Application of GPU Accelerated Paired Explicit Runge-Kutta Methods to Turbulent Flow Over a Sphere

Alex Lalaian

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

October 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Alex Lalaian**

Entitled: **Application of GPU Accelerated Paired Explicit Runge-Kutta Methods to Turbulent Flow Over a Sphere**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Marius Paraschivoiu

_____ Examiner
Dr. Marius Paraschivoiu

_____ Examiner
Dr. Ebenezer Ekow Essel

_____ Supervisor
Dr. Brian Vermeire

Approved by _____
Martin D. Pugh, Chair
Department of Mechanical and Industrial Engineering

_____ 2023

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Application of GPU Accelerated Paired Explicit Runge-Kutta Methods to Turbulent Flow Over a Sphere

Alex Lalaian

The design of next generation aircraft will rely on the use of high-fidelity computational fluid dynamics (CFD). For this purpose, the Paired Explicit Runge-Kutta (P-ERK) time stepping method was developed. It is a variation of explicit Runge-Kutta methods that allows the pairing of multiple methods within a given simulation. This results in higher stability methods being used in stiff regions, and lower cost methods being used in non-stiff regions, resulting in significant performance improvements. This work explores the utility of Graphical Processing Unit (GPU) acceleration combined with P-ERK schemes Subcritical flow over a smooth sphere at a Reynolds number (Re) of 3700 was used as the validation case. The flux reconstruction (FR) spatial discretization scheme was used with the implicit Large Eddy Simulation (ILES) turbulence modelling approach. Instantaneous quantities such as velocity fluctuations, Strouhal numbers, and time-averaged quantities such as drag coefficient, pressure coefficient, back pressure ratio, and Reynolds stresses were obtained. In addition, Q-criterion contours were generated and used to obtain separation angle and recirculation bubble length. This study shows that the P-ERK method can achieve good agreement with both reference simulation, and experimental data. Moreover, when compared to the traditional fourth order Runge-Kutta (RK) method the P-ERK scheme shows an average speedup factor of 4.73 using GPUs and of 5.82 using CPUs with regards to solution polynomial scaling, and with regards to resource scaling it requires approximately 8 times more resources using CPUs, or each CPU has an approximately 45 times greater runtime compared to one GPU . This is a significant reduction in computational times, while maintaining accuracy.

Acknowledgments

Firstly, I would like to express my deepest gratitude to my thesis supervisor, professor Brian C. Vermeire, for their invaluable guidance, patience, and support throughout my research. Their expertise and insightful feedback have been instrumental in the completion of this work.

I would also like to thank my colleagues for their assistance over the duration of my degree, especially Carlos for his insight and assistance, and Christina for the many academic exchanges over the years.

Finally, I would like to thank my family and friends for their love, unwavering support, and encouragement over the years without which, I would not have been able to achieve any of my objectives.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, climate change and its impact on both the environment and human health has come to the forefront of political discourse. Already, the changing climate has caused worsening natural disasters and other major ecological changes [6]. This has resulted in various climate action plans being adopted over the years, such as the Kyoto Protocol [7] with moderate levels of success. The most recent climate discussions yielded the Paris Agreement [8]. While Canada had withdrawn from the Kyoto Protocol in 2012 [9], it has committed itself to meeting the Paris Agreement's ambitious objectives. This agreement requires reducing greenhouse gas emissions to below 2005 levels by 2030. To achieve theses objectives, industries will have to develop many new, clean, and efficient technologies, including the aviation industry.

With the adoption of the Waypoint 2050 project by the Air Transport Action Group (ATAG), the aviation industry has committed itself to becoming net-zero for carbon emissions by 2050 among a slew of other objectives aimed at tackling climate change [1]. These objectives are in line with the goals of the Paris Agreement and go beyond them by setting a variety of other climate-related objectives and by extending even further in time. Many different avenues have been proposed to achieve these ambitious goals, such as improvements to infrastructure, the use of sustainable fuels and investments in new technologies, to name a few. The ATAG estimates that, of the methods proposed, new technologies such as innovations to airframes and propulsion systems will represent

22% of the goal assuming that the industry does not simply remain on the current course [1]. Figure 1.1 shows how the ATAG breaks down contributions by method, with new technologies accounting for 22% and the transition to sustainable fuels accounting for 61%.

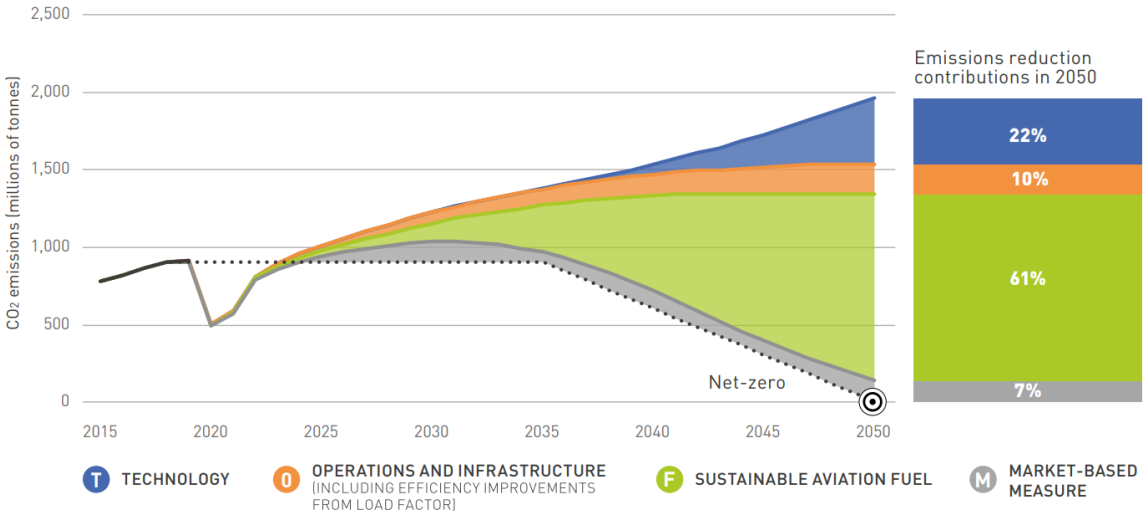


Figure 1.1: Emission reduction contributions ATAG Scenario 1 [1]

This will require redesigns of airframes and propulsion systems. Innovative airframe designs such as blended wing-body configurations like that explored by Airbus [10] and various high-lift configurations show promise, but also pose significant design challenges as many of the tools developed to accelerate aircraft design no longer apply. When looking at improvements to propulsion systems, one of the biggest roadblocks to innovation is the complexity of designing multi-stage axial turbines and compressors. In fact, obtaining type certificates for new aircraft engines can take many years and requires many flight tests, which results in high costs for new development programs. Therefore, the increasing use of Computational Fluid Dynamics (CFD) to compute the solutions of the flow in these difficult-to-analyse sections has enabled more cost-effective and rapid development of innovative gas turbine engine designs. As the Navier-Stokes (NS) equations which govern fluid behaviour cannot be solved exactly, CFD requires the use of numerical methods and various turbulence modelling approaches to obtain solutions with engineering levels of accuracy as well as feasible and affordable computational costs.

1.2 Turbulence Modelling

There are three primary turbulence modelling approaches being used in modern CFD applications. These are the Direct Numerical Simulation (DNS), Reynolds-Averaged Navier-Stokes (RANS) and, Large Eddy Simulation (LES). These will be explored in greater depth in Chapter 3 of this study.

1.2.1 Direct Numerical Simulation

While Direct Numerical Simulation is often grouped with the various turbulence modelling methods, strictly speaking it is not a model at all. In fact, it simply refers to an approach where all of the temporal and spatial scales found in the flow are numerically solved directly. The term DNS was first used in 1970 by Orszag [11] in his study of small-scale isotropic incompressible turbulent flow at a small Reynolds number of 35 to differentiate it from the various models being developed at that time. It was further developed by Rogallo in 1981 [12] in their study of the effects of mean shear, irrotational strain, and rotation on homogeneous turbulence.

While this method is relatively straightforward, its shortcomings led to the development of more complex turbulence models. The primary issue with DNS stems from the fact that it directly solves for all the spatial scales from the smallest to the largest. This imposes a requirement on the mesh at high Reynolds numbers to be fine enough to capture the smallest scales such as at a wall boundary. This results in high memory usage and means that for any given iteration a relatively large amount of floating point operations are required. For most CFD applications this is simply not an effective method due to the wide range of turbulent scales present and the resulting computational cost.

Most flow regimes that are of engineering interest are at high Reynolds numbers and include wall boundaries within the computational domain, such as, flow over a plate, channel flow, flow in a pipe, and flow over a sphere to name a few. Due to the cost associated with DNS, it was historically not feasible to perform these. It was only in the late 1980s with the studies by Moser and Moin [13] of flow in a curved channel, and by Kim et al [14] of plane channel flow that DNS became feasible for wall-bounded turbulent flows. Since then, more studies have been conducted using DNS, including studies of compressible flow [15, 16, 17, 18], turbulent mixing [19], shock-waves

[20], and others. For modern applications, DNS is primarily used in conjunction with physical experiments as a means of providing reliable validation data for the development of novel and more efficient methods [21]. However, it remains prohibitively expensive for practical aerospace applications.

1.2.2 Reynolds-Averaged Navier-Stokes

The Reynolds-Averaged Navier-Stokes (RANS) approach addresses some of the challenges present with DNS. The primary principle of this method involves separating instantaneous quantities, such as velocity, into a time-averaged component and its instantaneous fluctuation. It was first proposed by Oswald Reynolds in 1894 [22]. However, it could not be practically applied at that time, as there was no way to close the resulting system of equations for the mean flow. It was only with the supporting work of Boussinesq in 1877 [23] that introduced the use of an eddy-viscosity based model to close the system of equation, and the work of Prandtl in 1925 [24] who introduced the concept of a mixing length as a method of determining the eddy-viscosity, that it became possible to solve an approximate form of these equations.

RANS has the advantage of significantly reduced computational costs compared to DNS. This is due to only solving a single steady field instead of the complex turbulent fields inherent to the NS equations that must be solved directly with DNS. This also means that more generous grid spacing can be used, further reducing the amount of computation required. This method however suffers from two major drawbacks: it requires the use of turbulence models to close the system of equations and by its time-averaged nature can only provide information about steady-state flow. For many applications, computing time-averaged statistics is sufficient as these quantities will include terms such as: lift, drag, pressure coefficient, and others. However, it is also often interesting to obtain information about instantaneous behaviour in the flow which this method cannot provide.

Further work on RANS modelling has come in the form of a series of turbulence models to more accurately close the system of equations. This includes the mixing-length approach proposed by Prandlt [24]. This was however, found to be deficient as it tried to use clearly separated scales to resolve the flow, which is unrealistic at high Reynolds. This made way for the method proposed by Pandtl in 1945 [25] that used a model to obtain the turbulent kinetic energy and used that as a

basis for the eddy-viscosity instead. While methods based on this approach were considered more accurate, they still had many of the drawbacks inherent to eddy-viscosity based methods. The work of Rotta in 1951 [26] saw the emergence of Reynods-stress based turbulence closure models. This approach remains the basis for most modern RANS methods used today. Significant contributions in this field were made by Naot et al [27] (who first proposed a tensor-based approach), Jones and Launder [28], Hanjalic and Launder [29], Launder, Reece and Rodi [30], Schumannn [31], and Lumley [32]. Many different models are used for modern applications, each having various advantages and disadvantages. Some of the more common models are the $k-\epsilon$ model developed by Hanjalic and Launder[29], Menter's shear stress transport turbulence model [33], and the $k-\omega$ model developed by Wilcox [34]. It is important to consider the choice of model carefully when using RANS for CFD as, depending on the particular situation, they can introduce significant error. In particular, RANS models are generally deficient for transitional and separated turbulent flows, which are found in a wide range of aerospace applications.

1.2.3 Large Eddy Simulation

While the DNS method solves NS at all scales of motion from the smallest (Kolmogorov scale) to the largest, and the RANS method instead only solves the time-averaged large scale behaviour and models all of the smaller-scale behaviour, LES serves as a compromise between these two extremes. LES was initially proposed by Smagorinsky in 1963 [35] for meteorological applications and was first successfully used in an engineering context by Deardorff in 1970 [36] and Schumann in 1975 [37].

Instead of time-averaging the NS equation, LES uses filtering to eliminate only the smallest scales which means that it computes all of the instantaneous parameters of the flow for most scales and can therefore capture most of its unsteady behaviour, which RANS cannot including: vortex shedding frequencies, boundary layer behaviour, and more. This also opens up LES method to many interesting applications, such as more accurate and informative combustion modelling [38] where there is significant small-scale transient behaviour that is pertinent to capture. This, however, means that its computational cost is significantly greater than for a similar RANS simulation; the increase being in the order of 10 to 100 times [39] is not negligible and must therefore be taken into

consideration when selecting a turbulence modelling approach.

Primary advancements with LES have to do with the development of novel sub-grid scale (SGS) models to successfully resolve the flow behaviour at the very small scales that are no longer solved directly by the LES method. This includes work by: Leonard in 1974 who proposed the usage of a kernel to eliminate scales under a given length-scale [40], Bardino et al in 1983 [41], Speziale in 1985 [42], Germano et al in 1991 [43], and Germano in 1992 [44] who proposed and tested various sub-grid scale modelling approaches. Additionally, novel LES methods that do not make use of an SGS model have been developed, such as Implicit Large Eddy Simulation (ILES), which uses the numerical truncation error to model the SGS behaviour first proposed by Boris in 1990 [45]. This method is of particular interest for this study and will be explored in greater depth in Chapter 3.

1.3 Discretization

1.3.1 Spatial Discretization Schemes

To be able to apply DNS, RANS or LES to the governing partial differential equations (PDE) they must first be discretized. Many methods for spatial discretization have been proposed over the years such as the: Finite Difference (FD), Finite Volume (FV), Finite Element (FE), Discontinuous Galerkin (DG) [46], Spectral Volume (SV) [47], Spectral Difference (SD) [48], and Flux Reconstruction (FR) [49] methods.

The finite difference method is one of the oldest spatial discretization methods commonly used in CFD [50]. It consists of evaluating a given function at an arbitrary number of points using a Taylor expansion to approximate the local solution. For a small enough grid spacing, the truncation error can be reduced sufficiently so that the derivative for any given point can be approximated explicitly using information from nearby points. This method can obtain high orders of accuracy depending on the order of the derivatives being computed and is relatively simple to understand and implement. It struggles greatly however, when applied to irregular geometries requiring magnitudes more effort in the implementation so as to be applicable to such cases. This drawback is the primary reason why it is not more widely used [51, 52].

The finite volume method traces its origins to the 1970s and work conducted by McDonald [53].

It addresses the structured grid restrictions of the FD method allowing FV to be applied to a wider range of geometries. It treats each cell in the mesh as a control volume with a piecewise constant internal solution and with discontinuous boundaries. In addition, it uses the governing equations in their integral form which, when combined with the previously mentioned characteristics, results in the method being simple to implement, having low memory usage, and relatively quick computation speeds. It's cost increases drastically however as the order of the method increases and extensions to high order are non-trivial on unstructured meshes. This makes the method unsuitable for high-order applications [51, 52].

The finite element method has origins in structural analysis but has been extended to fluid dynamic applications as well. It comprises a family of methods that share a similar structure. With the finite element method, the solution is evaluated at a number of points. Basis functions are used at each point to approximate the solution surrounding it with a continuity constraint for the solution to match at each interface with neighbouring elements. The solution produced by FEM is therefore defined globally and is continuous over the entire domain. This is, however, computationally intensive requiring the inversion of a global mass matrix, and suffers from stability issues when applied to certain type of flow such as compressible flow [51, 52].

The discontinuous Galerkin method is, strictly speaking, a type of FE method [50]. It attempts to resolve the issues inherent to FE methods by implementing features from FV methods in its framework. It evaluates the solution in its integral form at a number of points contained within control volumes. The solution is expressed within each volume by a polynomial passing through a set of solution points. This results in the solution being continuous within each element, and discontinuous at element interfaces. This, therefore, requires integration quadratures across each element boundary to compute common fluxes. Its internal polynomial representation results in the method being able to scale effectively to high orders, unlike the FV method, while having fewer stability issues than the FE method, and reduced computational cost. Additionally, as the solution is evaluated locally instead of globally, this method can be easily parallelized [50, 52].

The spectral volume method is in many ways similar to the DG method [47]. While the DG method makes use of a control volume and evaluates the solution at a series of points within said volume, the SV method instead subdivides the "spectral volume" into a set of "control volumes".

The solution is then approximated using independently-computed averages for each control volume to obtain a solution for the spectral volume. The fluxes at each spectral volume boundary are computed similarly to the DG method. This results in the method maintaining the advantages of the FV method while allowing it to scale to higher orders by modifying the number and shape of the control volumes which is computationally relatively inexpensive [47, 52, 54].

The spectral difference method is very similar to the DG and SV methods [48]. Its difference lies in that it combines elements from the FD and FV methods. In fact, in 1D the SV and SD methods are identical [54]. The solution is evaluated in its differential form at a set of solution points and the flux is evaluated at a set of flux points within a given element. Generally, Gauss quadrature points are used for the solution points and the Gauss-Lobatto quadrature points are used for the flux points [55]. As with the FD method, the solution is evaluated at the solution points exclusively. The flux derivatives are expressed at the flux points and are used to couple the solution in a given cell with its neighbours. As the solution flux is discontinuous, a common flux must be computed at the boundaries of the elements as with the DG method. This method is very promising with good performance at high orders and high levels of stability [48, 55].

The flux reconstruction method serves as a framework by which most other methods can be recovered. This gives it a special advantage over other methods as whichever method is most appropriate for the application can be used in a smart way. It does this by introducing a general correction field for the flux. Depending on the choice of correction field various other methods such as: FV, DG, SD, and SV can be recovered [49]. Table 1.2 is an extension of the table generated by Hesthaven and Warburton [56] illustrating the relative advantages of the various methods. Due to its advantages, the FR method is of particular interest for this study and will be explored in Chapter 4.

1.3.2 Temporal Discretization Schemes

Knowing the initial and boundary conditions of a given governing equation, and having selected an appropriate spatial discretization method, the solution for a single initial instant in time can be evaluated. However, for DNS and LES the objective when performing CFD is to evaluate the solution over a period of time. Whether it is to obtain time-averaged statistics such as the pressure coefficient (Cp) or to obtain instantaneous values such as shedding frequencies, a method must be

Schemes	Unstructured Method	High-Order Accurate	Explicit Form	Quadrature Free	Unifying
Finite Difference Method	✗	✓	✓	✓	✗
Finite Volume Method	✓	✗	✓	✓	✗
Finite Element Method	✓	✓	✗	✓	✗
Discontinuous Galerkin	✓	✓	✗	✗	✗
Flux Reconstruction	✓	✓	✓	✓	✓

Figure 1.2: Comparison of spatial schemes [2]

used to march time forward. These methods are known as temporal discretization methods. There are two primary families of discretization schemes: explicit, and implicit schemes, as well as the family of implicit-explicit schemes (IMEX) that use parts of each.

Explicit methods are defined as time-marching methods that rely solely on the current and past times to determine future times. Each time step can therefore be solved explicitly without requiring iterative methods to arrive at a converged future time. When explicit methods are discussed in the context of CFD, it is usually in reference to the explicit Runge-Kutta (RK) methods [57]. These methods consist of splitting a given time step into an arbitrary number of intermediate stages. The intermediate stage solutions are used to obtain the subsequent stage residuals and are combined with appropriate coefficients to obtain the solution for the next time step. The coefficients used are organized with the help of a Butcher tableau. For an explicit method, the tableau will always be lower triangular. This method has many advantages such as being simple to implement, relatively quick per step and can scale to high-order accuracy in a straightforward manner.

However, it also has many significant disadvantages, such as only being able to achieve conditional stability for the step size. This stability constraint has been studied extensively over the years, such as by Courant, Friedrichs and Lewy in 1928 [58] in their study that established that the ratio of the flow velocity multiplied with the time step size divided by the spatial step size must be maintained below a limit to achieve stability for an explicit method. This means that for explicit methods, the number of computations required to achieve a desired final time while maintaining stability can be high. In addition, as the order of the solution increases, the number of intermediate

stages required also increases. This results in large amounts of memory being needed to store the intermediate stages' residuals to be able to compute the next solution's time step. [57]

To circumvent the limitations of explicit time stepping methods, implicit schemes were developed. Implicit schemes differ from explicit schemes in that they require information of a future time step to resolve the solution. As that information cannot be obtained explicitly without using an explicit time stepping method as above, iterative numerical methods are employed to obtain the information at the future time. There are many possible implicit methods that have been developed over the years, and they generally differ in what current and future values are required to arrive at a solution. [59]

An example of an implicit method would be the implicit Runge-Kutta methods (IRK). These are conceptually similar to the explicit RK methods. However, their Bucher tableaus are not lower triangular which means that their stages are at least partially dependent on future stages for any given stage. Therefore, to compute a given time step, all of the intermediate stages must be treated as a system of equations, and employing iterative methods, achieve a converged solution for each stage. There are many ways to go about performing these iterations: the Beam-Warming algorithm [60], Jacobi [61], Gauss-Seidel [62], or Krylov subspace methods such as conjugate gradient or generalized minimal residual method (GMRES) [63].

