

Space-Time Multiscale Methods for Large Eddy Simulation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 23 mei 2006 om 12.30 uur

door

Edwin Albert MUNTS

ingenieur luchtvaart en ruimtevaart
geboren te Dordrecht

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. R. de Borst

Samenstelling promotiecommissie:

Rector Magnificus,	Voorzitter
Prof. dr. ir. R. de Borst,	Technische Universiteit Delft, promotor
Prof. dr. ir. B. Koren,	Technische Universiteit Delft
Prof. dr. ir. B.J. Geurts,	Universiteit Twente
Prof. dr. D. Lohse,	Universiteit Twente
Prof. Dr.-Ing. W.A. Wall,	Technische Universität München
Dr. S.J. Hulshoff,	Technische Universiteit Delft
Dr. ir. M.I. Gerritsma,	Technische Universiteit Delft

© 2006 by E.A. Munts

Any part of this work may be copied and distributed freely, provided that the copied material refers to the original work and its author.

Printed in the Netherlands by Printpartners Ipskamp, Enschede.

ISBN-10: 90-9020704-X

ISBN-13: 978-90-9020704-9

Voorwoord

Dit proefschrift beschrijft mijn promotie onderzoek dat ik uitgevoerd heb bij de Faculteit der Luchtvaart- en Ruimtevaarttechniek van de Technische Universiteit Delft. Ik ben vele mensen dankbaar die hebben bijgedragen aan de totstandkoming van dit proefschrift, van wie ik er een aantal bij name wil noemen.

Allereerst gaat mijn dank uit naar mijn dagelijkse begeleider, Steven Hulshoff. Ik had geen betere begeleider had kunnen treffen. Zijn continue hulp en motivatie waren onmisbaar gedurende dit onderzoek. Niet alleen op het werk, maar ook daarbuiten ervaar ik onze vriendschap als zeer plezierig. Onze trip naar Albuquerque was het absolute hoogtepunt tijdens mijn promotie onderzoek.

Professor René de Borst dank ik voor zijn rol als promotor en zijn onvoorwaardelijke steun en vertrouwen in mijn onderzoek. Onder zijn leiding onderscheidt de Engineering Mechanics leerstoel zich als een unieke onderzoeksgroep die alles heeft wat een onderzoeker zich kan wensen.

Ook ben ik veel dank verschuldigd aan Erik Jan Lingen. Zijn expertise en ondersteuning op het gebied van de eindige elementen methode en de software-matige implementatie daarvan heeft geleid tot een indrukwekkende code, waar vele toekomstige onderzoekers binnen onze groep van zullen profiteren.

Scott Collis ben ik zeer dankbaar voor onze vruchtbare samenwerking bij Sandia National Laboratories in Albuquerque (NM) in de Verenigde Staten. Deze leerzame ervaring is van groot belang geweest voor mijn onderzoek.

Bendiks Jan Boersma dank ik voor zijn hulp bij het opzetten van de turbulente kanaalstroming en de vele discussies die we hierover gehad hebben.

Verder dank ik al mijn (voormalig) collega's met wie ik een zeer plezierige tijd gehad heb. In het bijzonder wil ik noemen Noël, Sotiris, Christian, Alex, Marcela, Juliana, Ido en Chris. Carla ben ik zeer dankbaar voor haar hulp en medeleven in de vele dingen die op mijn pad kwamen de afgelopen jaren. Ook heb ik een zeer plezierige tijd gehad met Harold, onze talloze luchtige discussies over wederzijdse interesses vormden een zeer aangename en onmisbare afleiding tijdens mijn promotie onderzoek.

Ik dank mijn ouders, broer en zus die altijd voor me klaar gestaan hebben tijdens mijn gehele studie. Zonder hun steun was dit alles nooit mogelijk geweest. Leonie ben ik zeer veel dank verschuldigd voor al haar geduld en de goede zorgen die ik genoten heb de afgelopen jaren. Als laatste gaan mijn gedachten uit naar mijn grootouders, die dit zo graag hadden meegemaakt.

