismic simulations on unstructured meshes. In *2015 IEEE International Conference on Cluster Computing*, pp. 314–317. IEEE.

Reynolds, O. 1883 Philos. trans. r. soc. london .

Reynolds, O. 1894 Study of fluid motion by means of coloured bands. *Nature* **50** (1285), 161–164.

Rodi, W. 1997 Comparison of les and rans calculations of the flow around bluff bodies. *J. Wind Engg. Ind. Aero.* **69**, 55–75.

Rogallo, R. S. & Moin, P. 1984 Numerical simulation of turbulent flows. *Ann. Rev. Fluid Mech.* **16** (1), 99–137.

Rossinelli, D., Hejazialhosseini, B., Hadjidoukas, P., Bekas, C., Curioni, A., Bertsch, A. & others 2013 11 pflop/s simulations of cloud cavitation collapse. In *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–13. IEEE.

Rossinelli, D., Hejazialhosseini, B., Spampinato, D. G. & Koumoutsakos, P. 2011 Multicore/multi-gpu accelerated simulations of multiphase compressible flows using wavelet adapted grids. *SIAM Journal on Scientific Computing* **33** (2), 512–540.

Rubio-González, C., Nguyen, C., Nguyen, H. D., Demmel, J., Kahan, W. & others 2013 Precimonious: Tuning assistant for floating-point precision. In *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–12. IEEE.

Sagaut, P. 2001 *Large eddy simulation for incompressible flows*. Springer-Verlag, Berlin: Scientific Computation.

Schroeder, B. & Gibson, G. A. 2007 Understanding failures in petascale computers. *Journal of Physics: Conference Series* **78** (1).

Schwartzkopff, T., Dumbser, M. & Munz, C. D. 2004 Fast high order ADER schemes for linear hyperbolic equations. *J. Comp. Phys.* **197** (2), 532–539.

Selby, G. V., Lin, J. C. & Howard, F. G. 1990 Turbulent Flow Separation Control Over a Backward- Facing Ramp Via Transverse and Swept Grooves. *J. Fluids Engng.* **112**.

Shah, K. B. & Ferziger, J. H. 1997 A fluid mechanicians view of wind engineering: Large eddy simulation of flow past a cubic obstacle. *J. Wind Engg. Ind. Aero.* **67**, 211–224.

Shinde, S., Johnsen, E. & Maki, K. 2017 Understanding the effect of cube size on the near wake characteristics in a turbulent boundary layer. In *AIAA, 47th Fluid Dynamics Conference*.

Shinde, S., Tandon, S., Johnsen, E. & Maki, K. 2016 Flow separation over a backward-facing ramp with and without a vortex generator. In *AIAA, 46th Fluid Dynamics Conference*.

Shinde, S. D. 2018 A Computational Study of Flow Over a Wall-Mounted Cube in a Turbulent Boundary Layer Using Large Eddy Simulations. PhD thesis, University of Michigan, Department of Mechanical Engineering.

Shoga, K., Rountree, B., Schulz, M. & Shafer, J. 2014 Whitelisting msrs with msr-safe. In *3rd Workshop on Exascale Systems Programming Tools, in conjunction with SC14*.

Sim, J., Dasgupta, A., Kim, H. & Vuduc, R. 2012 A performance analysis framework for identifying potential benefits in gpgpu applications. In *Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*, pp. 11–22.

Simpson, R. L. 2001 Junction flows. *Ann. Rev. Fluid Mech.* **33** (1), 415–443.

Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E. & Mavriplis, D. 2014 Cfd vision 2030 study: a path to revolutionary computational aerosciences. *Technical Report CR-2014-218178, NASA* .

Smits, A. J., McKeon, B. J. & Marusic, I. 2011 High–reynolds number wall turbulence. *Ann. Rev. Fluid Mech.* **43**.

Song, S., DeGraaff, D. B. & Eaton, J. K. 2000 Experimental study of a separating, reattaching, and redeveloping flow over a smoothly contoured ramp. In *International J. Heat and Fluid Flow*, , vol. 21, pp. 512–519.

Stengel, H., Treibig, J., Hager, G. & Wellein, G. 2015 Quantifying performance bottlenecks of stencil computations using the execution-cache-memory model. In *Proceedings of the 29th ACM on International Conference on Supercomputing*, pp. 207–216.

Tan, K. A., Morison, R. P. & Leslie, L. M. 2005 A comparison of high-order explicit and non-oscillatory finite difference advection schemes for climate and weather models. *Meteorol. Atmos. Phys.* **89** (1-4), 251–267.

Tandon, S. & Johnsen, E. 2020 Improving the parallel efficiency of recovery-assisted discontinuous galerkin methods on modern hpc systems In preparation.

Tandon, S., Maki, K. J. & Johnsen, E. 2019 Understanding the dependence of turbulent flow modulation on the spacing between adjacent cubes on a backward-facing ramp. In *AIAA Aviation 2019 Forum*, p. 3633.