The primary advantage of implicit methods has to do with their stability. They can be unconditionally linearly stable, meaning that arbitrarily large steps can be taken. However, to achieve temporal convergence, relatively complex computational iterations must be performed repeatedly until convergence is achieved. In addition, these methods are applied to the entire solution simultaneously. This is time- and resource-intensive. Therefore, the set of problems where implicit methods are preferable over explicit methods is limited to highly stiff systems because the computational cost of the implicit method will be lesser than the cost of the explicit method due to its stability constraints [59].

Implicit-explicit schemes aim to alleviate the limitations of each of the implicit and explicit methods outlined above. They do so by applying an unconditionally stable implicit method for the stiff portions of the solution, and a low-cost explicit method in the non-stiff portion of the solution. They are paired together such that a time stepping method is obtained that maintains the

order of accuracy of the component methods. This means that IMEX methods will have a lower computational cost than fully implicit methods, while circumventing the stability restrictions of explicit methods. IMEX methods have been shown to achieve speed up factors of 91 and a 70% reduction in memory usage compared to fully implicit schemes [64].

However, these methods have issues extending to high order due to the nature of the implicit portion requiring many simultaneous computations, which in turn demands large amounts of memory and computational power. In fact for hexahedral elements, it can be expected that a 5th order scheme will use approximately 250 times the amount of memory as a second order scheme. This scaling similarly applies to the dimensionality of the solution where a three-dimensional solution could be expected to use 6.5 times the memory of a one-dimensional solution [65].

While IMEX schemes are promising as a method to solve locally stiff problems at a reduced cost, their complexity makes implementation and use challenging [59]. This means that it is of interest to find a different method that does not require the simultaneous use of many variables, that is relatively simple to implement, that scales to higher orders, and that can at least partially circumvent issues related to numerical stiffness. The Paired Explicit Runge-Kutta method proposed by Vermeire in 2019 addresses these requirements using families of fully explicit methods [66]. The method consists of developing a special kind of Butcher tableau that allows low-stability, low cost, and high-stability, high cost RK methods to be used together seamlessly for any given problem. High stability methods are used for stiff regions, whereas low-stability methods are used for low stiffness regions. This allows the method to mitigate the effects of numerical stiffness without using implicit methods. It has previously been extended to third-order accuracy by Hedayati Nasab, where speedup factors up to 4 were observed [67]. This method is the focus of the current study and will be explored in greater detail in Chapter 5.

1.4 High-Performance Computing

Moore's law, as seen in Figure 1.3 stipulates that the number of transistors which can be fitted on a microchip will double approximately every two years and will therefore allow for smaller process sizes to be utilized. For the past six decades, this has held true and has allowed process levels to

advance from 20 micrometers introduced by RCA process to a current process of 3 nanometres introduced by TSMC [68]. These leaps have allowed processors and computers to become more affordable and powerful year over year, and has lead to the widespread use of computers both for personal computing and industrial applications. In turn, this enabled progress in fields which had previously been constrained due to the lack of high computational power. Similarly, the clock rate (or frequency) of the processor, which can be defined as the number of operations in can perform per unit of time, has seen an important increase from an initial 1 Hz to more than 5 GHz.

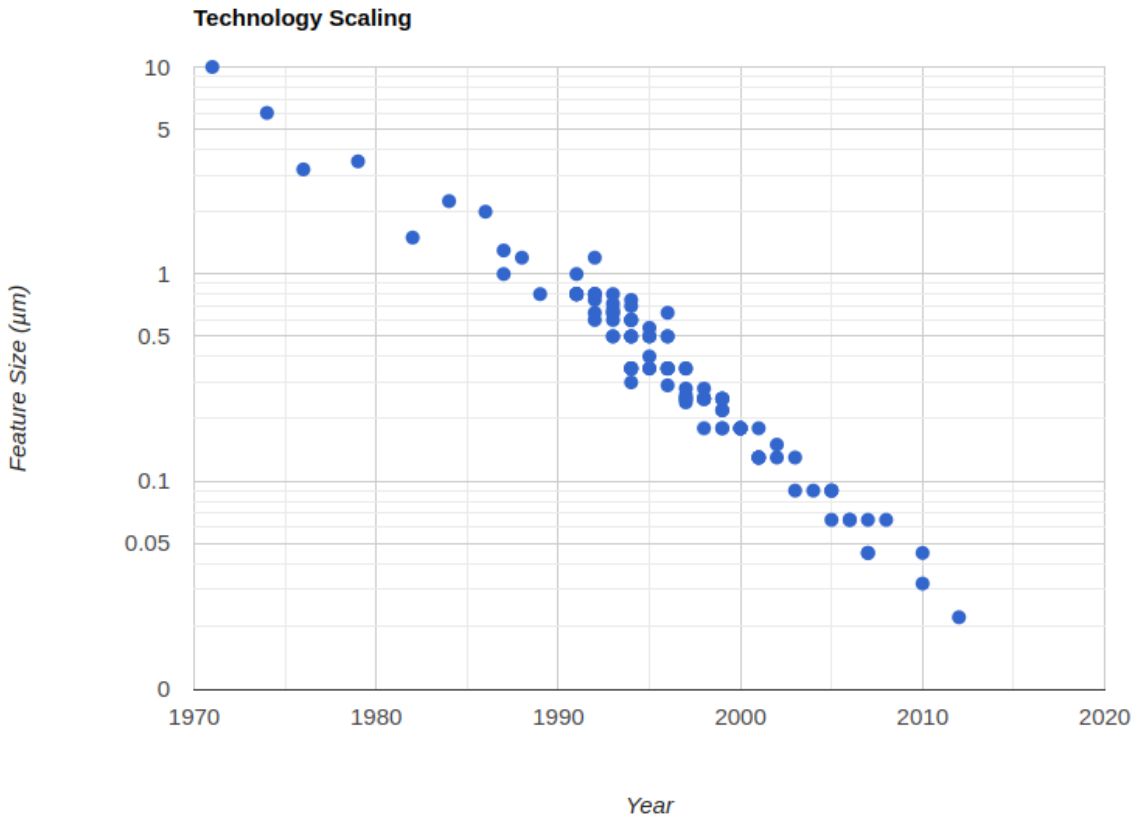


Figure 1.3: Moore's law [3]

These important advances have allowed high-performance computing to accelerate the development of research fields, and also allowed for the development of more advanced types of computation to further increase efficiency and reduce compute time for applications and processes. High-performance computing typically involves the use of computer clusters made up of individual

computers referred to as nodes. One of these methods is called parallel computing, where multiple computers are utilized for a given application to tackle its different aspects simultaneously and therefore increase the time efficiency. As clusters have a number of nodes, this allows for a higher degree of parallelization due to their interconnection. A task can be broken down into sub-tasks, which can then be performed concurrently on separate computing nodes or on a single node instead of sequentially on a single computer. It should be mentioned that as the number of transistors on a process has increased exponentially, this has resulted in an increased number of resources, such as computer cores, being able to be fit in a single chip [69]. Modern computers therefore make use of parallel computing as well to better handle the increasing amount of resources made available to them. The benefits of more efficient parallel processes ultimately extended beyond its usage in high-performance computing to every day applications.

While Moore's law and parallel computing methods have enabled central processing units (CPU) to achieve important performance milestones, the difficulty of further reducing the size of components and issues with effective heat dissipation have started to limit the possible performance gains using this method [70]. This is in part due to the architecture used in CPUs that prioritizes performing a few computations at very high speeds by making use of larger and more powerful cores. This is ideal for performing complex operations sequentially with low latency. Graphics processing units (GPU) in turn rely on a large number of smaller cores to perform many operations in parallel. This is due to their original purpose of rendering images and video in real time requiring many repetitive operations. The key factor in maximizing the performance gain when using GPUs is to ensure that the operation to be performed can be broken up into many smaller tasks because otherwise it is possible that not all of the resources available can be utilized effectively. Figure 1.4 illustrates the advantages of parallelization and GPU acceleration when applied to CFD using Ansys Fluent. It shows the speedup experienced when using Ansys Fluent to perform aerodynamic analysis of a car which is significant. In fact it shows that even a single GPU can outperform an entire cluster of CPUs.

Another key element to consider is memory. Memory dictates the maximum size and amount of variables that can be retained by the program while it is running. The availability of memory has increased over time just as with CPU performance. This has enabled larger and more complex

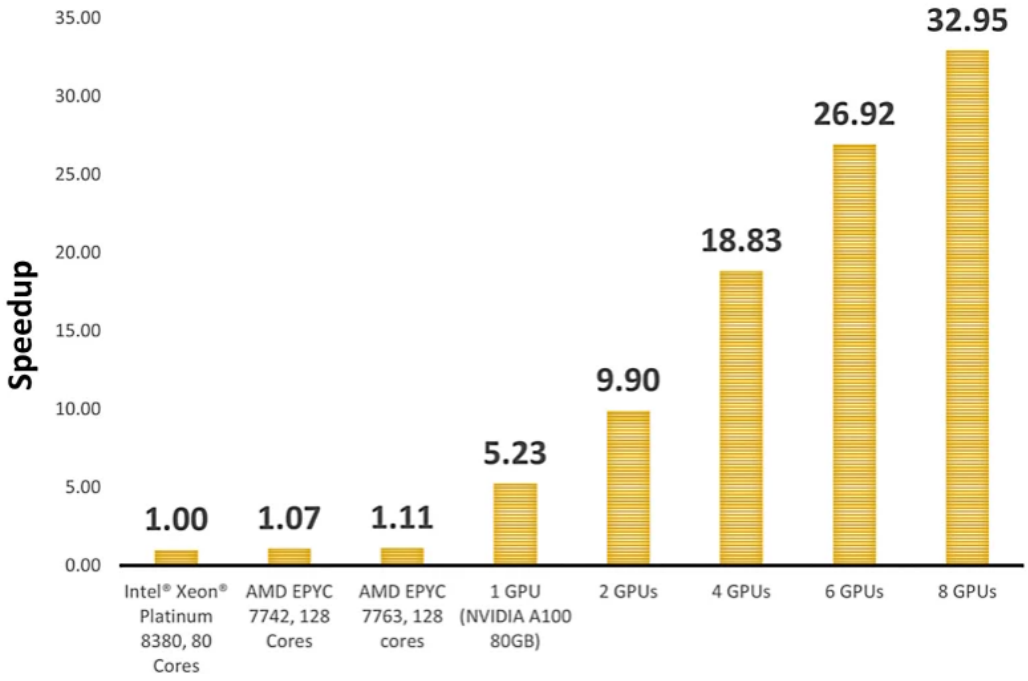


Figure 1.4: Speedup for different CPU and GPU clusters [4]

programs to be created and executed effectively. In the case of CFD, the memory requirement scales primarily with the number of elements in the computational domain, the order of the stored solution, and the complexity of the methods used. Therefore, high accuracy CFD simulations tend to require large amounts of memory.

In the case of CFD, graphics acceleration can provide significant improvement to computation times. This is due to the nature of CFD requiring several similar computations to be performed over a domain. However, depending on the methods used, this gain can be lessened or increased. The primary factor to consider is the element locality of the method. In other words, is the solution evaluated at each element independently with minimal impact from its neighbours, or is it dependent on information from larger portions of the domain. High element locality is desired when parallelizing and is therefore desirable when performing GPU acceleration. This indicates that, of the methods outlined in Chapter 1.3, the FR method among spatial discretization methods [71] and explicit time stepping methods, such as the explicit RK methods, among time stepping methods are especially suited to obtaining performance gains via GPU acceleration.

1.5 Objectives and Contributions

The P-ERK method has shown promise for reducing computation times while requiring significantly reduced resources. However, there is not an abundance of studies exploring this method. The objective of this study is to explore the capabilities of the P-ERK scheme including: accuracy, computational performance, and compatibility with various hardware configurations.

To achieve this objective, the formulation for the general P-ERK scheme will be presented. A subcritical flow over a sphere will be simulated. Instantaneous and time-averaged statistics will be collected. Comparisons will be drawn to reference numerical and experimental data as a means of validation. Performance data will be gathered for a variety of polynomial degrees and for a variety of hardware configurations. The collected performance data will be compared to the classical fourth order RK method.

The contributions of this work consists of furthering the body of knowledge associated with this novel time stepping scheme.

1.6 Thesis Outline

Chapter 1 explores the background and historical context of this work including: turbulence modelling approaches, spatial discretization, temporal discretization, trends in high-performance computing, and an overview of the challenges and objectives of this study.

Chapter 2 presents the theoretical concepts of fluid mechanics. The governing equations of fluid mechanics such as the conservation laws and the additional equation of state are explored including the mathematical manipulations that are required to employ them in a computational context.

Chapter 3 presents the various turbulence modelling approaches mathematical. The form of the governing equations is presented as well as their limitations.

Chapter 4 presents the spatial discretization scheme employed for this study. The theoretical basis for the method is explored primarily in its one-dimensional expression and its extension into many dimensions.

Chapter 5 presents the theoretical basis for explicit time discretization schemes such as: the classical Runge-Kutta schemes and the focus of this study, the Paired Explicit Runge-Kutta schemes.

Chapter 6 presents numerical results obtained in the context of this study to provide supporting evidence of the claims being made of both accuracy and performance.

Chapter 7 concludes this work with a summary and recommendations for future work.

Chapter 2

Governing Equations

Fluid behaviour is described by the following set of 3 fundamental laws. [72]

(1) Conservation of mass

(2) Conservation of momentum

(3) Conservation of energy

In this section, these governing equations will be presented in their integral and divergence forms.

2.1 Conservation of Mass

The law of conservation of mass states that for a closed system no mass may be created or destroyed [72]. This can be expressed mathematically as

$$\frac{dm}{dt} = 0. \tag{1}$$

According to the Reynolds transport theorem, two terms must be determined to compute this rate. First, the change in quantity within the control volume and second, the rate of flow into or out of the volume of the same quantity. Knowing that the amount of mass (m) per unit volume (V) also known as the density $\rho = \rho(\vec{x}, t)$, where $\vec{x}$ is position in space and t is time can be obtained from

$\frac{m}{V} = \rho$, it follows the total amount of mass in the volume is

$$m(t) = \int_V \rho \, dV. \tag{2}$$

Knowing the total mass contained in the volume, its rate of change can be obtained by taking its time derivative

$$\frac{dm}{dt} = \frac{d}{dt} \int_V \rho \, dV. \tag{3}$$

The rate of mass flow into the volume can itself be defined as the sum of the flow entering the volume from its surfaces. This can be expressed as

$$\oint_S \rho(\vec{v} \cdot \hat{n}) \, dS, \tag{4}$$

where S represents the surfaces bounding the volume, $\vec{v}$ represents the velocity of the incoming/outgoing flow, and $\hat{n}$ is the outward pointing normal vector to the surface. These quantities are illustrated in Figure 2.1. They can then be combined to obtain the final expression

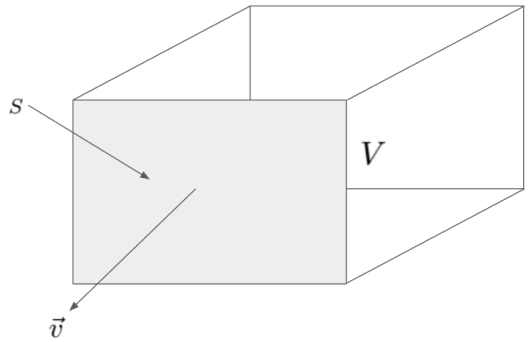


Figure 2.1: Control Volume

$$\frac{dm}{dt} = \frac{d}{dt} \int_V \rho \, dV + \oint_S \rho(\vec{v} \cdot \hat{n}) \, dS = 0. \tag{5}$$

2.2 Conservation of Momentum

The law of conservation of momentum is also know as Newton's second law, states that the sum of forces $(\sum \vec{F})$ applied to a body is equal to the time rate of change of its momentum [72]. For a constant mass it can be expressed simply as

$$\sum \vec{F} = m\vec{a}, \tag{6}$$

where $\vec{a}$ is the acceleration. However, in a fluid system the mass can change and so it must be expressed in the general form

$$\sum \vec{F} = \frac{d(m\vec{v})}{dt} \tag{7}$$

Using the Reynolds transport theorem, and recognizing that the momentum for a unit volume is $\rho\vec{v}$, it can be expressed as

$$\sum \vec{F} = \frac{d}{dt}\int_V \rho\vec{v}\, dV + \oint_S \rho\vec{v}(\vec{v}\cdot\hat{n})\, dS. \tag{8}$$

Having obtained this expression, $\sum \vec{F}$ must now be determined. This requires having knowledge of all the forces acting on the volume which includes surface and body terms. For the current application, body forces such as gravity will be ignored. Two primary surface forces are considered: pressure and shear. Firstly, the pressure (p) is a distributed force that acts over a surface and is always perpendicular to the surface it is affecting. As such it can be expressed as the following

$$\vec{F}_p = \oint_S -p\hat{n}\, dS. \tag{9}$$

Secondly, the viscous effects must be considered. Shear, unlike pressure, acts parallel to the surface it is affecting. It can be expressed using the Cauchy stress tensor (τ) which for a Newtonian fluid is defined as [73]

$$\tau = \mu(\nabla\vec{v} + (\nabla\vec{v})^T) - \frac{2}{3}\mu(\nabla\vec{v})\boldsymbol{I}, \tag{10}$$

were μ is the dynamic viscosity, and $\boldsymbol{I}$ is the identity matrix. From this expression, the total viscous force can be obtained by integrating it over the surface, obtaining

$$\vec{F}_v = \oint_S \tau \cdot \hat{n}\, dS. \tag{11}$$

Combining the terms computed above, conservation of momentum can be expressed as

$$\frac{d}{dt}\int_V \rho\vec{v}\, dV + \oint_S \rho\vec{v}(\vec{v}\cdot\hat{n})\, dS = \sum \vec{F} = \oint_S -p\hat{n}\, dS + \oint_S \tau\cdot\hat{n}\, dS\,. \tag{12}$$

This can be further simplified by grouping like terms together to obtain

$$\frac{d}{dt}\int_V \rho\vec{v}\, dV + \oint_S (\rho\vec{v}\otimes\vec{v} - \sigma)\cdot\hat{n}\, dS, \tag{13}$$

where

$$\sigma = -p\boldsymbol{I} + \tau. \tag{14}$$

2.3 Conservation of Energy

Conservation of energy states that the rate of change of energy (E) within a system is equal to the heat added to the system $(\dot{Q})$ minus the work produced by the system on its surroundings $(\dot{W})$ [72]. It can be expressed as

$$\frac{dE}{dt} = \dot{Q} - \dot{W}. \tag{15}$$

From the above, the rate of change of the energy in a unit volume (ρe) can be obtained from:

$$\rho e = \rho(c_v T + \frac{1}{2}(\vec{v}\cdot\vec{v})), \tag{16}$$

where e is the specific energy, c_v is the constant volume specific heat, and T is the temperature. Using the Reynolds transport theorem, the above equation can be written as

$$\frac{d}{dt}\int_V \rho e\, dV + \oint_S \rho e(\vec{v}\cdot+\hat{n})\, dS = \dot{Q} - \dot{W} \tag{17}$$

As with the conservation of momentum, knowledge of the terms on the right hand side is needed to resolve this equation. Considering that body forces are being neglected, work can only be generated by surface forces acting on the volume, such as pressure and viscosity, while heat can only be generated via thermal diffusion through conduction. As pressure acts always perpendicularly to the surface, its contribution to work can be expressed as

$$W_p = \oint_S (\vec{v} \cdot \hat{n}). \tag{18}$$

Knowing that shear acts parallel to the surface, and using the Cauchy stress tensor defined above, its contribution to work can be written as

$$W_v = -\oint_S \tau \cdot \vec{v}\, dS, \tag{19}$$

where work is defined as energy leaving the system, hence the negative expression. Heat transfer through convection can be expressed using Fourier's law. Therefore, the heat dispersed at any given point is given by:

$$\vec{q} = -k\nabla T, \tag{20}$$

where $\vec{q}$ is the heat diffusion at a given point, k is the thermal conductivity of the medium, and where ∇ is $\frac{d}{d\vec{x}}$. Taking into account that only the normal component of the heat flux enters the volume at any given point, the total heat flow for the surface can be obtained by:

$$\dot{Q} = \oint_S k\nabla T \cdot \hat{n}\, dS. \tag{21}$$

Combining the above equations, the conservation of energy can be expressed as

$$\frac{d}{dt}\int_V \rho e\, dV + \oint_S \rho e(\vec{v} \cdot \hat{n})\, dS = \dot{Q} - \dot{W} = -\oint_S p(\vec{v} \cdot \hat{n}) + \oint_S \tau\vec{v}\, dS + \oint_S k\nabla T \cdot \hat{n}\, dS, \tag{22}$$

which can be further simplified by grouping the surface terms together into

$$\frac{d}{dt}\int_V \rho e\, dV + \oint_S (\rho e\vec{v} + p\vec{v} - \tau\vec{v} - k\nabla T) \cdot \hat{n}\, dS = 0. \tag{23}$$

2.4 Equation of State

While the above laws describe fluid flow, there are too many variables to solve them explicitly, therefore an additional relation must be defined to be able to close them. Generally, the ideal gas law is used to resolve this issue. For fluids it can be described as

$$p = \rho R T \tag{24}$$

where (R) is the specific gas constant for the fluid [50]. While it is convenient to make use of this equation, it must be acknowledged that not all real gasses behave as ideal gasses. Notably, at high temperatures and high compression levels, the behaviour of real gasses deviates strongly from that described by the ideal gas law.