Edwin Munts

Delft, April 2006

Summary

The Variational Multiscale (VMS) method has appeared as a promising new approach to the Large Eddy Simulation (LES) of turbulent flows. The key advantage of the VMS approach is that it allows different subgrid-scale (SGS) modeling assumptions to be made at different ranges of the resolved scales. Typically, in the VMS method, SGS modeling is confined to the smallest resolved scales, leaving the dynamically important large scales free from the direct influence of the SGS model. Prior implementations of the VMS approach have been restricted to either incompressible formulations, simple geometries and/or small time steps. We propose a space-time VMS method for the compressible Navier-Stokes equations, which aims to overcome the difficulties associated with prior VMS implementations. In particular, we aim to develop a method that is applicable to complex flow geometries with a minimum number of degrees of freedom, and that can march at time steps which are chosen to resolve the physical phenomena of interest rather than to satisfy stability constraints.

The spatial discretization of the proposed computational approach corresponds to a high-order continuous Galerkin method, which due to its hierarchical nature provides a natural framework for 'a priori' scale separation, which is crucial for the VMS method. As the method is formulated in a space-time framework, it supports continuous as well as discontinuous discretizations in time. Time-discontinuous discretizations offer great flexibility for adaptation, but may be computationally expensive. Time-continuous discretizations, on the other hand, potentially offer a good com-

promise between accuracy and computational cost. We consider three different time discretizations, viz. a first-order time-continuous Galerkin method (TCG) in time, a second-order time-continuous Petrov-Galerkin method (TCPG) and a third-order discontinuous Galerkin method (TDG).

We consider the efficacy of the spatial VMS discretization for the computation of fully-developed turbulent channel flow. We show that the present method leads to reduced resolution requirements compared to traditional LES approaches applying similar SGS models directly to all the resolved scales. The crucial parameter for obtaining reliable low-order statistics is found to be the large/small partition of the resolved scales. In particular, it is shown that when using simple eddy-viscosity models, the finite element basis functions capable of representing the basic dynamics of the near-wall coherent structures should be released from the direct influence of the SGS model.

As space-time methods are necessarily implicit, a challenge is to ensure that the computations are carried out at reasonable cost. Therefore, we have conducted a detailed performance analysis to investigate the factors that influence the accuracy and computational cost of the proposed methods. For this purpose we consider again the turbulent channel flow.

First, we examine the different time discretizations. It is demonstrated that the TCG method is not a competitive time discretization for the time steps of interest. The TCPG and TDG method, on the other hand, produce accurate and very similar results for relatively large time steps. However, the TDG method is considerably more computationally expensive as it uses twice the number of degrees of freedom compared to the TCPG method. Therefore, except for reasons of adaptation, the TCPG method is preferred here.

Next, we compare the accuracy and cost of different spatial *hp*-resolutions for a similar total number of degrees of freedom. It is shown that the spatially higher-order methods lead to increased accuracy compared to a standard linear Galerkin method

which cannot exploit the advantages of the present VMS formulation. However, higher-order methods are inherently expensive, as the computational work required within a time step scales quadratically with the number of finite element basis functions, while it scales only linear with the number of elements. Higher-order methods also have significantly denser system matrices resulting in rapidly increasing memory requirements with the order of the scheme. As the computational cost associated with higher-order methods is still relatively high, additional research areas are suggested for the goal of improving the method's cost efficiency.