Tandon, S., Maki, K. J. & Johnsen, E. 2020*a* Modulation of flow over a backward-facing ramp by a wall-mounted cube. *J. Fluid Mech.* Under Review.

Tandon, S., Maki, K. J. & Johnsen, E. 2020*b* Modulation of flow over a backward-facing ramp by an array of wall-mounted cubes In preparation.

Tandon, S., Marincic, I., Hoffmann, H. & Johnsen, E. 2020*c* Enabling power-performance balance with transprecision calculations for extreme-scale computations of turbulent flows. In *AIAA Aviation 2020 Forum*, p. 3633.

Tandon, S., Shinde, S., Johnsen, E. & Maki, K. 2017 Flow control using passive vortex generators. In *AIAA, 47th Fluid Dynamics Conference*.

Tandon, S., Shinde, S., Maki, K. & Johnsen, E. 2018 Near-wake flow modulation by a cube on a backward-facing ramp. In *2018 AIAA Flow Control Conference*, p. 3526.

Taylor, G. I. & Green, A. E. 1937 Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* **158** (895), 499–521.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Dave, L., Peterson, G. D., Roskies, R., Scott, J. R. & Wilkins-Diehr, N. 2014 Xsede: Accelerating scientific discovery. *Comput. Science & Engng.* **16** (5), 62–74.

Uranga, A, Persson, P-O, Drela, M & Peraire, J 2011 Implicit large eddy simulation of transition to turbulence at low reynolds numbers using a discontinuous galerkin method. *International Journal for Numerical Methods in Engineering* **87** (1-5), 232–261.

Vanderwel, C. & Ganapathisubramani, B. 2015 Effects of spanwise spacing on large-scale secondary flows in rough-wall turbulent boundary layers. *J. Fluid Mech.* **774**.

Weller, H. G., Tabor, G., Jasak, H. & Fureby, C. 1998 A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **12** (6), 620.

WESTPHAL, R. V., JOHNSTON, J. P. & EATON, J. K. 1984 Experimental study of flow reattachment in a single-sided sudden expansion .

DE WIART, CC & HILLEWAERT, K 2015 Development and validation of a massively parallel high-order solver for dns and les of industrial flows. In *IDIHOM: industrialization of high-order methods-a top-down approach*, pp. 251–292. Springer.

CARTON DE WIART, C., HILLEWAERT, K., DUPONCHEEL, M. & WINCKELMANS, G. 2014 Assessment of a discontinuous galerkin method for the simulation of vortical flows at high reynolds number. *International Journal for Numerical Methods in Fluids* **74** (7), 469–493.

WILCOX, D. C. 1998 *Turbulence modeling for CFD*, , vol. 2. DCW industries La Canada, CA.

WILLIAMS, S., WATERMAN, A. & PATTERSON, D. 2009 Roofline: an insightful visual performance model for multicore architectures. *Communications of the ACM* **52** (4), 65–76.

WILSON, T. C., KC, R., LUCIDO, N. A., ELBING, B. R., ALEXANDER, A. S., JACOB, J. D., IRELAND, P. & BLACK, J. A. 2019 Computational investigation of the conformal vortex generator. In *AIAA Scitech 2019 Forum*, p. 2138.

WU, X. & MOIN, P. 2009 Direct numerical simulation of turbulence in a nominally zero-pressure-gradient flat-plate boundary layer. *J. Fluid Mech.* **630**, 5–41.

YANG, J. & ANDERSON, W. 2018 Numerical study of turbulent channel flow over surfaces with variable spanwise heterogeneities: topographically-driven secondary flows affect outer-layer similarity of turbulent length scales. *Flow Turb. Comb.* **100** (1), 1–17.

YANG, X. I. A., XU, H. H. A., HUANG, X. L. D. & GE, M.-W. 2019 Drag forces on sparsely packed cube arrays. *J. Fluid Mech.* **880**, 992–1019.

YAO, C., LIN, J. & ALLEN, B. 2002 Flowfield measurement of device-induced embedded streamwise vortex on a flat plate. In *1st Flow Control Conference*, p. 3162.

YARKHAN, A., KURZAK, J., ABDELFATTAH, A. & DONGARRA, J. 2019 An empirical view of {SLATE} algorithms on scalable hybrid systems .

ZHANG, W., WEI, W. & CAI, X. 2014 Performance modeling of serial and parallel implementations of the fractional adams-bashforth-moulton method. *Fractional Calculus and Applied Analysis* **17** (3), 617–637.

ZURAS, D., COWLISHAW, M., AIKEN, A., APPLEGATE, M., BAILEY, D., BASS, S. & OTHERS 2008 Ieee standard for floating-point arithmetic. *IEEE Std* **754** (2008), 1–70.