2.5 Compact Form

As all three laws above are obtained using the Reynolds transport theorem, they have similar forms. They all include the time derivative of a volume integral of the relevant quantity, and a set of surface integrals of the fluxes across the boundary. Therefore the equations can be combined into a set of vectors to represent them compactly [72]. The first vector will be the conserved quantities

$$\vec{w} = \begin{bmatrix} \rho \\ \rho\vec{v} \\ \rho e \end{bmatrix} . \tag{25}$$

The second, the tensor of the inviscid fluxes

$$\vec{F}_{inviscid} = \begin{bmatrix} \rho\vec{v} \\ \rho\vec{v} \otimes \vec{v} + p\boldsymbol{I} \\ \rho e\vec{v} + p\vec{v} \end{bmatrix} , \tag{26}$$

and the third being the tensor of the viscous fluxes:

$$\vec{F}_{viscous} = \begin{bmatrix} 0 \\ \tau \\ \tau\vec{v} - \vec{q} \end{bmatrix}. \tag{27}$$

Having established these tensor forms, the integral form of the Navier-Stokes equations can be written as:

$$\frac{d}{dt}\int_V \vec{w}\, dV + \oint_S \left[\vec{F}_{inviscid} - \vec{F}_{viscous}\right] \cdot \hat{n}\, dS = 0 \tag{28}$$

2.6 Divergence Form

These equations can be expressed in their integral form, as above, or in their divergence form [72]. The primary difference between these forms is related to their applicability. The integral form can be applied to a control volume, while the divergence form is applied to points. This means that for many applications the integral form is sufficient and in CFD it is applied when using the finite volume method. However, for many CFD applications (such as when using the DG or FR spatial discretization methods) the conserved variables are computed at points and therefore the divergence form of the equations is required. To convert from the integral form to the divergence form, the divergence theorem needs to be applied to each equation. The divergence theorem establishes an equivalence between volume integrals of a vector field and a surface integral of the same vector multiplied by the normal unit vector to the surface

$$\int_V \nabla\vec{F}\, dV = \oint_S \vec{F} \cdot \hat{n}\, dS. \tag{29}$$

2.6.1 Conservation of Mass

Applying the divergence theorem to Equation 5 the divergence form of conservation of mass can be obtained. The integral form of conservation of mass is

$$\frac{d}{dt}\int_V \rho\, dV + \int_V \nabla \cdot (\rho\vec{v})\, dV = 0. \tag{30}$$

Noting that in general derivatives and integrals are commutable and that both integrals in Equation 5 have the same bounds this can be compacted to

$$\int_V \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v})\, dV = 0. \tag{31}$$

For this equation to be valid other than in trivial cases, the integrand must satisfy the condition:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0. \tag{32}$$

2.6.2 Conservation of momentum

Applying the same manipulations as above to Equation 13 the divergence form of the momentum equation can be obtained

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{v} - \sigma) = 0. \tag{33}$$

2.6.3 Conservation of energy

Similarly, applying the above manipulations to Equation 23 the divergence form of conservation of energy is obtained

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \vec{v} + \rho \vec{v} - \tau \vec{v} - k \nabla T) = 0. \tag{34}$$

2.6.4 Compact Divergence Form

The divergence theorem can also be directly applied to the compact integral form in Equation 28 obtaining

$$\frac{\partial \vec{w}}{\partial t} + \nabla \cdot \left[\vec{F}_{inviscid} - \vec{F}_{viscous} \right] = 0. \tag{35}$$

Chapter 3

Turbulence Modelling

One of the major challenges when employing CFD to resolve the Navier-Stokes equations is the handling of turbulence. Extensive research has been conducted over time to develop methods to handle turbulence in more effective ways. For modern applications, three different approaches to turbulence modelling are used depending on the situation. Each of these methods has its advantages and disadvantages. They are the direct numerical simulation [11] method, the Reynolds-Averaged Navier-Stokes [22] method and, large eddy simulation [35] method.

3.1 Direct Numerical Simulation

The DNS method is technically not a modelling method at all. It consists of solving directly all scales of the flow from the smallest scales to the largest. This is the most intuitive and straightforward method of solving these equations.

Kolmogorov in his 1941 theory [74] proposed a method by which the smallest turbulent scales can be determined. Considering that turbulence attains its energy from the mean flow and that large eddies pass down their energy to smaller eddies at a rate (Π) of [75]

$$\Pi \approx u^3/\ell, \tag{36}$$

where ℓ is the integral scale and noting that for steady turbulence, the small scale rate of energy

decay ϵ must be equivalent to Π, it can be expressed as

$$\epsilon \approx \Pi \approx u^3/\ell. \tag{37}$$

Setting η to represent the characteristic length such that $v\eta/\nu \approx 1$ where ν is the kinematic viscosity, and knowing that the small scale dissipation rate can be expressed as

$$\epsilon \approx \nu v^2/\eta^2 \tag{38}$$

by eliminating v the following expression can be obtained

$$\eta = (\frac{\nu^3}{\epsilon})^{\frac{1}{4}} = (\frac{ul}{\nu})^{-\frac{3}{4}} l, \tag{39}$$

which is known as the Kolmogorov micro scale [75]. Knowing that the Reynolds number (Re) which is a measure of the relative strength of the viscosity [72] in a flow, can be expressed as

$$Re = \frac{ul}{\nu}, \tag{40}$$

the above identity can be rewritten as

$$\eta = Re^{-3/4} l. \tag{41}$$

This indicates that as the Reynolds number increases, the Kolmogorov scales become smaller. This is especially significant considering that to solve these scales the spatial step size (Δx) cannot be much larger than η. It follows, then, that the minimum approximate number of points for a three-dimensional square mesh using DNS can be obtained from

$$N_x^3 \approx \left(\frac{L_{box}}{\Delta x}\right)^3 \approx \left(\frac{L_{box}}{l}\right)^3 Re^{(\frac{3}{4})^3}, \tag{42}$$

where Nx is the number of grid points in any given direction and L_{BOX} is the flow field size in a given direction. As this expression will only give minimal resolution sufficient to capture a single

eddy it stands that

$$Re \leq N^{\frac{4}{3}}, \tag{43}$$

is needed to accurately capture flow behaviour at these scales. This is problematic as for moderate to high Reynolds numbers the amount of points required scales very poorly. This is especially challenging when considering that in Orzarg's work [11] the Reynolds number was only $Re = 20$ while the ranges that are of interest to engineers range from such small values to $Re \geq 10^6$. Because of these limitations, even with modern hardware, DNS is only used for a relatively small range of problems.

3.2 Reynolds-Averaged Navier-Stokes

The Reynolds-Averaged Navier-Stokes method is the first major turbulence modelling framework that was developed and it is still in use today. This method, models all turbulent scales using a turbulence model obtaining a steady-state solution. The NS equation is then computed in a time-averaged form. By applying Reynolds averaging to the velocity, its mean and fluctuating components are separated obtaining

$$u(\overrightarrow{x}, t) = \overline{u(\overrightarrow{x})} + u(\overrightarrow{x}, t)', \tag{44}$$

where $u(\overrightarrow{x}, t)$ is the instantaneous velocity,$\overline{u(\overrightarrow{x})}$ is the time-averaged velocity and $u(\overrightarrow{x}, t)'$ is the velocity fluctuation. Noting the properties of Reynolds averaging include 1) time-averaging being a linear process

$$\overline{\alpha u + \beta v} = \alpha \overline{u} + \beta \overline{v}. \tag{45}$$

2) Time-averaging of integrals and derivatives commutes

$$\overline{\int u \, dy} = \int \overline{u} dy, \qquad \overline{\frac{\partial u}{\partial t}} = \frac{\partial \overline{u}}{\partial t}. \tag{46}$$

3) the time-average of the fluctuation of a variable is always zero

$$\overline{u'} = 0, \tag{47}$$

and 4) the average of a product is obtained as

$$\overline{uv} = \overline{\overline{u}\overline{v}} + \overline{u'v'}, \tag{48}$$

from which it follows that

$$\overline{\overline{u}} = \overline{u}, \qquad \overline{u'\overline{v}} = 0, \tag{49}$$

and, applying the same method to the momentum equation the time-average form for incompressible Navier-Stokes

$$\rho\left(\frac{\partial\overline{v_i}}{\partial t} + \frac{\partial}{\partial x_j}(\overline{v_j}\overline{v_i})\right) = -\frac{\partial\overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}(\overline{\tau}_{ji} - \rho\overline{v'_j v'_i}), \tag{50}$$

can be obtained [72] where $\rho\overline{v'_j v'_i}$ is known as the Reynolds stress. For a three-dimensional flow case, 6 Reynolds stresses are generated which are unknowns that must be solved in addition to the variables inherent to the governing equations in 3D to close the system of equations. This is challenging as in this current form the system of equations is under defined. Various turbulence models have been developed to address this issue. The selection of an appropriate turbulence model is the primary challenge when using RANS.

It is important to note that by the very nature of time-averaging, the RANS method can only ever produce accurate results for steady-state flow behaviour. For this reason it struggles when simulating transient behaviours such as boundary layer separation, vortex shedding, and stall.

3.3 Large Eddy Simulation

Large Eddy Simulation serves as a middle ground between DNS and RANS. LES solves most spatial scales directly while using filtering methods to truncate the very small scale behaviours. This can be seen clearly in Figure 3.1 below. This means that this method can provide information

on transient behaviours, unlike RANS, but at a significantly reduced cost compared to DNS. The

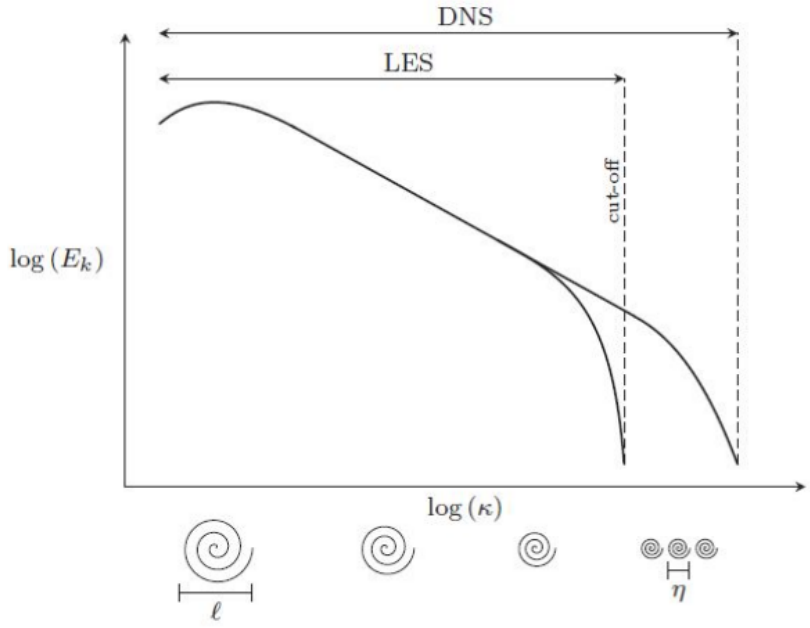


Figure 3.1: Turbulent kinetic energy cascade [5]

form of the NS equation used by LES is in many ways similar to that of RANS. Whereas RANS decomposes the velocity field into its mean and fluctuating components, LES decomposes it into filtered and residual components

$$u = \bar{u} + u', \tag{51}$$

where u is the velocity field, $\bar{u}$ is the filtered velocity, and u' is the residual. While RANS time-averages the NS equations, LES uses spatial filtering to obtain, for incompressible Navier-Stokes

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial (u_i \bar{u}_j)}{\partial x_j} = \frac{1}{\rho}\left(\frac{\partial \tau_{ij}^R}{\partial x_j} - \frac{\partial \bar{p}}{\partial x_i}\right) + \nu \nabla^2 \bar{u}_i, \tag{52}$$

where

$$\tau_{ij}^R = \rho[\bar{u}_i \bar{u}_j - u_i \bar{u}_j], \tag{53}$$

is the residual stress tensor. The residual stress tensor is analogous the the Reynolds stresses found in the RANS equations and must be modeled somehow [75]. More importantly though, LES only

requires modelling for a small fraction of turbulent length scales, whereas RANS must model the entire turbulent cascade. Generally a sub-grid-scale (SGS) turbulence model is used such as those developed by Leonard [40]. However, the implicit large eddy simulation approach allows the truncation error of the method to resolve these SGS eddies instead [45]. The turbulence modelling approach implemented in the the High-Order Unstructured Solver (HORUS) used for this study is an ILES method.

Chapter 4

Spatial Discretization

4.1 Flux Reconstruction

While there are many methods of spatial discretization that have been proposed, such as those outlined earlier in Chapter 1.3.1, in this study the flux reconstruction approach initially proposed by H. T. Huynh [49] is of interest. Its one-dimensional and multi-dimensional expressions will be provided in the following section. A special characteristic of the FR scheme is its ability to recover several different high order methods such as DG, SV, and SD [49, 76]. It was further extended into 2D by Wang and Gao [76], and into 3D by both Haga et al using the Lifting Collocation Penalty (LCP) formulation [77], and by Wang and Gao using the Correction Procedure via Reconstruction (CPR) scheme [76]. Additionally, Vincent et al extended the 1D formulation to develop a class of energy stable FR methods [78] which were later extended into 3D by Williams and Jameson [79]. Many other contributions include: Romero and Jameson's extensions to triangular elements [80], Vermeire and Vincent's applications to ILES [81, 82], Vermeire and Nadarajah's application to adaptive IMEX schemes [65], Vermeire et al's application to turbulent flow over an airfoil [83], Pereira and Vermeire's work on hybridized flux reconstruction schemes [84], and others.

4.1.1 One-Dimensional

A one dimensional conservation law can be written as

$$\frac{\partial u}{\partial t} + \frac{\partial F}{\partial x} = 0, \tag{54}$$

where u represents the conserved variables and F represents the fluxes of the variables. The FR approach discretizes equations of this form by first dividing the computational domain (Ω) into N_i non-overlapping elements (Ω_i) where $x \in [x_{i,L}, x_{i,R}]$ are respectively the left and right bounds of each element as shown in Figure 4.1. Then, for simplicity every element Ω_i is mapped from the

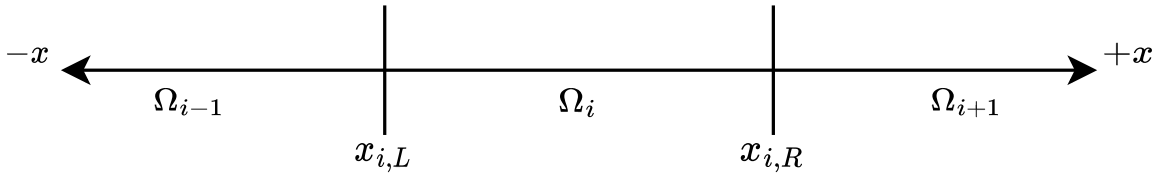


Figure 4.1: 1D discretization

physical space $x \in \mathbb{R}$ to a reference space $\xi \in [-1, 1]$ were the reference element Ω_i' is used. This transformation is performed by

$$x(\xi) = \frac{1}{2}(x_{i,L} + x_{i,R} + \xi h_i) \rightleftharpoons \xi(x) = \frac{2}{h_i}\left(x - \frac{x_{i,L}, +x_{i,R}}{2}\right), \tag{55}$$

where h_i represents the size of the element in physical space. The PDE is then converted using:

$$\frac{\partial \xi}{\partial x} = \frac{2}{h_i} \rightleftharpoons \frac{\partial x}{\partial \xi} = \frac{h_i}{2} \tag{56}$$

obtaining

$$\frac{\partial u_i}{\partial t} + \frac{2}{h_i}\frac{\partial F_{x,i}}{\partial \xi} = 0. \tag{57}$$

Within each element, the solution is defined at N_p points determined by the desired polynomial degree P^o of the solution where the number of points is defined by $N_p = P^o + 1$. The fluxes, in turn, are approximated by a degree $P^o + 1$ polynomial. While the solution within each element must be continuous, there is no such requirement at the element edges. Therefore the solution polynomial

can be written as

$$u_i^{\delta}(\xi, t) = \sum_{n=1}^{N_i} u_{i,n}\phi_n(\xi), \tag{58}$$

where $u_{i,n}$ is the solution at one of the solution points N_i and $\phi_n(\xi)$ is the nodal basis function expressed as

$$\phi_n(\xi) = \prod_{n=1, i\neq n}^{N_i} \frac{\xi - \xi_n}{\xi_i - \xi_n}. \tag{59}$$

Figure 4.2 shows an example set of nodal basis functions using equidistant points. Given that the solution points for every element are at the same relative position in reference space, the nodal basis functions only need to be generated once for the entire solution as they will be identical. Being now able to solve the continuous solution within any given element, the flux within the elements must be determined. Using a similar method, the flux can be obtained by

$$F_i^{\delta}(\xi, t) = \sum_{n=1}^{N_i} F_{i,n}(t)\phi_n(\xi), \tag{60}$$

where $F_{i,n}(t)$ is the flux computed at a point N_i. As mentioned previously, the fluxes at the edges of each element are not necessarily continuous. Therefore, they must be stitched to generate a continuous flux over the entire domain that approximates the discontinuous fluxes while maintaining continuity across element boundaries. This flux will have a polynomial degree of $P^o + 1$ and be computed at $N_i + 1$ points. The correction function used to stitch these is the following:

$$\zeta_i = \left[F_{i,L}^* - F_i^{\delta}(-1)\right] \boldsymbol{g}_L(\xi) + \left[F_{i,R}^* - F_i^{\delta}(1)\right] \boldsymbol{g}_R(\xi), \tag{61}$$

which results in a final relation between the fluxes described by

$$F_i^c = F_i^{\delta} + \zeta_i, \tag{62}$$

where F_i^c is the continuous flux on an element, F_i^* is the common interface flux at either boundary,