Contents

Voorwoord	iii
Summary	v
1 Introduction	1
1.1 Turbulence simulation	2
1.2 Variational multiscale approach for LES	3
1.3 Space-time variational multiscale method	7
1.4 Research objectives	10
1.5 Thesis outline	11
2 Governing equations	13
2.1 The compressible Navier-Stokes equations	13
2.2 Non-dimensionalization	15
2.3 Quasi-linear form	16
3 Finite-element formulation	19
3.1 Space-time topology	19
3.2 Time-discretization methods	21
3.2.1 Time-continuous Galerkin	22
3.2.2 Time-continuous Petrov-Galerkin	23
3.2.3 Time-discontinuous Galerkin	23

3.3	Finite-element basis	24
3.3.1	Full tensor-product basis	25
3.3.2	Serendipity basis	28
3.3.3	Modal connectivity	29
3.4	Application to convection-diffusion	31
3.4.1	Time-continuous Galerkin	33
3.4.2	Time-continuous Petrov-Galerkin	35
3.4.3	Time-discontinuous Galerkin	36
3.4.4	Temporal convergence	39
3.4.5	Spatial convergence	42
4	Numerical solution of the Navier-Stokes equations	51
4.1	Time-continuous Galerkin	52
4.1.1	Space-time variational form	52
4.1.2	Non-linear system of equations	53
4.2	Time-continuous Petrov-Galerkin	55
4.2.1	Space-time variational form	55
4.2.2	Non-linear system of equations	56
4.3	Time-discontinuous Galerkin	57
4.3.1	Space-time variational form	57
4.3.2	Non-linear system of equations	58
4.4	Predictor multi-corrector algorithm	59
4.5	Assembly of the linear systems	61
4.5.1	Time-continuous Galerkin	61
4.5.2	Time-continuous Petrov-Galerkin	62
4.5.3	Time-discontinuous Galerkin	63
4.6	Solution method for linear systems	64
4.6.1	Iterative solution techniques	65

4.6.2	Jacobian-free Newton Krylov	66
4.6.3	Preconditioning	67
4.7	Application to laminar channel flow	70
5	The variational multiscale method	75
5.1	Variational multiscale formulation	75
5.1.1	Subgrid-scale modeling	78
5.1.2	Implementation	82
5.2	Turbulent channel flow	84
5.2.1	Problem description	84
5.2.2	Physical aspects	85
5.2.3	Boundary conditions and mesh geometry	86
5.2.4	Initial condition	87
5.2.5	Turbulence statistics	88
5.3	Numerical results	90
5.3.1	Coarse mesh results	92
5.3.2	Wall-normal h-refinement	95
5.3.3	Uniform h-refinement	95
5.3.4	Uniform p-refinement	97
5.3.5	Serendipity expansion	100
5.4	Summary	102
6	Performance analysis	121
6.1	Temporal discretization	121
6.1.1	Accuracy	122
6.1.2	Computational cost	122
6.1.3	Discussion	128
6.2	Spatial discretization	130
6.2.1	Accuracy	132

6.2.2	Computational cost	132
6.3	Numerical integration	139
6.4	Convergence criteria	142
6.4.1	Linear system	142
6.4.2	Non-linear system	144
6.5	Summary	145
7	Conclusions and Recommendations	149
7.1	Conclusions	149
7.2	Recommendations	151
	Bibliography	153
A	Coefficient matrices for the Navier-Stokes equations	163
A.1	Conservative variables	164
A.2	Primitive density variables	168
A.3	Primitive pressure variables	173
B	Jacobi polynomials	179
C	Modal connectivity and boundary condition enforcement	181
	Samenvatting	187
	Publications	191
	Curriculum Vitae	193

Chapter 1

Introduction

The calculation of turbulent flows is of great practical importance in many engineering disciplines. A few examples are the turbulent flow around airplanes and ships, turbulent combustion processes in gas turbines, and the well-known atmospheric turbulence experienced daily by many airline passengers. Over the years, turbulent flows have been studied using a variety of analytical, experimental, and numerical techniques.

Analytical techniques are often useful in establishing theories describing the basic physical processes of turbulence. However, due to the complexity of the equations describing the dynamics of turbulence, the Navier-Stokes equations, analytical techniques are of little use for problems of practical interest.