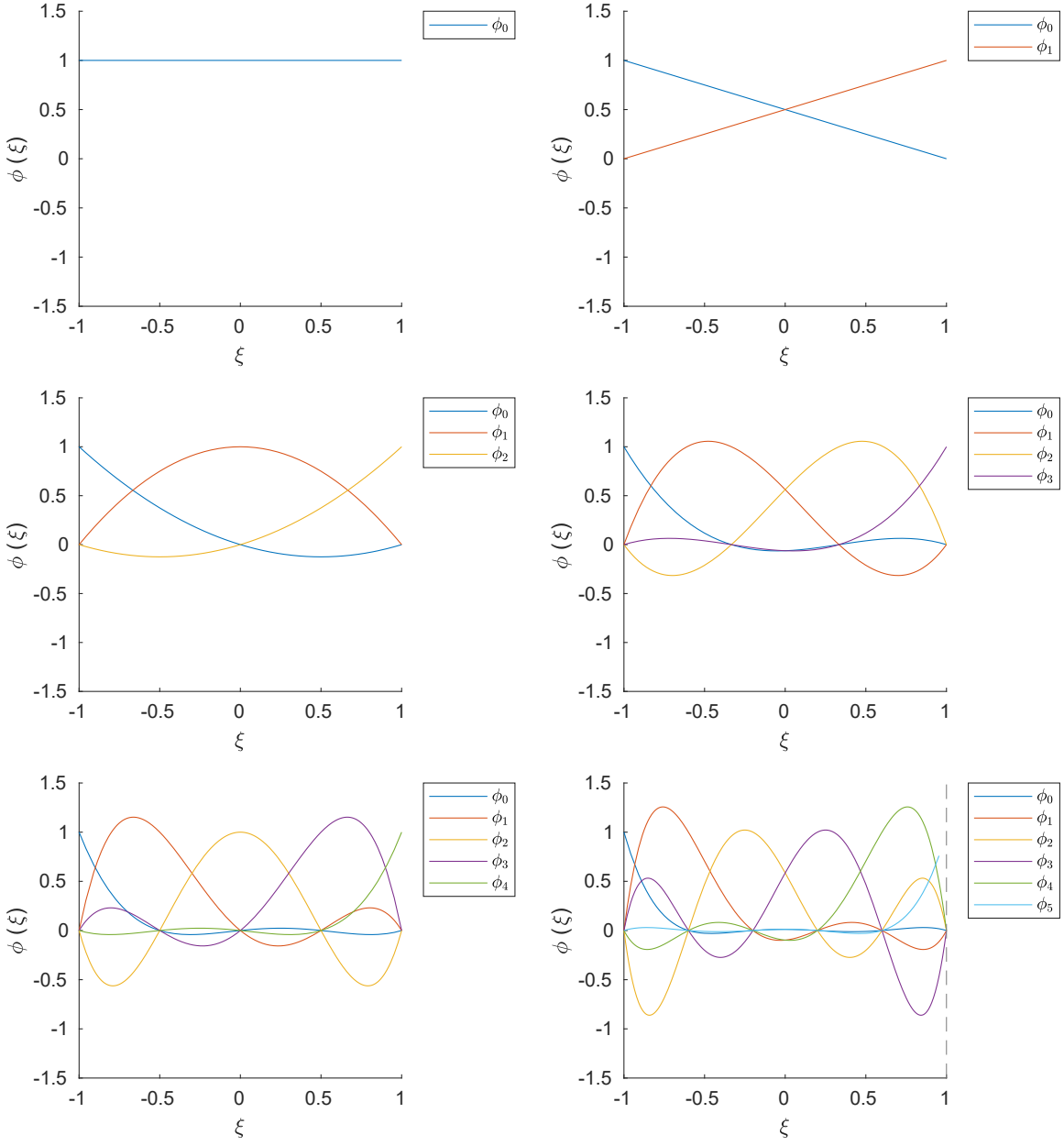


Figure 4.2: 1D Nodal basis functions for P_0^o to P_5^o

and the correction functions $\boldsymbol{g}$ must respect:

$$g_L(-1) = 1, \qquad g_L(1) = 0, \tag{63}$$

$$g_R(-1) = 0, \qquad g_R(1) = 1, \tag{64}$$

and, must be of polynomial degree P^o+1. The common fluxes must be determined using a Riemann solver. There are many possible schemes, such as upwind or central schemes that introduce different amounts and types of error to the final solution. Therefore, an appropriate solver must be selected for the application. Depending on the correction function used, various different schemes can be recovered such as SD and SV. Huynh [49] describes how Radeau polynomials can be used to recover DG schemes. The left Radeau polynomials describe g_R and the right polynomials describe g_L. They can be obtained from the normalized Legendre polynomials shown in Figure 4.3 by the following identity

$$g = -1^{i-1}\frac{1}{2}(S(i) - S(i-1)), \tag{65}$$

where g is the Radau polynomial, i is the polynomial degree and S is the normalized Legendre

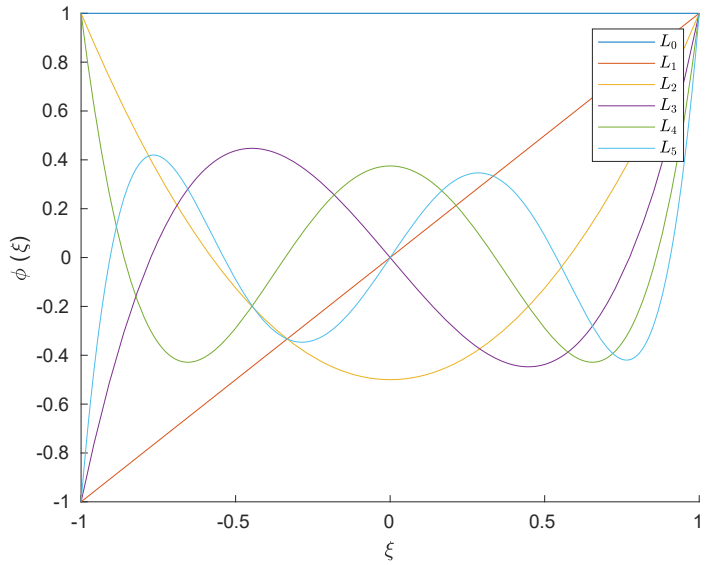


Figure 4.3: Normalised Legendre polynomials

polynomial. The Radau polynomials are visualized in Figure 4.4.

Replacing the discontinuous flux with the continuous flux in the conservation law formulation

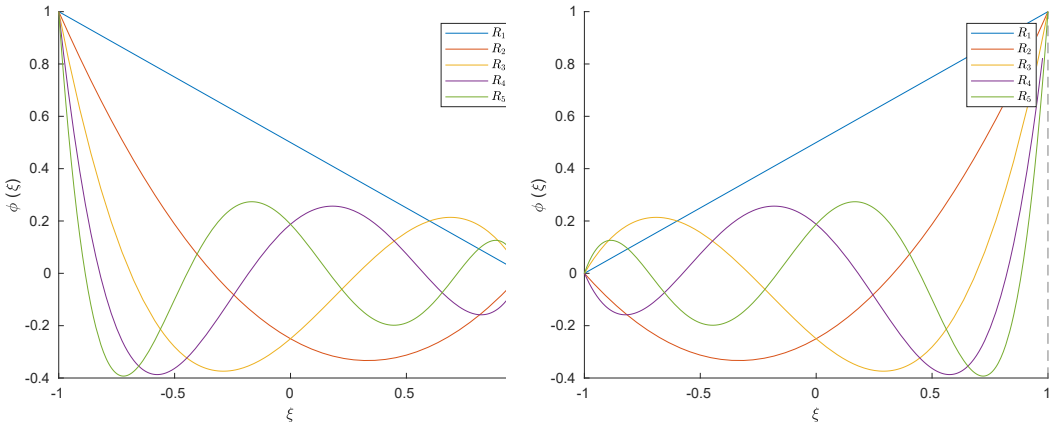


Figure 4.4: Radau polynomials

for a reference element results in:

$$\frac{\partial u_i}{\partial t} + \frac{2}{h_i}\frac{\partial}{\partial \xi}(F_i + \zeta) = 0. \tag{66}$$

4.1.2 Multi-Dimensional

This method was initially introduced only for 2D quadrilateral cases in Huynh's paper [49]. It is through the works of Wang and Gao [76] and Haga et all [77] that it was expanded in general multiple dimensions. The conservation law can be expressed as

$$\frac{du}{dt} + \nabla \cdot \vec{F} = 0. \tag{67}$$

Similar to the 1D case, it is mapped into N_i number of non-overlapping elements, Ω_i. Assuming square elements, and using a reversible mapping function such that

$$(\vec{x}) = Fm_i(\vec{\xi}) \leftrightarrow (\vec{\xi}) = Fm_i^{-1}(\vec{x}) \tag{68}$$

where Fm is the mapping function and $\vec{\xi}$ is the variable representing $\vec{x}$ in the reference space. Using a polynomial basis representation, the solution polynomial is expressed as

$$u_i^{\delta}(\vec{\xi}, t) = \sum_{m,n=1}^{N_i} u_{i,n}\phi_n(\vec{\xi}). \tag{69}$$

Hence, the discontinuous fluxes can be expressed as

$$\vec{F}_i^{\delta}(\vec{\xi}, t) = \sum_{n=1}^{N_i} \vec{F}_{i,n}(t)\phi_n(\vec{\xi}). \tag{70}$$

As with the one-dimensional formulation, these discontinuous fluxes must be stitched into a continuous form using a correction function ζ_i described as:

$$\zeta_i = \frac{1}{|\Omega_i|} \sum_{f \in S} \sum_{l} a_{i,f,l} [\tilde{F}]_{f,l} S_f, \tag{71}$$

where Ω_i represents the element size, $a_{i,f,l}$ represents the constant lifting coefficients for any geometry, $[\tilde{F}]_{f,l}$ represents the flux difference at point l along boundary f, and S_f represents the the dimensions of the boundary. This results in the conservation law being written as

$$\frac{du}{dt} + \nabla \cdot \vec{F}_i + \zeta_i = 0. \tag{72}$$

It is important to note that the value used for $a_{i,f,l}$ will affect what type of scheme (such as DG, SV and, SD) is recovered. When using this method with the Navier-Stokes equations it is important to consider the inviscid and viscous components separately. This is because of their vastly different behaviours. As such, HORUS makes use of the Rusanov Riemann solver [85] for the inviscid fluxes and the BR2 solver [86] for the viscous fluxes. All of these are evaluated at the Gauss-Legendre points for tensor product elements.

Chapter 5

Time Discretization

As outlined in Chapter 1.4 there are many types of time discretizations that have been proposed and they all have certain advantages and disadvantages. For the purposes of this study the focus is on a comparison of the fourth order explicit Runge-Kutta method with Paired Explicit Runge-Kutta methods.

5.1 Explicit Runge-Kutta Methods

Runge-Kutta methods are a family of numerical methods extending from the first-order Euler method. The Euler method can be simply expressed as

$$u_{i+1} = u_i + \Delta t R, \tag{73}$$

where u is any conserved variable dependent on the variable t, and R is the residual of u. The equation is simply repeated to find the solution of any n future time steps. Runge-Kutta methods can be obtained by expanding on the Euler method by adding intermediate stages. These break down the time step into smaller stages, using the previous stages to compute future stages and by applying coefficients to the stage residuals and summing them to obtain the solution for the time

step. These coefficients are generally represented in a Butcher tableau of the form [87]

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}, \tag{74}$$

were A is a matrix of coefficients representing the contribution of each stage to others, c is a vector of coefficients representing the relative stage time, and b is a vector of quadrature weights representing the contribution of each stage residual to the final updated solution. The simplest such tableau is the one representing the Euler method described above. It is of the form

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}. \tag{75}$$

For the tableau to generate a first-order accurate method, it must satisfy the consistency condition. In other words the sum of the elements of b must be equal to 1 which can be expressed mathematically as

$$\sum_{j=1}^{N_s} b_i = 1, \tag{76}$$

where N_s is the total number of stages. For a method to be explicit, every stage can only rely on previous stages and, therefore, the upper triangular of the matrix is generally left blank to indicate the 0 values located there. Therefore, any explicit Runge-Kutta method can be expressed by the following

$$u_i = u_n + \Delta t \sum_{j=1}^{N_s} a_{i,j} R(t_n + c_j \Delta t, u_j) \tag{77}$$

where u_i is the intermediate value of the stage, and, $a_{i,j}$ is the corresponding coefficient in the A matrix. Hence, the value at the end of the step is obtained from

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{N_s} b_i R(t_n + c_i \Delta t, u_i). \tag{78}$$

From these tableaus, the order of accuracy of these methods can be determined. For any scheme of order P^t, it must satisfy $2n - 1$ constraints including the consistency condition. The additional

conditions for orders 2 to 4 are as follows

$$\sum_{i=1}^{N} b_i c_i = \frac{1}{2},$$
$$\sum_{i=1}^{N} b_i c_i^2 = \frac{1}{3}, \sum_{i=1}^{N}\sum_{j=1}^{N} b_i a_{i,j} c_i = \frac{1}{6},$$
$$\sum_{i=1}^{N} b_i c_i^3 = \frac{1}{4}, \sum_{i=1}^{N}\sum_{j=1}^{N} b_i c_i a_{i,j} c_j = \frac{1}{8}, \quad \sum_{i=1}^{N}\sum_{j=1}^{N} b_i a_{i,j} c_j^2 = \frac{1}{12}, \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} b_i a_{i,j} a_{j,k} c_i = \frac{1}{24}.$$

The classical Runge-Kutta method (also know as RK44) is of primary interest as a reference method for this study. Its Butcher tableau is the following:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{79}$$

It can be seen from this tableau that this method is an explicit fourth-order accurate method as it meets all the criteria outline above for a fourth order accurate scheme and its upper diagonal components are zero.

5.1.1 Stability

Taking a system of differential equations of the form [87]

$$x'(t) = \boldsymbol{q}x(t) \tag{80}$$

where $\boldsymbol{q}$ is a constant matrix, $z = \Delta t \boldsymbol{q}$, and Δt is the step size; the exact solution will increase by a factor of $exp(z)$ while the solution computed using an RK method will increase by a stability function $f(z)$ that is unique to the given scheme. The stability region can then be expressed as [87]

$$z \in \mathbb{C} : |f(z)| \leq 1. \tag{81}$$

As z is a function of h, the step size used must be selected such that the above condition is respected. This in turn means that these methods cannot be unconditionally stable. From [87] the stability functions for RK methods of orders $P^t = 1$ to $P^t = 5$ are as follows:

$$\begin{aligned} 1 + z & \qquad P^t = 1, \\ 1 + z + \tfrac{1}{2}z^2 & \qquad P^t = 2, \\ 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{6}z^3 & \qquad P^t = 3, \\ 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{6}z^3 + \tfrac{1}{24}z^4 & \qquad P^t = 4, \\ 1 + z + \tfrac{1}{2}z^2 + \tfrac{1}{6}z^3 + \tfrac{1}{24}z^4 + \tfrac{1}{120}z^5 + Cz^6 & \qquad P^t = 5, \end{aligned} \tag{82}$$

The C coefficient for the $P^t = 5$ order is dependent on the specific method used. The stability regions can be visually represented as in Figure 5.1. It can be seen from the figure that when using the standard method of building RK schemes, part of the stability region is on the right side of the origin. As systems with roots in the right half plane are unstable, part of the theoretical stability region cannot be used to obtain a stable method. Thus, part of the stability gain achieved by higher order methods is effectively wasted [88]. Achieving these higher orders following this approach imposes an increasing number of simultaneous computations for each step, making the computational time and memory cost associated with it high. For this reason, it is interesting to find variations for the high order methods that maximize the useful stability region all the while minimizing the computational cost by using a similar or smaller number of stages.

5.2 Paired Explicit Runge-Kutta Method

The Paired Explicit Runge-Kutta method (P-ERK) first proposed by Vermeire [66] aims to overcome shortcomings with traditional Runge-Kutta schemes, such as high memory requirements to store the stage residuals R, issues with stability scaling to higher orders, and relatively high computational cost to achieve a desired accuracy order. This first issue can be mitigated by modifying the

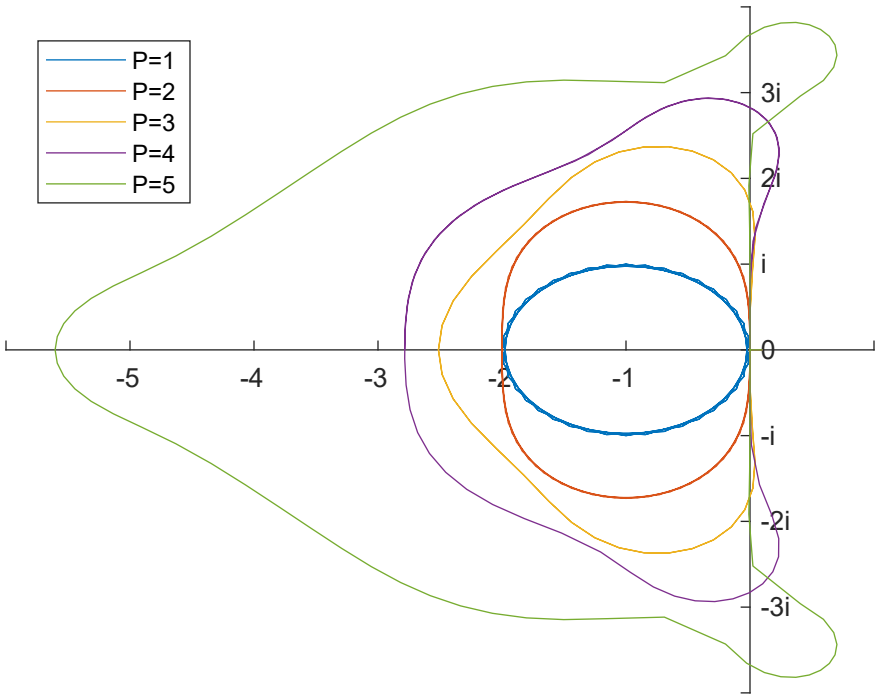


Figure 5.1: Stability regions of RK methods from $P = 1$ to $P = 5$

traditional method by setting

$$a_{i,j} = 0, \qquad j > 1, \qquad i > j + 1,$$
$$b_i = 0, \qquad i < n,$$
$$b_n = 1.$$

The corresponding Butcher tableau then takes on the following form

$$\begin{array}{c|cccccc} 0 & 0 & & & & & \\ c_2 & a_{2,1} & 0 & & & & \\ c_3 & a_{3,1} & a_{3,2} & 0 & & & \\ c_4 & a_{4,1} & 0 & a_{4,3} & \ddots & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \\ c_n & a_{n,1} & 0 & 0 & \cdots & a_{n,n-1} & 0 \\ \hline & 0 & 0 & 0 & \cdots & 0 & b_n \end{array} \tag{83}$$

This means that only two stages need to be stored per time step, greatly reducing the memory overhead of the method. It is equally important that the method maintain its order of accuracy. This can be achieved by respecting the conditions outlined above for RK methods of various orders. Hence, setting

$$c_n = \frac{1}{2}, \tag{84}$$

will ensure that the method is second-order accurate. Then choosing the remain c_i to be evenly spaced between the first and last stages

$$c_i = \frac{i-1}{2(n-1)}, \tag{85}$$

the general tableau above is simplified to the following

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1} & 0 & & & & \\
c_3 & c_3 - a_{3,2} & a_{3,2} & 0 & & & \\
c_4 & c_4 - a_{4,3} & 0 & a_{4,3} & \ddots & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & 0 & \\
c_n & c_n - a_{n,n-1} & 0 & 0 & \cdots & a_{n,n-1} & 0 \\
\hline
 & 0 & 0 & 0 & \cdots & 0 & b_n
\end{array} \tag{86}$$

where each c_i term is determined from

$$c_i = \sum_{j=1}^{n} a_{i,j}. \tag{87}$$

Therefore, for every additional stage that is added, only a single additional computation is required which is similar to the traditional RK method. Additionally, this method can be further modified to use e active stages out of the n total stages defined in the Butcher tableau within a single system and is represented as P-ERK$_{n,e,2}$ This is done by setting

$$a_{i+1,i} = 0, 2 \leq i \leq n - e + 1, \tag{88}$$

which results in the scheme having $e - 2$ unknown coefficients. Obtaining the stability polynomial by the method outlined by Hairer [89], resolves these unknown coefficients. An example stability

ıed

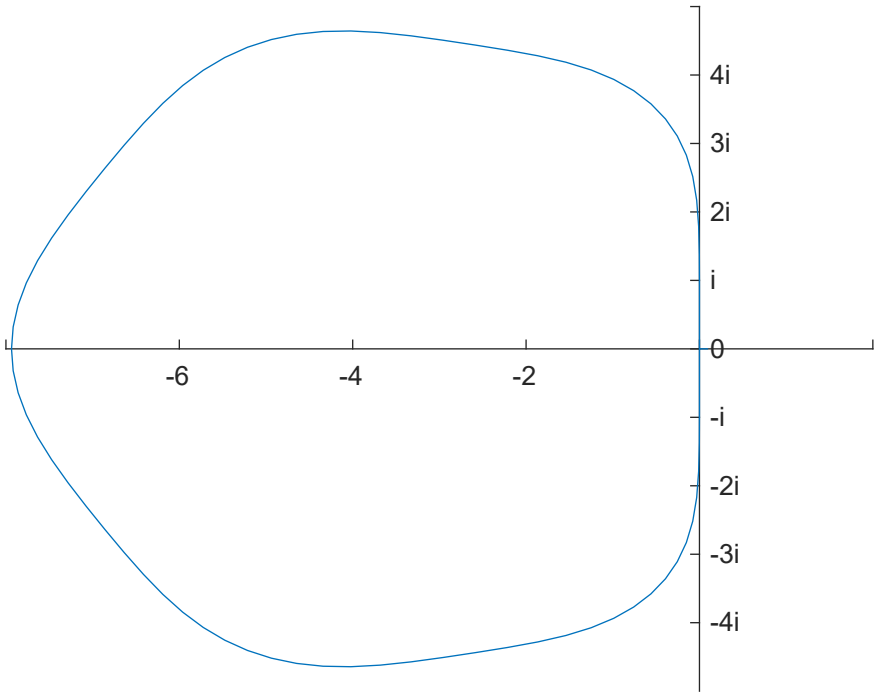


Figure 5.2: Stability region for a P-ERK$_{6,6,2}$ scheme

from [66] and were optimized in such a way as to ensure that the maximum stability region will be located in the usable side of the axis enclosing the spectra of the spatial discretization. Noting that the $\boldsymbol{b}$ and $\boldsymbol{c}$ vectors are identical for any P-ERK method having the same total amount of stages, methods with varying amounts of active stages can therefore be stitched together at points of contact. This is especially useful when dealing with systems that are only locally stiff such as most CFD problems. That is because this enables the use of high-order RK methods in the stiff regions while the non-stiff regions use lower-order methods, further reducing the computational cost of the overall solution.

While the utility of P-ERK methods has been established on CPUs, their utility for GPU accelerated simulations has yet to be established. The objective of the following sections is to explore this topic via applying P-ERK to LES simulations using FR as a spatial method on GPUs.

Chapter 6

Low Reynolds Flow Over a Sphere

6.1 Background

In CFD there are number of flow conditions that pose particular challenges for RANS. These include cases such as shock capturing, stall, and low Re flow. For this study, sub-critical flow over a sphere is explored. There have been many experimental [90, 91, 92, 93] and numerical [94, 95, 96, 97, 98] analyses of this case using a variety of computational methods. Through these experiments, this particular test case is well-defined, and thus it's an ideal candidate to asses GPU accelerated LES using P-ERK. As this is a sub-critical flow experiment, the viscous forces are important in determining the flow behaviour, making it especially pertinent to study the following quantities: lift and drag, boundary layer separation, vortex shedding, and pressure coefficient.

6.2 Case Description

The solver used to perform this experiment was HORUS which makes use of an FR approach for spatial discretization evaluated at the Gauss-Legendre points for tensor product elements. The Riemann solvers used to compute fluxes are the Rusanov solver for inviscid fluxes and the second method of Bassi and Rebay CBR2 solver for the viscous fluxes [99].

The case consists of simulating three-dimensional sub-critical flow over a smooth sphere. The flow was defined using a Reynolds number Re of 3,700, a Mach number (M) of 0.2, a Prandtl

number (Pr) of 0.71, and a ratio of specific heats (γ) of 1.4. At the initial state, velocities were set to zero so that the flow would develop naturally as time progresses. All parameters used in the simulation were non-dimensionalized.

The geometry was simulated using 88872 quadratically curved hexahedral elements. The elements have a higher density near the sphere and a reduced density at the far field. Figure 6.1 shows the discretization of the domain. The computational domain dimensions can be seen in Table 6.1. A

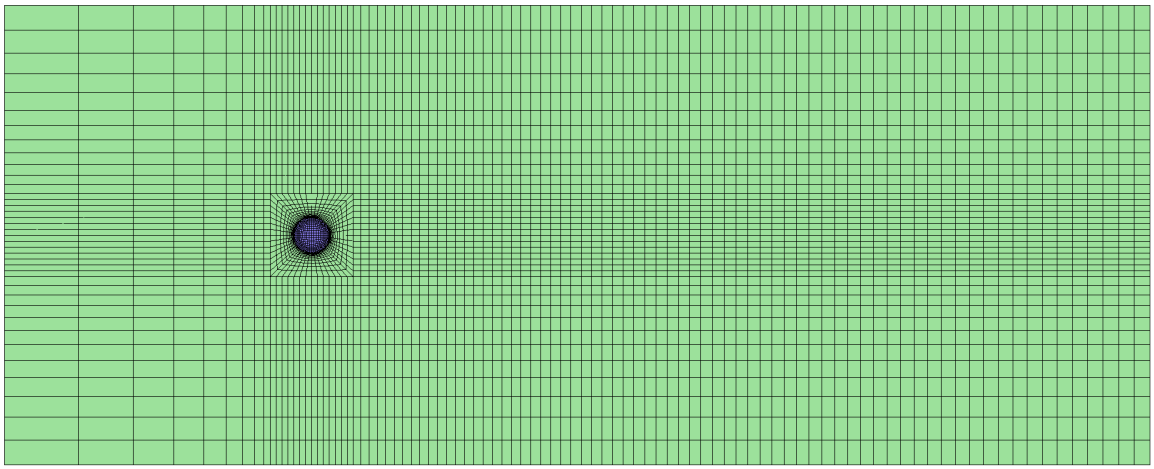


Figure 6.1: Computational domain mesh

	x	y	z
-	8	6	6
+	22	6	6

Table 6.1: Computational domain dimensions

P-ERK$_{16,16,2}$ scheme with 11,760 elements using all active stages and 77,112 elements P-ERK$_{16,3,2}$ using three active stages was used as the time time discretization scheme for this study. The number of elements in the mesh was adjusted to make maximal use of the memory available when using a solution polynomial degree of five. Figure 6.2 shows the distribution of these schemes.

Simulations were run on a general purpose cluster belonging to the Digital Research Alliance of Canada known as Narval. The computing capabilities of this cluster are outlined in Table 6.2. The experiment was run on 1, 2, 4, 8, 16, and 32 GPUs using 12 CPU cores per task.

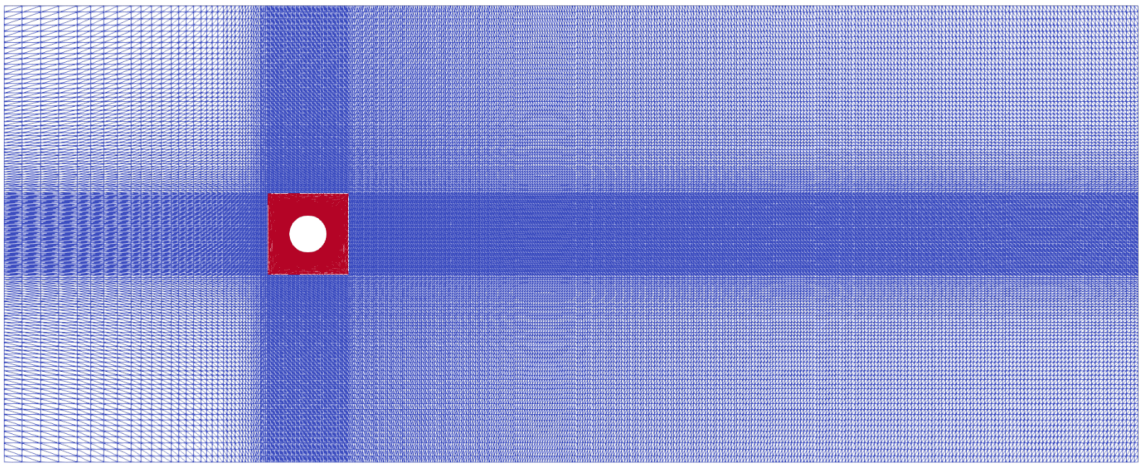


Figure 6.2: Solution timelevel distribution

nodes	cores	memory	CPU	storage	GPU
1145	64	249G or 255000M	2 x AMD Rome 7532 @ 2.40 GHz 256M cache L3	1 x 960G SSD	-
159	48	498G or 510000M	2 x AMD Milan 7413 @ 2.65 GHz 128M cache L3	1 x SSD of 3.84 TB	4 x NVIDIA A100 (40 GB memory)

Table 6.2: Narval node characteristics

6.3 Performance

As the primary objective of developing the P-ERK schemes was to acquire faster computational times for similar levels of accuracy and resources, it is important to evaluate its performance not only for a single configuration, but also to evaluate how it may scale with various configurations. This study therefore evaluates how performance will scale for various solution polynomial degrees, and for different resource configurations, namely the number of GPUs and CPU cores being used.

Figure 6.3 shows that for polynomial degrees of one to five, the P-ERK scheme is advantageous over the classic RK method. It shows that the advantage of the P-ERK scheme increases as the polynomial degree of the solution increases. This is significant as it shows that the performance of the P-ERK method scales logarithmically and would support its application to still higher orders for strongly locally stiff problems. This scaling is similar for both CPUs and GPUs, with one CPU core taking approximately 200 times longer than one GPU for the same polynomial degree. This can also be observed by looking at the speedup factor in Table 6.3,. which is defined as the ratio of

the previously mentioned time ratios. This shows that for the polynomial degrees studied, there is an average speedup factor of 4.73 using GPUs and of 5.82 using CPUs when attempting to reach a solution with the P-ERK$_{16,e,2}$ scheme compared to the performance of the traditional fourth order RK method.

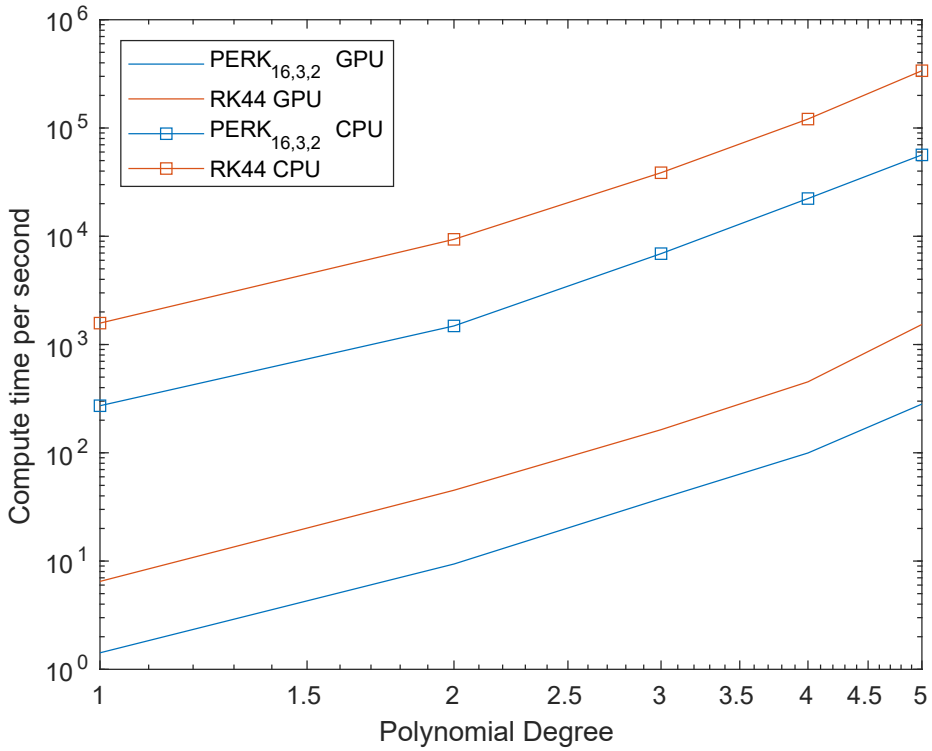


Figure 6.3: Clock/simulation time VS polynomial degree

	P^o	1	2	3	4	5
Speedup factor	GPU	4.56	4.80	4.33	4.54	5.44
	CPU	5.8	6.31	5.59	5.42	5.99

Table 6.3: Speedup factor for scaling polynomial degree

CFD is applied to a wide variety of problems, and the computational power available to run these schemes also varies widely. From low-complexity works running on single computers to high-complexity works running on multiple nodes it is important to identify where these new time stepping schemes are most applicable. As parallelizing has been showm previously to be an effective method to improve computation time, it becomes interesting to ascertain whether the P-ERK method scales better with increasing parallel tasks as opposed to the traditional method. It is especially

important to ascertain if, within the realistic range of hardware configurations, there is a point where the traditional method surpasses the P-ERK method. Figure 6.4 shows the resource performance scaling for these two methods.

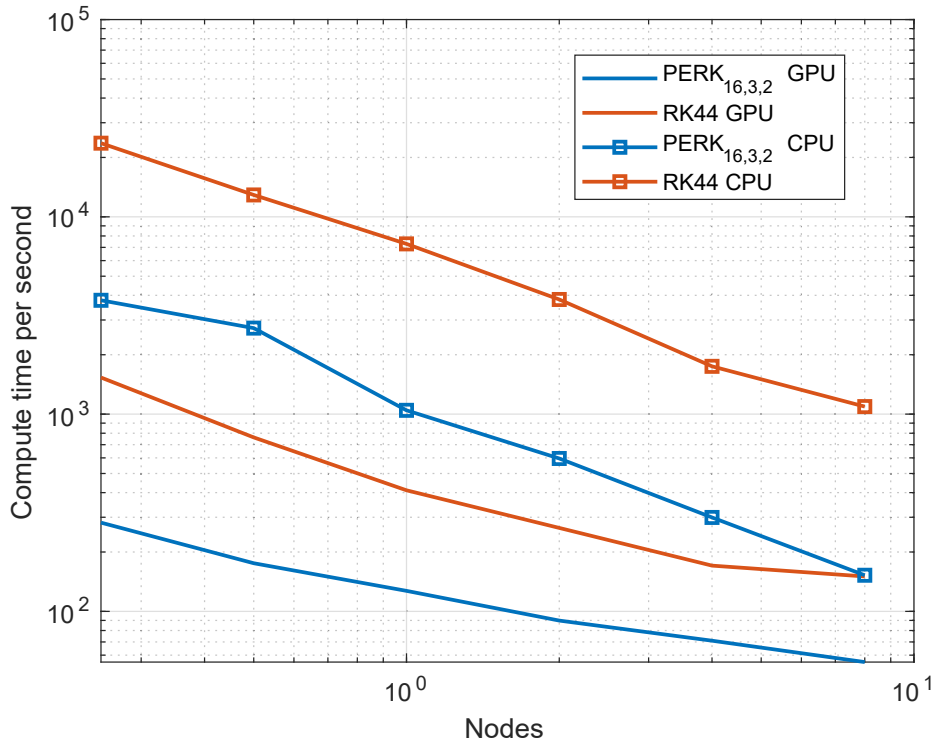


Figure 6.4: Clock/simulation time VS resources

From Figure 6.4, there are three important things to note. First, for all configurations observed, the P-ERK scheme runs faster. Second, runtimes on GPUs are orders of magnitude smaller than on CPUs. It shows that when comparing one entire CPU to a GPU, the gains are on average of 45 times and when comparing nodes a performance gain of average speedup of $\sim$36 is observed. Third, the gap between the two cases shrinks as the resources increase. It is important to note, however, that while the difference decreases it appears that the computational time is asymptotic. This would indicate that while they may approach similar values, the P-ERK method will remain faster. This can be attributed in part to the increase in computational overhead required to maintain communication between the various GPUs being used and the minimum time to perform the operations. This also means that infinite gains are not possible with simple parallelization and eventually a limit will be reached where the computation time will actually increase instead of decrease. In addition, Table

6.4 shows that for any given case it would take roughly eight times more resources using RK44 to achieve a similar performance when using a P-ERK.

	GPUs		CPUs	
Nodes	PERK	RK44	PERK	RK44
0.25	281.80	1533.18	3775.52	23623.99
0.5	175.43	761.88	2728.44	12912.14
1	127.07	410.98	1045.27	7292.41
2	89.83	264.91	595.78	3811.51
4	71.03	170.61	299.38	1746.66
8	55.35	150.35	152.48	1091.96

Table 6.4: P-ERK$_{16,3,2}$ and RK44 computer/simulated time

It is important to keep in mind, however, that for any parallel computing application, as the resources increase the computational overhead increases. This will result in diminishing returns and an eventual overall slowdown of the computation [100].

6.4 Numerical Results

Numerical results were collected for 1000 convective times with the flow becoming steady at roughly 150-200 convective times (t_c). The solution was sampled every $0.05t_c$ for P^o from 1 to 4, and the time averaged statistics were assessed at every $0.1t_c$, starting at $100t_c$. The instantaneous properties obtained include density, cartesian velocity components, and energy. The time-averaged quantities include pressure, streamwise and radial velocity components, and Reynolds stresses. Figure 6.5 shows that taking the time-average over a period of $350t_c$ after having achieved steady flow results in data that is reasonably time-independent to represent first- and second-order statistics. It can also be seen that for the time-averaged velocity, the second- and third-degree values converge nicely. The values for this figure were sampled at $x/D = 3$ and $r/D = 0$, which is a point downstream of the recirculation bubble.

6.4.1 Power Spectra

Power spectra were computed to determine the dominant frequencies present in the wake of the sphere. Various probes were used at axysymetric locations, as outlined in Table 6.5. As the

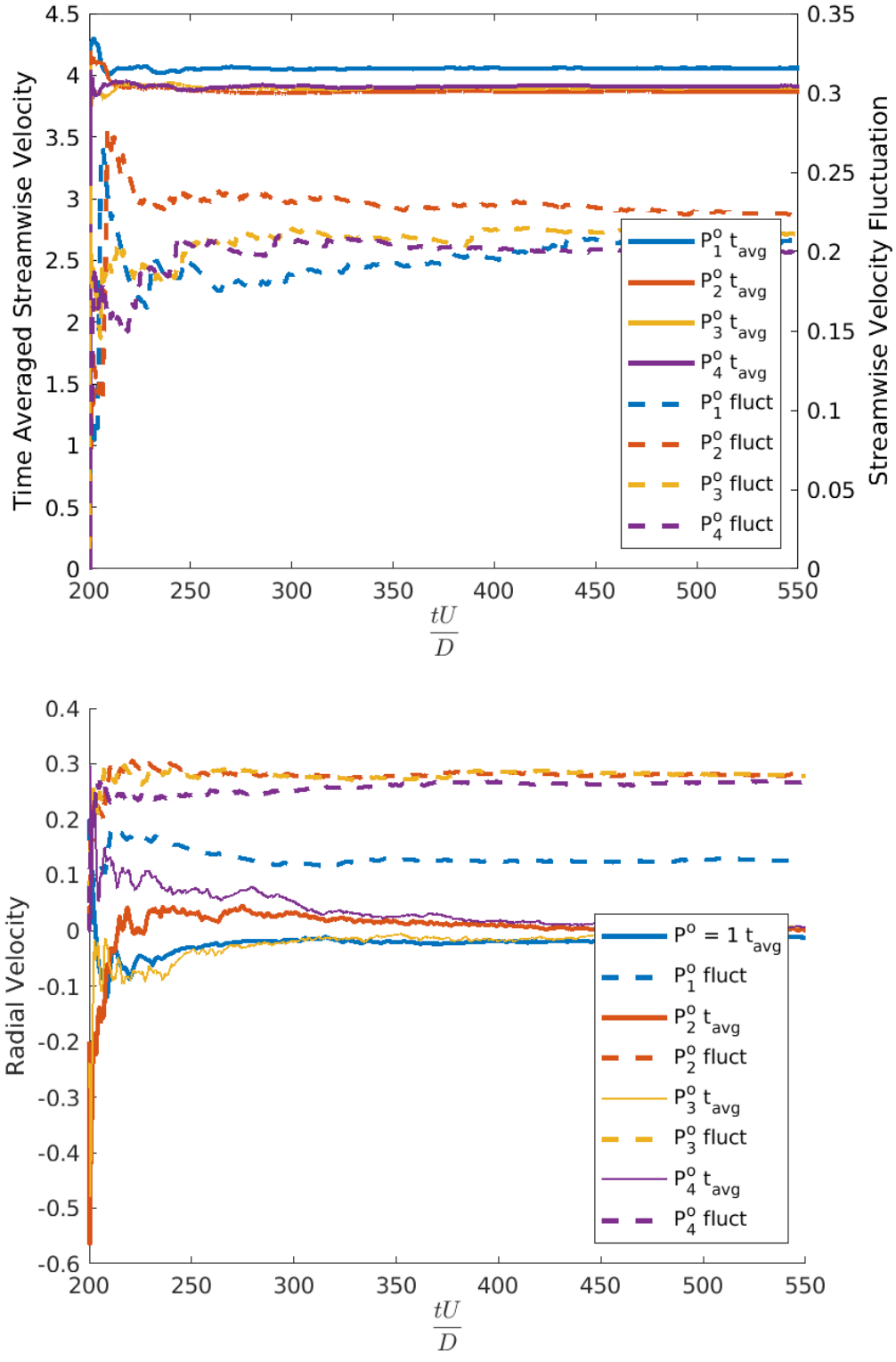


Figure 6.5: Streamwise and radial: velocity, time average, and fluctuation

table shows, for every streamwise station four probes are used at evenly spaced radial locations. The notable exception is the station at $x/D = 3$ which has an additional point located at $r/D = 0$ which is the one used in Figure 6.5. A sampling rate of $(t_c/2000)^{-1}$ was used to accurately capture both low-energy, high-frequency behaviours and high-energy low-frequency behaviours. The time series seen in Figure 6.6 were computed by averaging the velocity at each radial station

index	x/D	r/D	θ
1,2,3,4	1	0.6	0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$
5.6.7.8	2.2	0.6	0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$
9,10,11,12	2.4	0.6	0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$
13	3	0	0
14,15,16,17	3	0.6	0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$
18,19,20,21	5	0.6	0, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$

Table 6.5: Probe Locations

for a given streamwise direction. The power spectra were obtained by computing the power spectra for each radial station and averaging them, similarly to what was done for the time series. From the time series and power spectra shown in Figure 6.6, two observations are made: the dominant vortex shedding frequency in the wake, and the higher frequency disturbances in the boundary layer separation area. Firstly, from the peaks of the PSD far from the boundary layer, it can be observed that a large-scale vortex shedding frequency corresponding to the Strouhal number S_t of 0.219 is present for all solution polynomial degrees simulated. This peak is more clearly visible for the solution polynomial degree $P^o = 2$ line as it captures fewer of the low-energy eddies that can appear as noise in the PSD plot. This value correlates strongly with both numerical and experimental reference data as shown in Table 6.6. While this peak is present at all stations, it can be hard to notice due to the presence of noise and due to its decreasing magnitude as the distance from the body increases. Additionally, for all stations except the first, the energy decay with frequency increase corresponds with the Kolmogorov -5/3 law. Additionally, near the boundary layer, there is a higher frequency behaviour due to the shear layer separation that corresponds with the Kelvin-Helmholtz instabilities of $f_{kh} = 0.734$. In addition, the first station located at $x/D = 1$ appears to be located outside of the wake caused by the body, due to it only having very small fluctuations of radial velocity.