Experimental techniques, on the other hand, serve as a valuable tool providing additional insight into the dynamics of turbulent flows. Moreover, these techniques provide information which is indispensable for the validation of theoretical results. Unfortunately, experimental techniques are often extremely expensive, or even impossible, due to the complexity of many flow configurations of interest.

With the rapid growth of computer power, Computational Fluid Dynamics (CFD) has become an increasingly popular tool, which in addition to experimental and ana-

lytical techniques, has helped to provide a more complete understanding of turbulent flows.

1.1 Turbulence simulation

Turbulent flows are characterized by a very wide range of scales in both space and time. The most straightforward approach to turbulence simulation is to resolve all the scales of motion, which is referred to as Direct Numerical Simulation (DNS). However, the total computational cost of such an approach is on the order of Re^3 [52], where Re represents the Reynolds number, which is typically larger than 10^5 for many flows of practical interest [64, 74]. Therefore, despite the current computer power, DNS is still limited to relatively low Reynolds number flows [57].

An alternative approach to resolving all turbulent scales of motion is to model all or some of their effects. Modeling all the effects of turbulence, while predicting only the mean flow, corresponds to the Reynolds Averaged Navier-Stokes (RANS) approach. Since the RANS approach is relatively inexpensive, it currently serves as the workhorse for industrial applications [12]. The drawback of the RANS approach, however, is its relative inaccuracy, as it is difficult to develop turbulence models which account for all turbulent scales. In general, the effects of the largest turbulent scales cannot be included in universal RANS models, as these are intimately connected to the problem geometry. Inherently, such models cannot be applied to problems where unsteady large-scale structures play a dynamically important role. An extensive discussion of RANS methods can be found in the book by Wilcox [81].

An intermediate approach between DNS and RANS is Large Eddy Simulation (LES), in which the dynamically important large scales are resolved by the numerical simulation. The LES approach is motivated by the fact that the physical processes of interest in many turbulent flows are dominated by a relatively few number of large-scale coherent structures. A classical example is the turbulent mixing layer

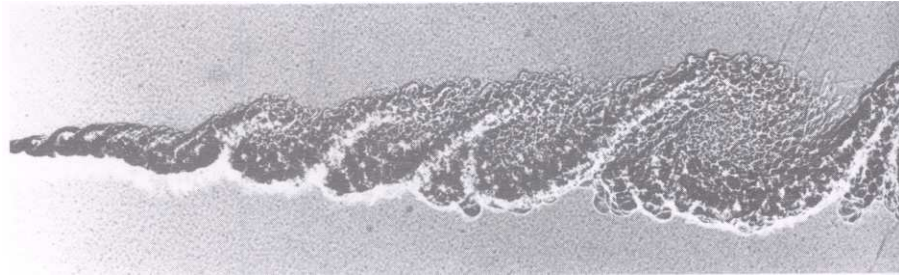


Figure 1.1: Large and small-scale structures in a turbulent mixing layer. (From Brown and Rosco [5].)

as shown in Figure 1.1, where the large-scale Kelvin-Helmholtz rollers are mainly responsible for scalar mixing and momentum transport. Another example is found in wall-bounded turbulence, in which the near-wall low-speed streaks (see Figure 1.2) are mainly responsible for increased momentum transport associated with turbulence induced drag, as well as the process of turbulence production. Since the large scales are solved explicitly in LES, only the effects of the small subgrid scales have to be modeled. This is advantageous, as the small scales have a universal structure [54], making their modeling considerably simpler compared to that of the RANS approach.

Traditionally, in LES formulations equivalent subgrid-scale modeling assumptions are applied to all resolved scales. However, with the development of the variational multiscale formulation for LES, it was shown that improved accuracy can be obtained by allowing different subgrid-scale modeling assumptions at different ranges of the resolved scales. This is discussed in more detail in the following section.

1.2 Variational multiscale approach for LES

The variational multiscale (VMS) method was introduced by Hughes *et al.* [26, 27, 28] as a general framework for deriving models and numerical methods for the simulation of multiscale phenomena present in many engineering problems. Later, the multiscale concepts were extended to turbulence simulation, leading to the development of the

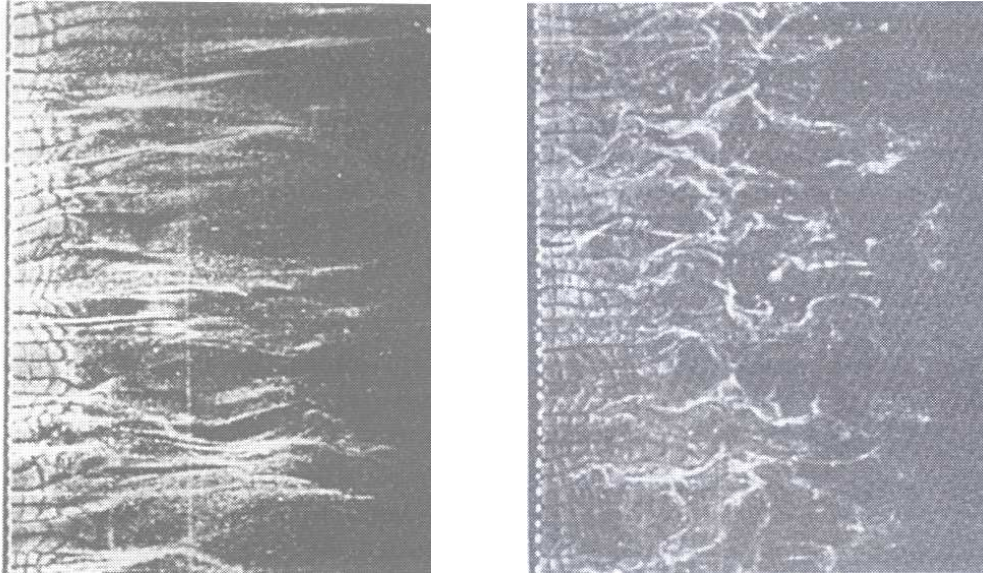


Figure 1.2: Low-speed streaks in a turbulent wall-bounded flow in a water channel. The flow is visualized by hydrogen bubbles in planes parallel to the wall at two different distances from the wall, $y^+ = 2.7$ (left) and $y^+ = 38$ (right). (From Kline *et al.* [41].)

VMS method for LES [29]. As opposed to the traditional approach of filtering [68], the VMS method employs a priori scale decomposition combined with variational projection. In particular, the resolved scales are decomposed *a priori* into large and small scales. Substitution of this decomposition into the governing equations (the Navier-Stokes equations) combined with variational projection leads to a coupled set of equations describing the dynamics of the large and small scales. These equations are referred to as the large and small scale equations. As demonstrated by Collis [10], the key feature of the VMS approach is that it allows for different modeling assumption at each range of the resolved scales. In particular, one typically confines modeling to the small-scale equation. In doing so, numerical consistency is retained in the large-scale equation, allowing full rate of convergence of the large-scale solution. This is in clear contrast to traditional approaches using eddy-viscosity models, which

suffer from reduced convergence rates in all resolved scales due to model-related artificial-viscosity effects. Meanwhile, the VMS approach for LES has become an increasingly popular topic of research.

Employing a global spectral discretization, the VMS method was first applied to decaying homogeneous isotropic turbulence [30]. Using a simple constant Smagorinsky model [73] only on the small scales, the VMS method proved superior to traditional LES formulations using a similar model, as well as a far more complex dynamic Smagorinsky model [18]. Next, the VMS method was applied to equilibrium and non-equilibrium turbulent channel flows [31, 55], employing a global spectral discretization in the homogeneous direction and a Galerkin method using Legendre polynomials in the wall-normal direction. Again, the VMS method using a constant-coefficient Smagorinsky model acting on the small scales outperformed a traditional LES employing dynamic modeling. Similar conclusions were found by Ramakrishnan and Collis [60] studying opposition control in turbulent channel flow, where the VMS method also showed better predictive capabilities.