The figure also shows that as the polynomial degree increases, more of the frequencies are

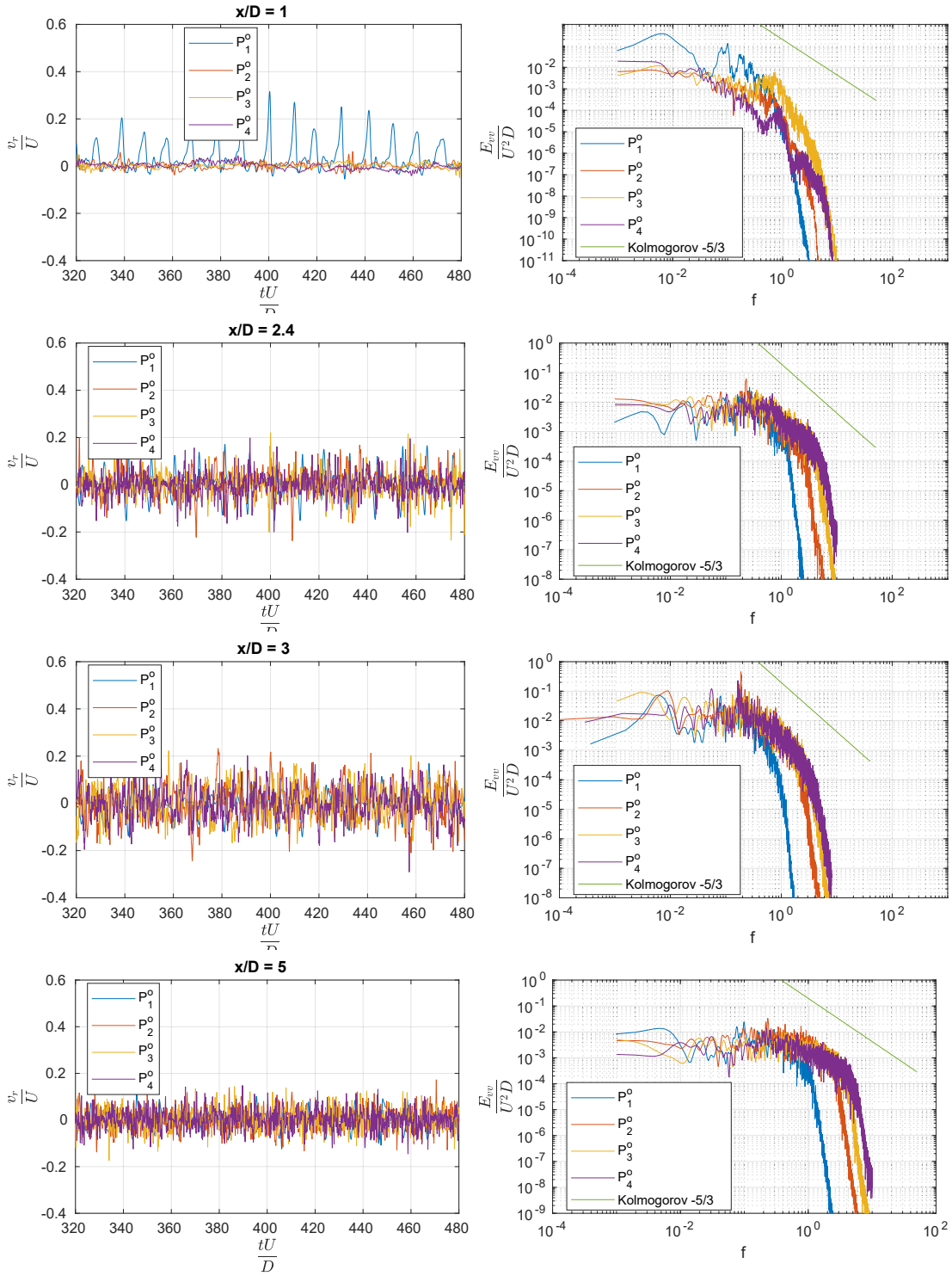


Figure 6.6: Average time series and power spectra for x/D = 1, x/D = 2.4, x/D = 3 and x/D = 5

captured. Figure 6.7 shows that for the streamwise velocity fluctuation at the center line there is also a peak at $f = 0.0143$ which can also be observed as a secondary peak for the probe located in the wake included in Figure 6.6. For this case, the polynomial degree has little effect on the captured frequencies. However, the magnitude of the fluctuations is more clearly captured.

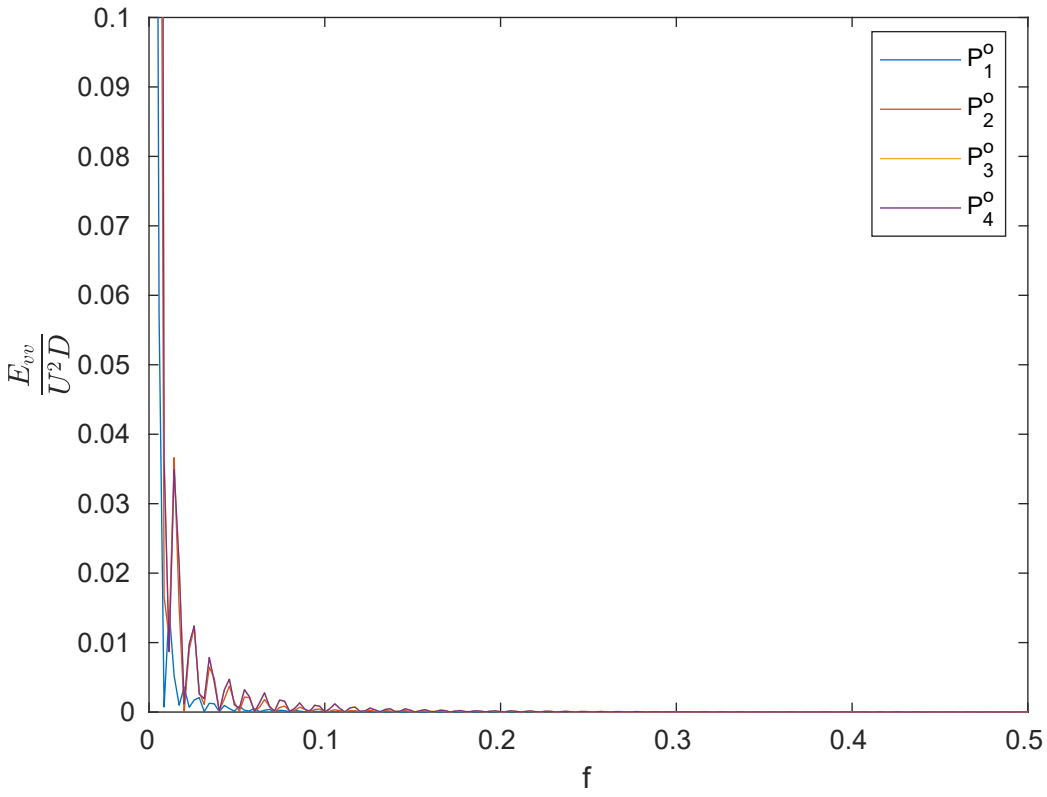


Figure 6.7: Power spectral density of the streamwise velocity fluctuation at $x/D = 3$ and $r/d = 0$

6.4.2 Coherent Structures

A generally accepted method to visualize and understand vortex formation and shedding is using the Q-criterion proposed by Hunt et al [101] as a method of identifying eddies. While other similar methods exist such as that proposed by Jeong and Hussain [102], for the purposes of this paper the Q-criterion method will be used. This value shows the relationship between vorticity and strain. For the Q criterion to be positive, the vorticity must be stronger than the strain. It can be defined mathematically as

$$Q = \frac{1}{2}(||\Omega||^2 - ||S||^2), \tag{89}$$

where $||\Omega||^2 = Tr(\Omega\Omega^t)$ represents the vorticity tensor and $||S||^2 = Tr(SS^t)$ is the strain rate tensor. Figure 6.8 shows how the wake develops behind the body of the sphere. It can be seen that the flow separates at an angle $\Phi_s = 89^o - 90^o$. However, the separated shear layer remains cohesive for some time before it becomes unstable and breaks down, joining the turbulent wake. This occurs at a similar location for P_2^o, P_3^o,and P_4^o. As expected, as the solution degree increases more of the small scale behaviours in the wake are being captured. While at P_1^o a few large shedding events are observed in the wake with little other significant behaviour, a dominant spiral pattern with smaller eddies also present emerges at higher orders. This spiral pattern is most noticeable for P_2^o as fewer small scale behaviours are present, but it is noticeable for P_3^o, and P_4^o as well.

The data in Figure 6.9 shows where the disturbances occur. While the time of the disturbances changes for any given position, its magnitude is relatively unchanged for a set x/D station. When comparing the stations within the shear layer to those within the transition area, they get amplified and propagate. This causes the flow at the further stations to be much more chaotic, indicating that the transition to turbulent flow occurs between these two stations for the bulk of the flow. In addition, as the disturbances don't occur simultaneously at each θ, this would contribute to the helical behaviour observed in Figure 6.8. This behaviour has been documented in Rodriguez et al [98], in Achenbach [103], in Contantinescu and Squires [96] and in Yun et al. [97].

6.4.3 Time Averaged Statistics

	Re	St	$\Phi_s(^o)$	$\bar{C_d}$	$\bar{C_{pb}}$	L/D
Current study P-ERK$_{16,e,2}$ LES P$^o_4$	3700	0.219	89-90	0.394	-0.205	2.16
Rodriguez et all. IMEX DNS [98]	3700	0.215	89.4	0.394	-0.207	2.28
Kim and Durbin Exp [91]	3700	0.225			-0.224	
Sakamoto and Hanui Exp [92]	3700	0.204				
Yun et al. IMEX LES [97]	3700	0.21	90	0.355	-0.194	2.622
Schlichting Exp [104]	4000			0.39		
Tomboulides DNS [95]	1000	0.195	102			1.7
Seidl et al. DNS [94]	5000		89.5	0.38		2.1
Constatninescu and Squires LES [96]	10000	0.195	84-86	0.393		1.7

Table 6.6: Summary of significant quantities

There are a number of values that are compared with reference data to assess the validity of the

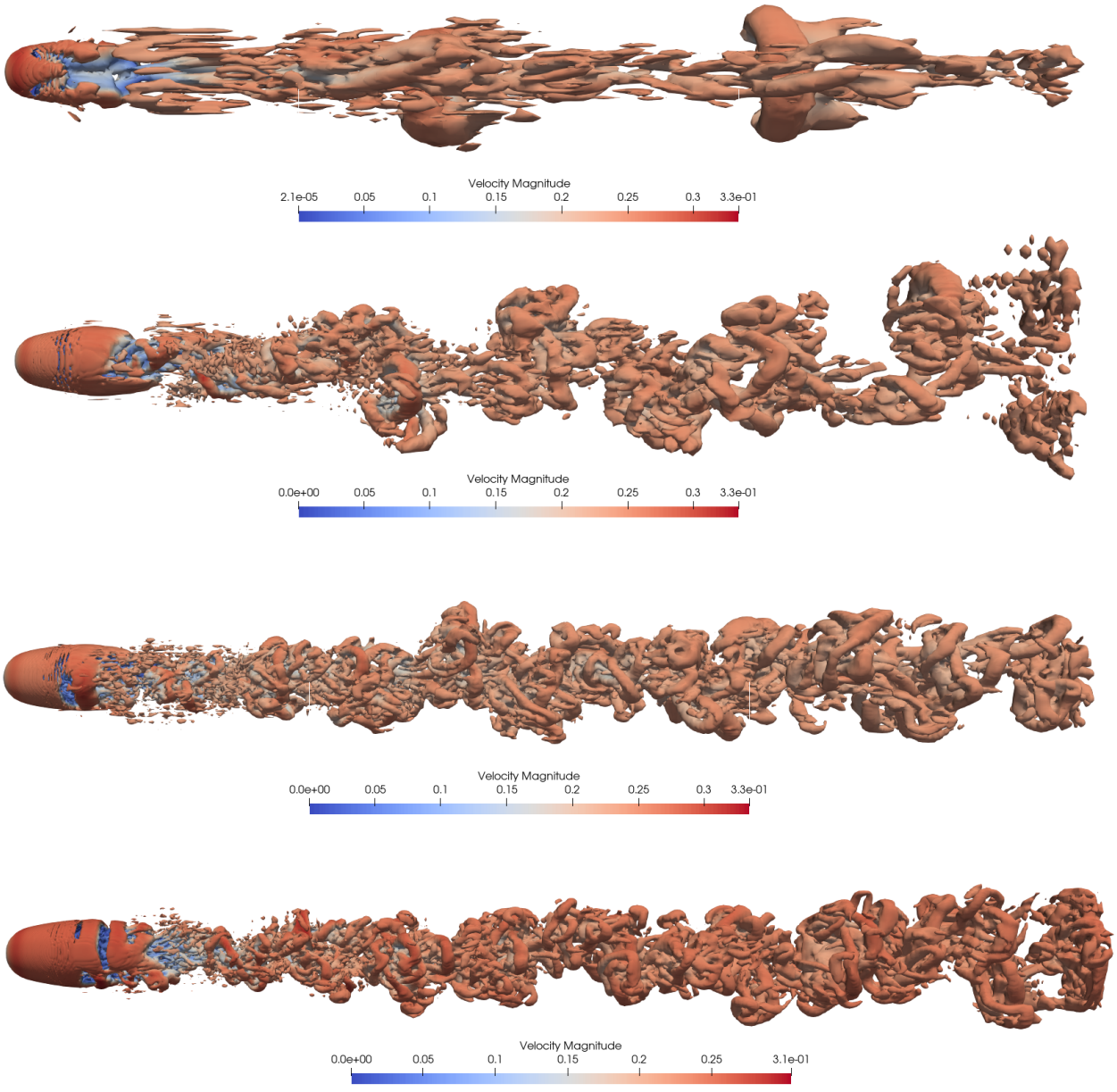
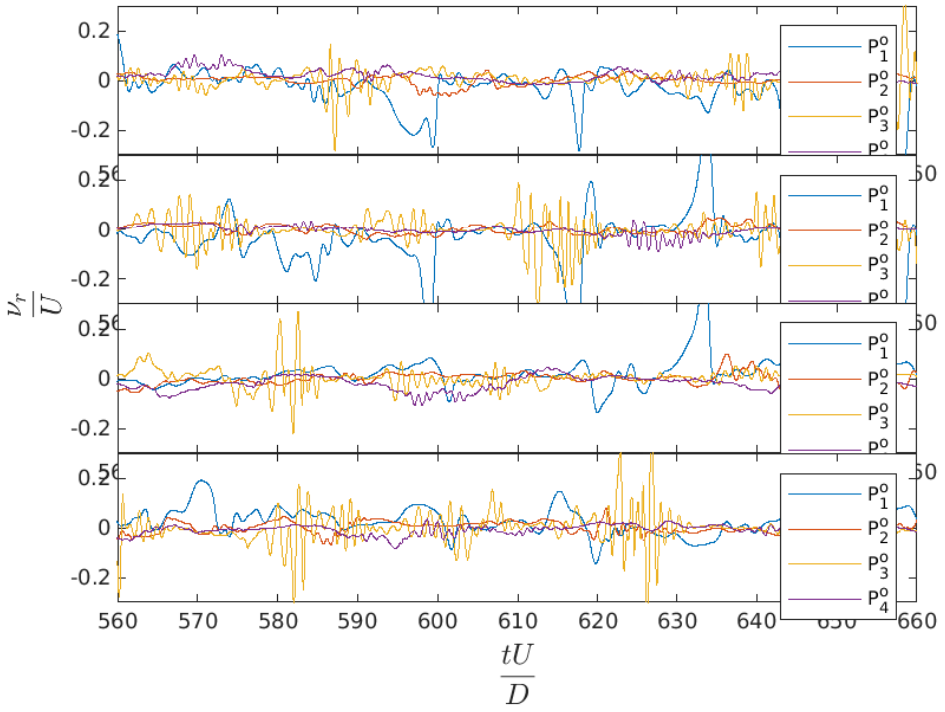


Figure 6.8: Q criterion contours for P^o_1, P^o_2, P^o_3, and P^o_4



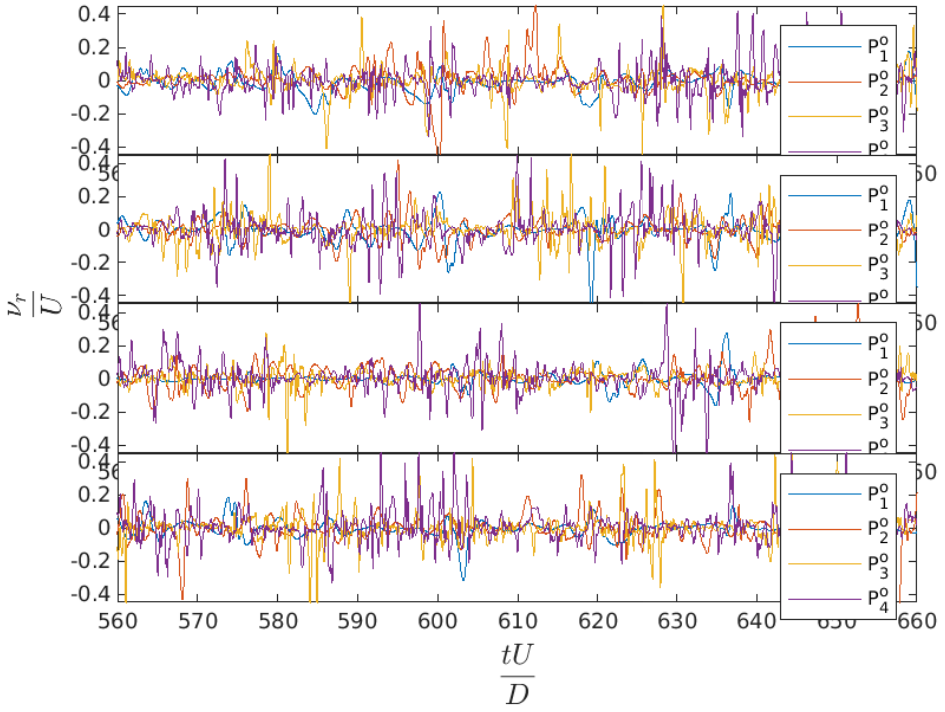


Figure 6.9: Time series of radial velocity at (a) $x/D = 1$ and (b) $x/D = 2.2$

methods used: the Reynolds number, Strouhal number, separation angle, average drag coefficient, back pressure ratio, and length of the separation bubble. These are all compiled in Table 6.6. The Strouhal number was obtained from the PSD plots in Figure 6.6 above, the separation angle was obtained by analyzing the Q-criterion contours in Figure 6.8, the drag coefficient was obtained directly from the forces, the back pressure ratio was obtained by evaluating the pressure coefficient (Cp) as in Figure 6.10, and the length of the separation bubble was obtained by looking at the time averaged velocity along the centerline as seen in Figure 6.11. All of these values were computed starting from a time after steady flow was obtained.