The VMS discretizations discussed so far employ global spectral methods, which allow a natural scale decomposition as they approximate the solution using a series of sinusoidal functions. The drawback of such methods, however, is that they are suitable only for problems with periodic boundary conditions. A VMS discretization suitable for more realistic problems was introduced by Jansen *et al.* [35], in which a continuous Galerkin approach employing a mesh-entity-based hierarchical basis was successfully applied to decaying isotropic turbulence.

An alternative approach, suitable for complex flow geometries, was proposed by Collis [12], who merged the VMS method with a high-order accurate discontinuous Galerkin (DG) method. Investigations using this DG-VMS combination for low Reynolds number turbulent channel flow [61] confirmed a potential advantage of the VMS method, viz. if the large scales are sufficiently refined to resolve the most important coherent structures, the constant-coefficient Smagorinsky model acting on

the small scales does not adversely affect the solution quality.

Another VMS formulation capable of handling unstructured domains was introduced by Koobus and Farhat [43], who employed a finite volume/element method on tetrahedral meshes, in which scale separation was based on cell agglomeration. Results obtained using this VMS method for a vortex shedding flow past a square cylinder were in better agreement with experimental data than those predicted by a traditional LES formulation.

So far, all the methods discussed use constant-coefficient modeling procedures to account for the effect of the unresolved scales. This requires a user-defined parameter, which in general is undesirable. Therefore, in order to further investigate the robustness of the VMS approach, Holmen *et al.* [23] extended the dynamic model [18] to the VMS formulation. It was shown that the VMS method in combination with the dynamic model is relatively insensitive to the partition between large and small resolved scales compared to the static modeling procedure. The static model procedure, on the other hand, achieved the highest absolute accuracy at its optimal partition.

The VMS method was applied to a discretization for channel flows which defined scale separation in the homogeneous directions only by Ramakrishnan and Collis [61, 63]. It was found that applying the VMS method only in the homogeneous directions is as successful as applying it in all directions. The partition between large and small resolved scales as well as the overall resolution were found to be crucial parameters for obtaining high-quality solutions. Applying the VMS approach in only one of the homogeneous directions led to a consistent method for choosing the partition and resolution requirements, in relation to the physically important structures in the flow.

It was shown by Vreman [78] that the VMS method can also be extended to the classical filtering approach to LES. This led to the observation that multiscale methods based on standard eddy-viscosity models are closely related to (anisotropic)

hyperviscosity models. Using this approach for turbulent channel flow, several multi-scale models were found to be as accurate as the standard dynamic model, while being simpler to implement. A similar filter-based approach to VMS was also used by Levasseur *et al.* [45] for decaying isotropic turbulence. Using a low-order Galerkin/least-squares method provided appealing results for decaying isotropic turbulence.

An extensive investigation of the DG-VMS combination, labeled the *local VMS method* (ℓ VMS), was performed by Ramakrishnan [64]. It was found that the introduction of the VMS approach leads to increased efficiency in terms of degrees of freedom for turbulent channel flow compared to traditional LES formulations using similar subgrid-scale models. Particularly good results were obtained using ℓ VMS combined with polynomial de-aliasing [40], a technique to reduce high-wavenumber aliasing errors from the solution. Additionally, relating the partition parameter to the important length scales in the flow [63] was also shown to be successful for ℓ VMS.

Although ℓ VMS is an attractive method for a variety of reasons, a major disadvantage of discontinuous Galerkin methods in general is the high computational cost due to the large number of degrees of freedom [70], and the necessity of evaluating element-boundary integrals. The latter can be of significant expense in multi-dimensional unstructured domains. Additionally, the use of explicit time discretization such as used in ℓ VMS leads to severe time steps restrictions due to stability requirements. This is unfortunate, as one is often interested in large-scale quantities such as average lift and drag, which are not significantly influenced by small time scales.

1.3 Space-time variational multiscale method

In the current research, we aim to develop a space-time VMS method which is applicable to complex geometries with a minimum number of degrees of freedom and can march at time steps chosen to resolve the physical phenomena of interest, rather than to satisfy the stability constraints. To this end, we propose a space-time spectral ele-

ment method as the basis of our computational approach to turbulence simulation. In particular, the spatial discretization corresponds to a high-order continuous Galerkin method, which is advantageous in that it employs significantly less degrees of freedom for a given degree of the polynomial approximation space than the corresponding discontinuous Galerkin formulation. Therefore, despite some reduced flexibility compared to discontinuous Galerkin methods, it is believed that continuous methods offer more potential for large-scale computations.

As space-time formulations are necessarily implicit, it is hoped that relatively large time steps are permitted without negatively affecting the large-scale quantities of interest. An additional advantage of space-time formulations is that they provide a natural and accurate treatment of moving domains. As opposed to the alternative Arbitrary Lagrangian-Eulerian (ALE) formulations [15], space-time formulations allow a straightforward satisfaction of the Geometric Conservation Law (GCL) [44]. The GCL states that a uniform flow on a moving mesh should be exactly represented by the discretization. It can be shown [16] that increased accuracy and stability is obtained when the GCL is satisfied discretely. However, for ALE formulations employing arbitrary combinations of spatial and temporal discretization methods, this can be a laborious task.

Space-time finite element formulations may be continuous or discontinuous in time, both of which we consider in the present investigation. The time-discontinuous Galerkin method was developed by Hughes *et al.* [24, 25, 32] for fluid and solid mechanics problems. An extensive analysis of this method in the context of the compressible Euler and Navier-Stokes equations was performed by Shakib [71, 72]. As observed by Johnson [37], the mathematical properties of the time-discontinuous Galerkin method for unsteady problems are completely analogous to those of the Galerkin method for steady problems. Consequently, a good numerical method for steady problems can be readily extended to unsteady problems in a space-time formulation. One of the key features of the time-discontinuous method is its flexibility for

adaptivity. Due to the time-discontinuous nature, the spatial mesh (h -refinement) and/or solution representation (p -refinement) can be changed from one time step to the next without difficulty. The time-discontinuous Galerkin method has been successfully applied to various incompressible [75] and compressible [1, 51, 71] flow problems. As discussed by Hansbo [20], however, a major drawback of the time-discontinuous Galerkin method is that the lowest possible order of temporal approximation, i.e. piecewise constant, leads to a variant of the backward Euler method, which is only first-order accurate. In general, this is insufficient for time-accurate solutions. Increasing the temporal approximation and weighting functions to piecewise linear leads to a third-order accuracy in time [71]. However, this scheme is relatively expensive as it requires twice as many degrees of freedom as the first-order scheme.

Alternatively, a good compromise between accuracy and efficiency can be obtained by a continuous Petrov-Galerkin time-discretization [70]. A Petrov-Galerkin method is defined as a Galerkin method which employs weighting functions which are different from the solution approximation functions. In particular, a second-order accurate time discretization is obtained by piecewise-continuous weighting functions together with a piecewise-linear solution approximation in time. This method was first analyzed by Aziz and Monk [2], but the concept dates back to a number of authors, such as Winther [82], Hulme [33], and Zienkiewicz [84]. The advantage of this method is that it employs half the number of degrees of freedom compared to the third-order time-discontinuous Galerkin scheme, but at the price of a reduced order of accuracy and reduced adaptive capabilities.