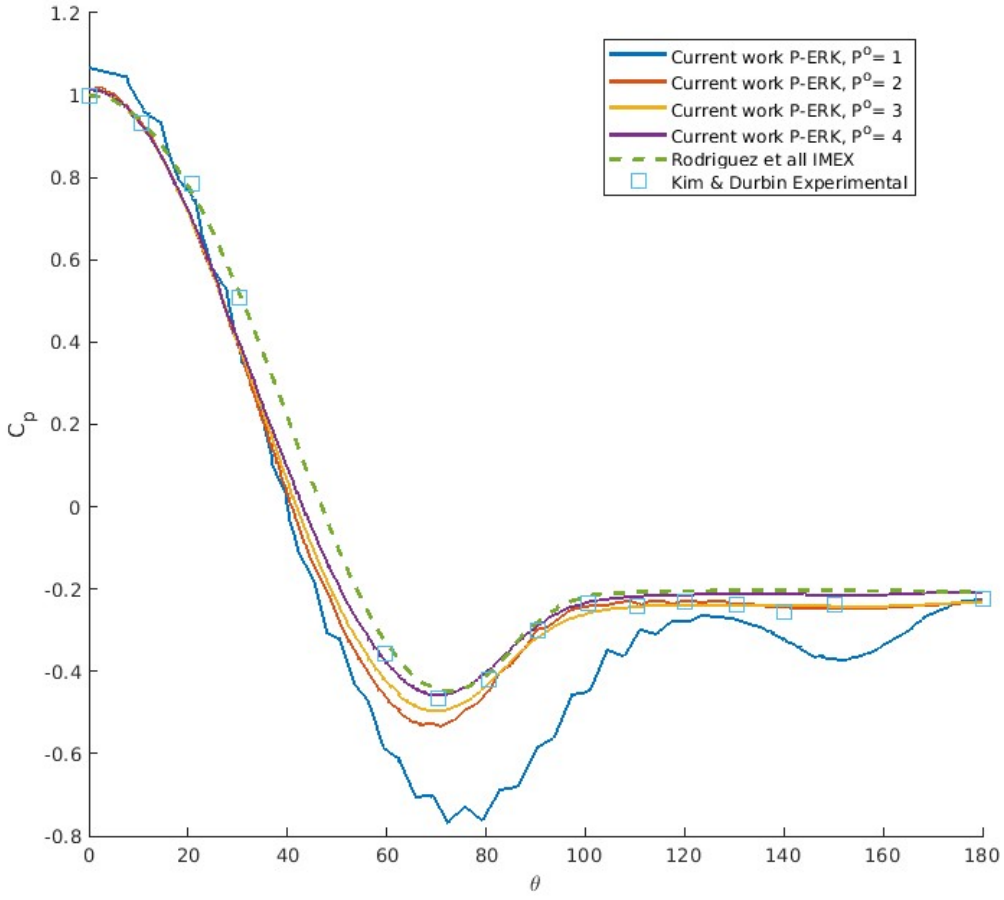


Figure 6.10: Cp VS angle of the sphere comparing current study to previous studies

Figure 6.10, illustrates the Cp distribution over the surface of the sphere. Qualitatively it correlates strongly with reference data for all points as the shape of the distribution follows closely both the DNS data obtained from Rodriguez et al [98] and the experimental data from Kim and Durbin [91]. Additionally, when looking at the back pressure ratio P_b it can be seen that it has a value of -0.209 which corresponds with reference data as outlined in Table 6.6.

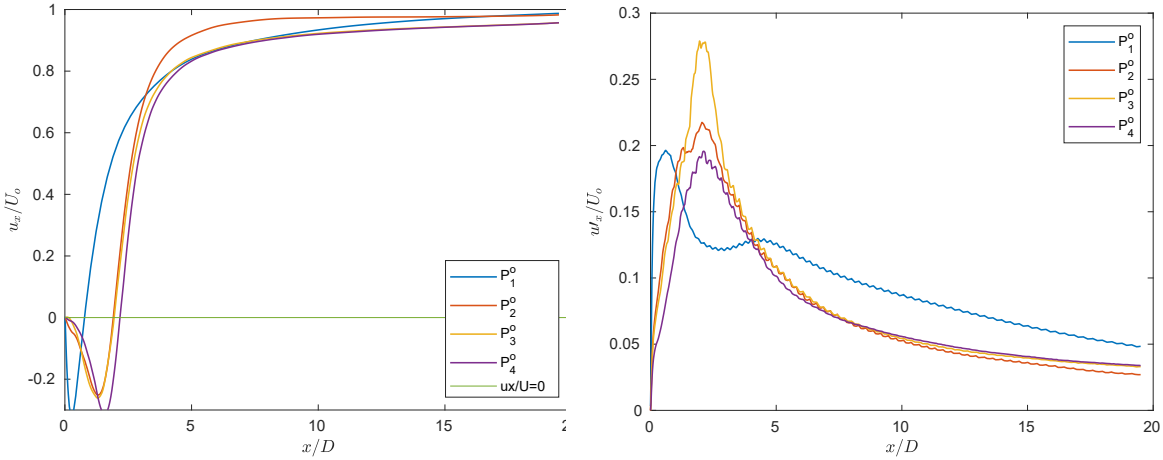


Figure 6.11: Time-averaged velocity(a) and root mean square velocity fluctuation(b) along the centerline of the wake

The length of the recirculation bubble was obtained by evaluating the distance between the rear surface of the sphere and the point where velocity becomes greater than 0, as seen in Figure 6.11 (a), sizes of L/D = 0.75, 1.90, 1.95, and 2.17. This can be further confirmed when looking at Figure 6.11 (b) as the mean fluctuation for each order has two peaks one at x/D = 1.95 for all orders and at x/D = 2.15 for P_2^o, and 2.13 for P_3^o and P_4^o. The peak with the greatest amplitude therefore corresponds with the end of the recirculation bubble. While there is some discrepancy in the size of the recirculation bubble when looking at the streamwise velocity distribution at various points along the wake such as in Figure 6.12, it can be seen that the same differences are present. The discrepancy is largest at x/D = 3 which is part of the recovery zone. In addition, this figure shows clearly that for polynomial degrees 2 and 3 there is a strong level of convergence in the observed values. It can be said that the results obtained from this experiment correlate with both DNS and experimental results at polynomial degrees 2 and greater. In addition, the correlation is stronger with increasing P^o with P_4^o closely following the DNS results obtained by Rodriguez et al [98] .

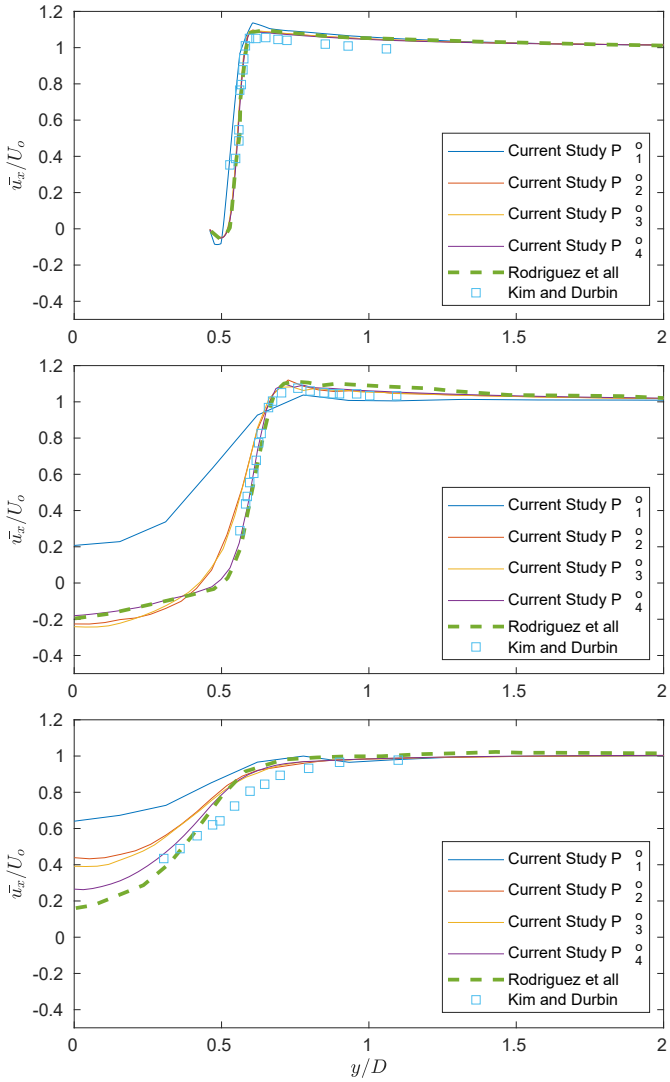


Figure 6.12: mean streamwise velocity at various points. x/D = 0.2, x/D=1.6 and, x/D=3

Figure 6.13, shows the mean velocity profiles at distances of 1.6, 2, 3, 5, and 10 x/D from the origin and from 0 to 2 y/D in the z = 0 plane. Negative streamwise velocities are observed for the x/D = 1.6 and 2 cases as near the centerline these are still within the recirculation region of the wake. This also partially explains to why the initial values for the velocity are small at x/D = 3, as that is relatively close to the end of the bubble. Figure 6.14 shows the mean fluctuation for the same region. It shows that for X/D = 1.6, 2, 3, and 5, there is a peak in the streamwise fluctuation at approximately y/D = 0.5 which decreases as y/D increases further. This can be understood by studying the Q-criterion presented in Figure 6.8, as that is the transition area between the surrounding flow regime

and the wake of the sphere. The magnitude of the fluctuations tends to decrease as the streamwise distance increases. The radial fluctuation presents a similar behaviour to the streamwise component for x/D = 1.6 and 2, but afterwards there is no noticeable peak and it adopts a general downwards slope.

Figures 6.15, 6.16, 6.17, and 6.18 show contours of the Reynolds streamwise normal stress, the cross-stream normal stress, and the shear stress for increasing polynomial degrees. While they do not perfectly match those presented in Rodriguez et al [98] and those presented in Jang and Lee [105], there is an increasing degree of agreement with increasing polynomial degree. It is important to note that when comparing to the vizualisation presented in Jang and Lee at Re=11,000 the recirculation bubble in the present work is larger and can partially account for the discrepancies. Of the values contoured in Figures 6.15, 6.16, 6.17, and 6.18 the Reynolds shear stresses show the strongest agreement.

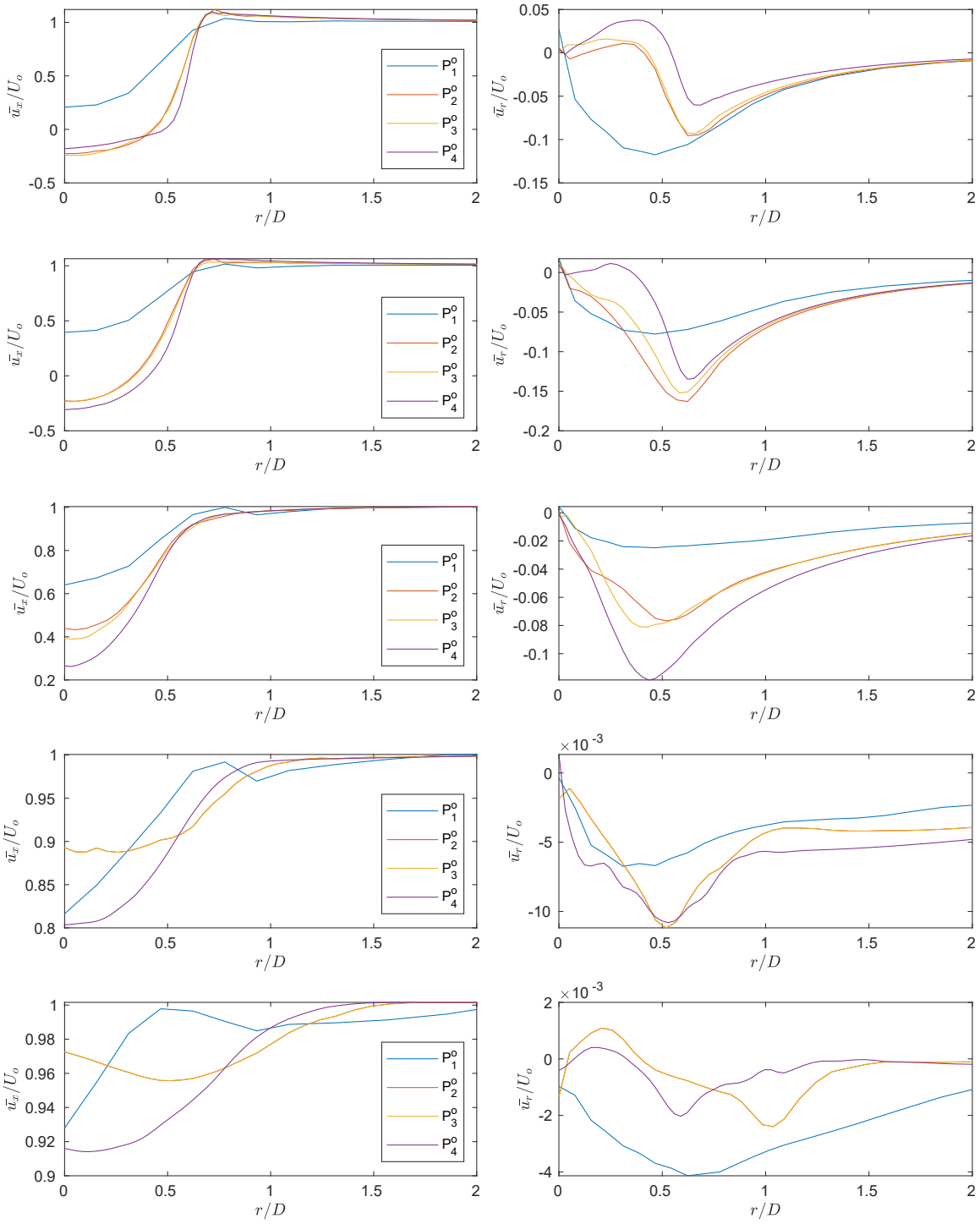


Figure 6.13: mean velocity profiles at various points in the wake. x/D, y/D, and z= 0

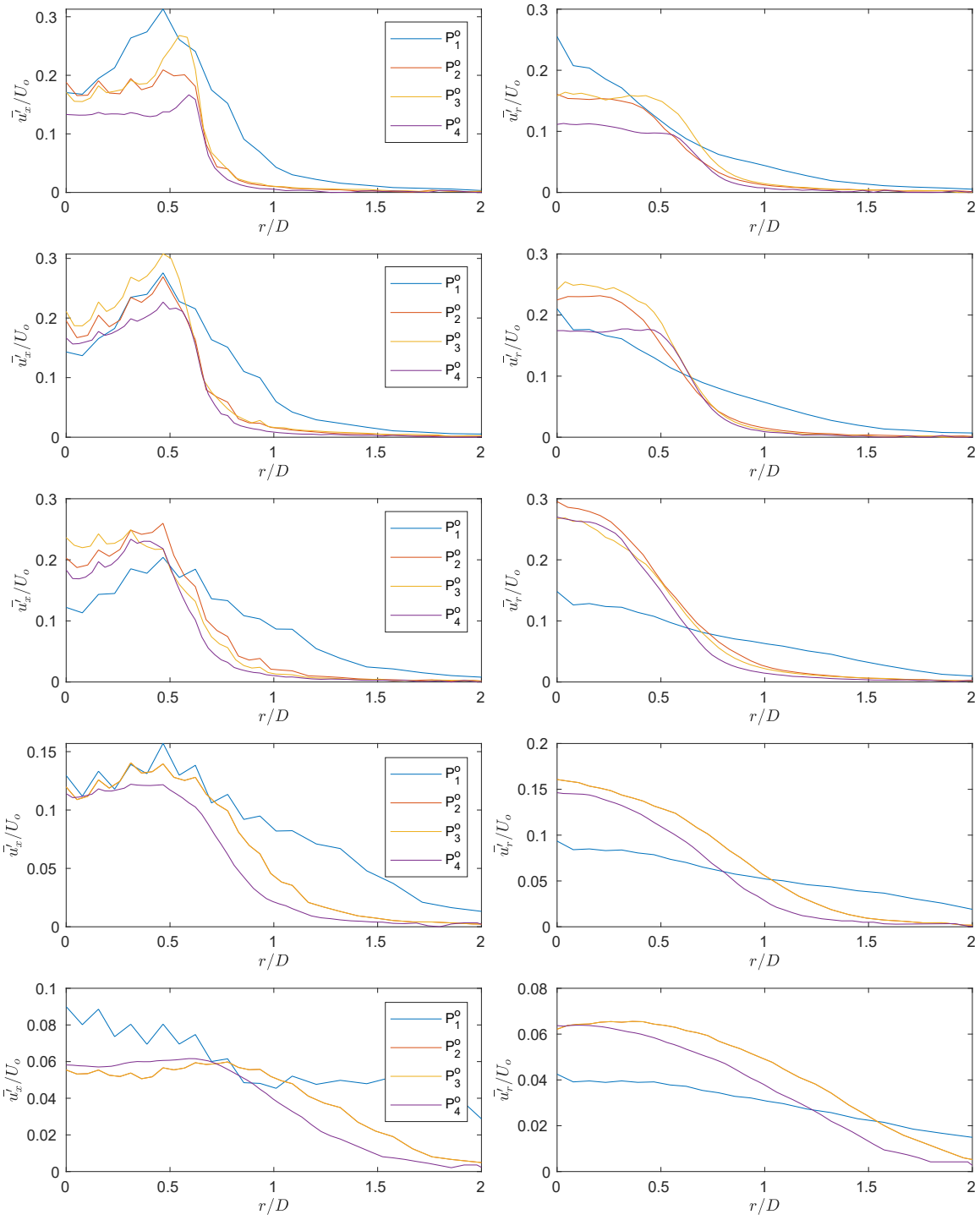


Figure 6.14: Root mean square velocity fluctuations at various points in the wake. x/D, y/D, and z= 0

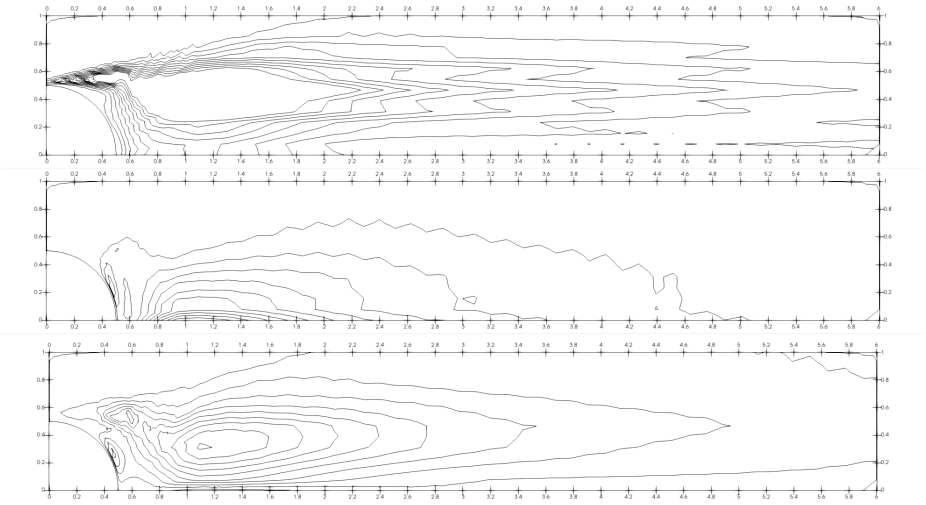


Figure 6.15: Reynolds streamwise normal stress, cross-stream normal stress, and shear stress contours for $P^o = 1$

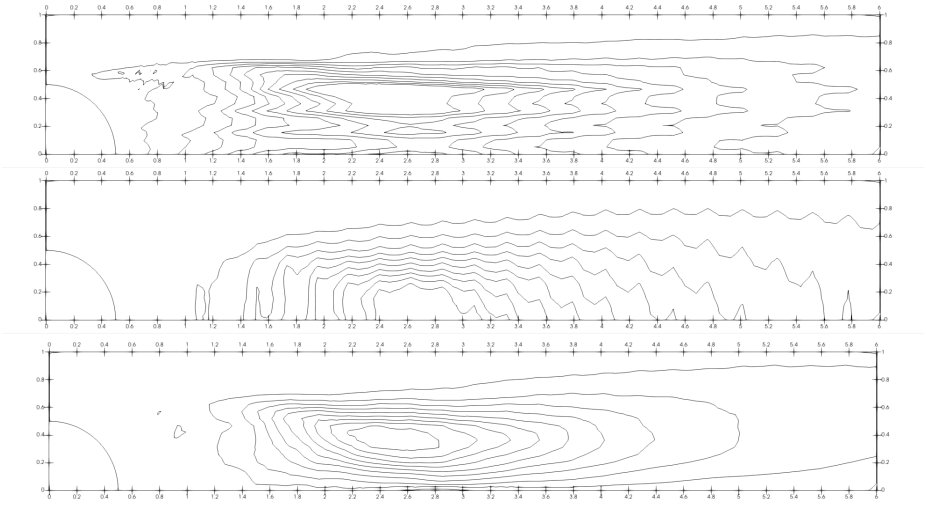


Figure 6.16: Reynolds streamwise normal stress, cross-stream normal stress, and shear stress contours for $P^o = 2$

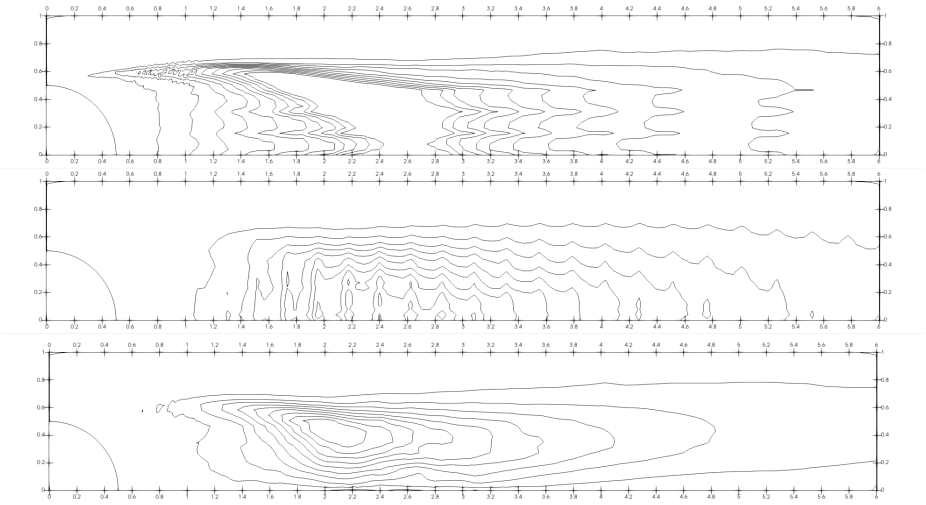


Figure 6.17: Reynolds streamwise normal stress, cross-stream normal stress, and shear stress contours for $P^o = 3$

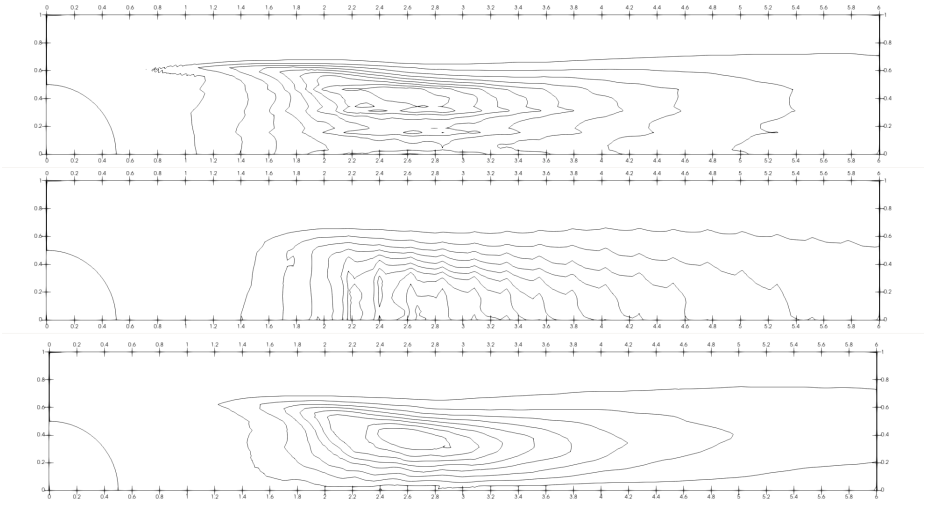


Figure 6.18: Reynolds streamwise normal stress, cross-stream normal stress, and shear stress contours for $P^o = 4$

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

This study provides supporting evidence for the the applicability and performance of the Paired Explicit Runge-Kutta time-stepping schemes using flux reconstruction spatial discretization and implict large eddy simulation turbulence modelling approach. A well documented case of sub-critical flow over a sphere was used as a test case, drawing from both numeric and experimental data as a reference.

This study has shown that for the given case, strong agreements can be observed between the P-ERK results and reference results. This agreement improves as the polynomial degree enters higher orders. It has also shown that for the given case on a single GPU with scaling polynomial degree the P-ERK method performs with a speedup factor of $\sim$4.7 compared to the traditional fourth order RK method, which is in line with previous findings of similar speedup factors for third-order schemes. In addition, with parallel computing using multiple GPUs, it was shown that the P-ERK method achieves superior performance to the traditional fourth order RK method with an average speedup factor of $\sim$4. This is especially significant when considering the order of improvements shown by other methods when compared to the RK method such as certain IMEX schemes showing a speedup factor of only up to 1.97. Moreover if the P-ERK method is combined with GPU acceleration when compared to RK44 on CPUs, a speedup of $\sim$45 is achieved when compared one to one and an average speedup of $\sim$36 when compared node to node is observed.

7.2 Recommendations

Considering the significant performance improvements that have been observed in this study, it becomes clear that there is merit in further exploring this novel time stepping method. It would be especially interesting to study how it performs under different numeric and experimental conditions and with expansion into still higher orders. The case studied in the current paper presents benchmark flow conditions that have been previously successfully analyzed using a variety of methods. It would be pertinent to explore how this method behaves and how its performance may be affected by more challenging flow conditions such as within a turbine cascade, a shock tube and, many others. While it has been shown that for third- and fourth-order accuracy, the P-ERK method shows a speedup factor of $\sim$4-5, it would be interesting to see how this value may change at higher orders.

Bibliography

[1] Air Transport Action Group. Waypoint 2050, 2021. https://aviationbenefits.org/environmental-efficiency/climate-action/waypoint-2050/.

[2] S. Hedayati Nasab. *Accelerated Temporal Schemes for High-Order Unstructured Methods*. PhD thesis, Concordia University, 2021. Unpublished.

[3] Stanford VLSI Group. Technology scaling. *CPU DB - Looking At 40 Years of Processor Improvements — A complete database of processors for researchers and hobbyists alike.* http://cpudb.stanford.edu/visualize/technology_scaling.

[4] G. Petrone. Unleashing the power of multiple gpus for cfd simulations. *ANSYS BLOG*, 2022. https://www.ansys.com/blog/unleashing-the-power-of-multiple-gpus-for-cfd-simulations#:~:text=The%20main%20benefits%20of%20using,as%20more%20than%202000%20CPUs!

[5] C. Pereira. Analysis of high-order element types for implicit large eddy simulation. Master's thesis, Concordia University, 1455 Boul. de Maisonneuve Ouest, Montréal, QC H3G 1M8, 2019.

[6] M.C. Sheehan, M.A. Fox, C. Kaye, and B. Resnick. Integrating health into local climate response: Lessons from the U.S. CDC climate-ready states and cities initiative. *Environmental Health Perspectives*, 125(9):094501, 2017.

[7] Kyoto protocol to the united nations framework convention on climate change, 1997.

[8] The paris agreement, 2015.

[9] B. Curry and S. Mccarthy. Canada formally abandons kyoto protocol on climate change. *The Globe and Mail*, 2011.

[10] AIRBUS. Imagine travelling in this blended wing body aircraft, 2020. https://www.airbus.com/en/newsroom/stories/2020-11-imagine-travelling-in-this-blended-wing-body-aircraft.

[11] S.A. Orszag. Analytical theories of turbulence. *Journal of Fluid Mechanics*, 41(2):363–386, 1970.

[12] R. S. Rogallo. *Numerical experiments in homogeneous turbulence*. National Aeronautics and Space Administration, 1981.

[13] R.D. Moser and P. Moin. The effects of curvature in wall-bounded turbulent flows. *Journal of Fluid Mechanics*, 175:479–510, 1987.

[14] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.

[15] W.J. Feiereisen, W. C. Reynolds, and J. H. Ferziger. *Numerical simulation of a compressible homogeneous, turbulent shear flow*. PhD thesis, Stanford University, California, Stanford University, California, Stanford University, California, 1981.

[16] G. Erlebacher, M. Hussaini, S. Sarkar, and H. Kreiss. The analysis and simulation of compressible turbulence. *Theoret. Computational Fluid Dyn.*, 2, 1990.

[17] S. Sarkar, G. Erlebacher, and M. Hussaini. Direct simulation of compressible turbulence in a shear flow. *Theoretical and Computational Fluid Dynamics*, 2, 1991.

[18] G.A. Blaisdell, N.N. Mansour, and W.C. Reynolds. Compressibility effects on the growth and structure of homogeneous turbulent shear flow. *Journal of Fluid Mechanics*, 256:443–485, 1993.

[19] A.W. Vreman, N. Sandham, and K. Luo. Compressible mixed layer growth rate and turbulence characteristics. *Journal of Fluid Mechanics*, 320:235 – 258, 1996.

[20] S. Lee, S.K. Lele, and P. Moin. Direct numerical simulation of isotropic turbulence interacting with a weak shock wave. *Journal of Fluid Mechanics*, 251:533–562, 1993.

[21] P. Moin and K. Mahesh. Direct numerical simulation: A tool in turbulence research. *Annu. Rev. Fluid Mech.*, 30:539–578, 1998.

[22] O. Reynolds. Iv. on the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:123–164, 1895.

[23] J. Boussinesq. *Essai sur la théorie des eaux courantes*. Imprimerie Nationale, 1877.

[24] L. Prandtl. 7. bericht über untersuchungen zur ausgebildeten turbulenz. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 5(2):136–139, 1925.

[25] L. Prandtl. Über ein neues formelsystem für die ausgebildete turblenz. *Nachr. Akad. Wiss. Gottingen, Math. Phys.*, 1, 1945.

[26] J.C. Rotta. Statistische theorie nichthomogener turbulenz. *Zeitschrift für Physik*, 129:547–572, 1951.

[27] D. Naot, A. Shavit, and M. Wolfshtein. Interaction between components of the turbulent-velocity correlation tensor. *Israel Journal of Technology*, 8, 1970.

[28] W.P Jones and B.E Launder. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 15(2):301–314, 1972.

[29] K. Hanjalic and B. Launder. A reynolds stress model of turbulence and its application to thin shear flows. *Journal of Fluid Mechanics*, 52:609 – 638, 1972.

[30] G. Launder, B.and Reece and W. Rodi. Progress in the development of a reynolds stress turbulence closure. *Journal of Fluid Mechanics*, 68:537 – 566, 1975.

[31] U. Schumann. Realizability of reynolds-stress turbulence models. *The Physics of Fluids*, 20(5):721–725, 1977.

[32] J.L. Lumley. Computational modeling of turbulent flows*1this work supported in part by the u.s. national science foundation, meteorology program under grant number atm77-22903, and in part by the u. s. office of naval research, fluid dynamics branch. it is a pleasure to acknowledge fruitful discussion with b.brumley, and the computational help of d. hatziavromidis. volume 18 of *Advances in Applied Mechanics*, pages 123–176. Elsevier, 1979.

[33] F.R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.

[34] D.C. Wilcox. Formulation of the k-w turbulence model revisited. *AIAA Journal*, 46(11):2823–2838, 2008.

[35] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99 – 164, 1963.

[36] J.W. Deardorff. Convective velocity and temperature scales for the unstable planetary boundary layer and for rayleigh convection. *Journal of Atmospheric Sciences*, 27(8):1211 – 1213, 1970.

[37] U. Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 18(4):376–404, 1975.

[38] K. Luo, J. Yang, Y. Bai, and J. Fan. Large eddy simulation of turbulent combustion by a dynamic second-order moment closure model. *Fuel*, 187, 2017.

[39] M. Breuer and W. Rodi. Flow around a surface mounted cubical obstacle: Comparison of les and rans-results. *Notes of Numerical Fluid Mechanics*, 53:22–30, 1996.

[40] A. Leonard. Energy cascade in large eddy simulations of turbulent fluid flow. *Adv. Geophys.*, 18A:237–248, 1974.

[41] J. Bardino, J.H. Ferziger, and W.C. Reynolds. Improved turbulence models based on large eddy simulation of homogeneous, incompressible, turbulent flows. 1983.

[42] C.G. Speziale. Subgrid scale stress models for the large-eddy simulation of rotating turbulent flows. *Geophysical & Astrophysical Fluid Dynamics*, 33(1-4):199–222, 1985.

[43] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[44] M. Germano. Turbulence: the filtering approach. *Journal of Fluid Mechanics*, 238:325–336, 1992.

[45] J.P. Boris. On large eddy simulation using subgrid turbulence models comment 1. In J.L. Lumley, editor, *Whither Turbulence? Turbulence at the Crossroads*, pages 344–353, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[46] B. Cockburn. Discontinuous galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.

[47] Z.J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation. *Journal of Computational Physics*, 178:210–251, 2002.

[48] Y. Liu and M. Vinokur. Discontinuous spectral difference method for conservation laws on unstructured grids. 2004.

[49] H.T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. *AIAA Paper AIAA 20074079*, pages 1–42, 2007.

[50] J. Blazek. Chapter 3 - principles of solution of the governing equations. In J. Blazek, editor, *Computational Fluid Dynamics: Principles and Applications (Third Edition)*, pages 29–72. Butterworth-Heinemann, Oxford, third edition edition, 2015.

[51] M.H. Zawawi, A. Saleha, A. Salwa, N.H. Hassan, N.M. Zahari, M.Z. Ramli, and Z.C. Muda. A review: Fundamentals of computational fluid dynamics (cfd). *AIP Conference Proceedings*, 2030(1):020252, 2018.

[52] J. Ekaterinaris. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41:192–300, 2005.

[53] P.W. McDonald. The computation of transonic flow through two-dimensional gas turbine cascades. volume ASME 1971 International Gas Turbine Conference and Products Show of *Turbo Expo: Power for Land, Sea, and Air*, 1971. V001T01A089.

[54] Z.J. Wang, Y. Liu, C. Lacor, and J.L. Azevedo. *Spectral Volume and Spectral Difference Methods*. 2016.

[55] Y. Liu, M. Vinokur, and Z. J. Wang. Spectral difference method for unstructured grids i: Basic formulation. *Journal of Computational Physics*, 216:780–801, 2006.

[56] J. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, volume 54. 2007.

[57] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8. 1993.

[58] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[59] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 14. 1996.

[60] R.M. Beam and R.F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22(1):87–110, 1976.

[61] C.G.J. Jacobi. Ueber eine neue auflosungsart der bei der methode der kleinsten quadrate vorkommenden linearen gleichungen. *Astronomische Nachrichten*, 22(20):297–306, 1845.

[62] L. Seidel. Ueber ein verfahren, die gleichungen, auf welche die methode der kleinsten quadrate fuhrt, sowie lineare gleichungen uberhaupt, durch successive annaherung aufzulosen. In *Abhandlungen der Mathematisch-Physikalischen Klasse der Königlich Bayerischen Akademie der Wissenschaften*, volume Bd.11 (1874), pages 83–108. Königlich Bayerische Akademie der Wissenschaften. Mathematisch-Physikalische Klasse., 1874. https://www.biodiversitylibrary.org/bibliography/10920.

[63] Y. Saad and M.H. Schultz. Gmres: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[64] B.C. Vermeire and S. Nadarajah. *Adaptive IMEX Time-Stepping for the Correction Procedure via Reconstruction Scheme*.

[65] B.C. Vermeire and S. Nadarajah. Adaptive imex schemes for high-order unstructured methods. *Journal of Computational Physics*, 280:261–286, 2015.

[66] B.C. Vermeire. Paired explicit runge-kutta schemes for stiff systems of equations. *Journal of Computational Physics*, 393, 2019.

[67] S. Hedayati Nasab and B.C. Vermeire. Third-order paired explicit runge-kutta schemes for stiff systems of equations. *Journal of Computational Physics*, 468:111470, 2022.

[68] 3nm technology. *3nm Technology*.

[69] M. Quinn. *Parallel Computing: Theory and Practice*. 1994.

[70] K. Niemeyer and C.J. Sung. Recent progress and challenges in exploiting graphics processors in computational fluid dynamics. *The Journal of Supercomputing*, 67:528–564, 2013.

[71] F.D. Witherden. *On the Development and Implementation of High-Order Flux Reconstruction Schemes for Computational Fluid Dynamics*. PhD thesis, Department of Aeronautics, Imperial College London, London SW7 2AZ, United Kingdom, 2015.

[72] B.C. Vermeire, C. Pereira, and H.R. Karbasian. *Computational Fluid Dynamics: An Open-Source Approach*. 2021.

[73] C. Hirsch. *Numerical Computation of Internal and External Flows (Second Edition)*. Butterworth-Heinemann, Oxford, second edition edition, 2007.

[74] A.N. Kolmogorov. Dissipation of energy in the locally isotropic turbulence. *Proceedings: Mathematical and Physical Sciences*, 434(1890):15–17, 1991.

[75] P. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. 2015.

[76] Z.J. Wang and G. Haiyang. A unifying lifting collocation penalty formulation including the discontinuous galerkin, spectral volume/difference methods for conservation laws on mixed grids. *Journal of Computational Physics*, 228(21):8161–8186, 2009.

[77] T. Haga, H. Gao, and Z.J. Wang. A high-order unifying discontinuous formulation for the navier-stokes equations on 3d mixed grids. *Mathematical Modelling of Natural Phenomena*, 6, 2011.

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

[79] D. Williams and A. Jameson. Energy stable flux reconstruction schemes for advection-diffusion problems on tetrahedra. *Journal of Scientific Computing*, 59, 2014.

[80] J. Romero and A. Jameson. Extension of the flux reconstruction method to triangular elements using collapsed-edge quadrilaterals. 2016.

[81] B.C. Vermeire and P.E. Vincent. On the properties of energy stable flux reconstruction schemes for implicit large eddy simulation. *Journal of Computational Physics*, 327:368–388, 2016.

[82] B.C. Vermeire and P.E. Vincent. On the behaviour of fully-discrete flux reconstruction schemes. *Computer Methods in Applied Mechanics and Engineering*, 315:1053–1079, 2017.

[83] B.C. Vermeire, S. Nadarajah, and P.G. Tucker. Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme. *International Journal for Numerical Methods in Fluids*, 82(5), 2016.

[84] C.A. Pereira and B.C. Vermeire. Performance and accuracy of hybridized flux reconstruction schemes. *Journal of Computational Physics*, 457, 2022.

[85] V.V. Rusanov. On difference schemes of third order accuracy for nonlinear hyperbolic systems. *Journal of Computational Physics*, 5(3):507–516, 1970.

[86] Francesco Bassi and Stefano Rebay. A high order discontinuous galerkin method for compressible turbulent flows. In Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu, editors, *Discontinuous Galerkin Methods*, pages 77–88, Berlin, H., 2000. Springer Berlin Heidelberg.

[87] J. Butcher. *Numerical Methods for Ordinary Differential Equations*. 2016.

[88] R. Dorf and R. Bishop. *Modern Control Systems, 13th Edition*. 2017.

[89] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8. 1993.

[90] E. Achenbach. Experiments on the flow past spheres at very high reynolds numbers. *Journal of Fluid Mechanics*, 54(3):565–575, 1972.

[91] H. Kim and P. Durbin. Observation of the frequencies in a sphere wake and of drag increase by acoustic excitation. *Phys. Fluids*, 31, 1988.

[92] H. Sakamoto and H. Haniu. A study on vortex shedding from spheres in a uniform flow. *Journal of Fluids Engineering-transactions of The Asme - J FLUID ENG*, 112:386–392, 1990.

[93] M. Ozgoren, A. Okbaz, A. Kahraman, R. Hassanzadeh, B. Sahin, H. Akilli, and S. Dogan. Experimental investigation of the flow structure around a sphere and its control with jet flow via piv. 2011.

[94] V. Seidl, S. Muzaferija, and M. Peric. Parallel dns with local grid refinement. *Applied Scientific Research*, 59:379–394, 1997.

[95] A. Touboulides and S. Orszag. Numerical investigation of transitional and weak turbulent flow past a sphere. *Journal of Fluid Mechanics*, 416:45 – 73, 2000.

[96] G. Constantinescu and K. Squires. Numerical investigations of flow over a sphere in the subcritical and supercritical regimes. *Physics of Fluids*, 16, 2004.

[97] G. Yun, D. Kim, and H. Choi. Vortical structures behind a sphere at subcritical reynolds numbers. *Physics of Fluids - PHYS FLUIDS*, 18, 2006.

[98] I. Rodriguez, R. Borrell, O. Lehmkuhl, C.D. Perez-Segarra, and A. OLIVA. Direct numerical simulation of the flow over a sphere at re = 3700. *Journal of Fluid Mechanics*, 679:263 – 287, 2011.

[99] B.C. Vermeire and S. Hedayati Nasab. Accelerated implicit-explicit runge-kutta schemes for locally stiff systems. *Journal of Computational Physics*, 429:110022, 2020.

[100] R. Arpaci, A. Arpaci-Dusseau, A. Vahdat, L. Liu, T. Anderson, and D. Patterson. The interaction of parallel and sequential workloads on a network of workstations. *ACM SIGMETRICS Performance Evaluation Review*, 23:267–278, 1995.

[101] J. Hunt, A. Wray, and P. Moin. Eddies, streams, and convergence zones in turbulent flows. *Studying Turbulence Using Numerical Simulation Databases*, -1:193–208, 1988.

[102] J. Jeong and F. Hussain. Hussain, f.: On the identification of a vortex. jfm 285, 69-94. *Journal of Fluid Mechanics*, 285:69 – 94, 1995.

[103] E. Achenbach. Vortex shedding from spheres. *Journal of Fluid Mechanics*, 62:209 – 221, 1974.

[104] H. Schlichting and K. Gersten. *Boundary-layer theory*, volume 9. 2016.

[105] Y.I. Jang and S.J. Lee. Visualization of turbulent flow around a sphere at subcritical reynolds numbers. *Journal of Visualization*, 10:359–366, 2007.