Based on the above observations, we consider the third-order accurate time-discontinuous (TDG) Galerkin method, and the second-order Petrov-Galerkin (TCPG) method. Additionally, we consider a time-continuous Galerkin (TCG) method that is only first-order accurate, which uses a linear solution approximation in time.

1.4 Research objectives

The main objective of the present research is to evaluate the capabilities of the proposed space-time VMS discretization, as to whether it constitutes a suitable candidate for accurate and efficient turbulence simulation. In this process, we aim to identify the strengths and weaknesses in terms of accuracy and computational cost of the current computational approach.

As a first investigation of the proposed formulation, we consider its performance for fully-developed turbulent channel flow. Here, we focus on the effect of subgrid-scale modeling together with resolution requirements, which leads to guidelines for obtaining reasonable low-order turbulence statistics. In this process, we compare results obtained using the present VMS formulation, in which modeling is confined to the small resolved scales, to those obtained using no subgrid-scale model as well as those obtained applying similar subgrid-scale models to all resolved scales.

Next, we focus on the computational cost of the present approach. As space-time methods are implicit, a challenge is to ensure that the computations can be carried out at reasonable cost. To solve the non-linear systems of equations resulting from the discretization, we employ a predictor multi-corrector algorithm. For each time step, this algorithm requires the solution of a sequence of linear systems of equations. Assembling and solving the linear systems in an efficient way is crucial for application to large-scale problems. Therefore, we investigate the factors that influence the computational cost and accuracy for turbulent channel flow simulation. In this process, we compare the different time discretizations, and different spatial hp -resolutions for similar number of degrees of freedom. Additionally, we investigate the effect of the quadrature order and convergence criteria on the accuracy and cost of the simulation.

1.5 Thesis outline

The remainder of the thesis is outlined as follows. Chapter 2 presents the governing equations describing the dynamics of the fluid, i.e. the compressible Navier-Stokes equations. Next, Chapter 3 gives a detailed description of the present space-time finite element formulation. This chapter ends with an application to the convection-diffusion equation. Subsequently, Chapter 4 describes the solution algorithm that is used to solve the non-linear system of equations resulting from the finite-element discretization of the compressible Navier-Stokes equations. This chapter ends with an application to laminar channel flow. Chapter 5 focuses on the implementation of the variational multiscale concept within the current formulation. The resulting method is evaluated using turbulent channel flow with the emphasis on *hp*-discretization requirements and turbulence-modeling parameters. Chapter 6 addresses the computational cost-efficiency of the considered time discretizations, as well as that of *hp*-resolution, parallel computing and quadrature order. Finally, conclusions and directions for future research are presented in Chapter 7.

Chapter 2

Governing equations

In this chapter, we present the compressible Navier-Stokes (NS) equations, which are the governing equations for the viscous flow computations presented in later chapters. These equations, representing the conservation of mass, momentum and total energy, constitute the most general equations describing the dynamics of fluid flow. First, in Section 2.1 we describe the NS equations in dimensional form. Next, in Section 2.2 we present the non-dimensionalization that is employed in the current work. We conclude with a description of the quasi-linear form of the NS equations in Section 2.3.

2.1 The compressible Navier-Stokes equations

The compressible Navier-Stokes equations [80] in conservation form can be written as

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathbf{F}_{i,i}^v = \mathbf{S} \quad \text{in } \Omega, \quad (2.1a)$$

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x}), \quad (2.1b)$$

where $\mathbf{U} = \{\rho, \rho\mathbf{u}, \rho e\}^T$ is the vector of conservative variables, ρ the fluid density, $\mathbf{u} = \{u, v, w\}^T$ the fluid velocity vector, and $e = i + \frac{1}{2}u_i u_i$ the total energy per unit

mass, where i the internal energy per unit mass. In the above equations the Einstein summation convention for repeated indices is employed, and subscripts following a comma denote differentiation with respect to the subscript. The inviscid fluxvectors $\mathbf{F}_i(\mathbf{U})$, the viscous flux vectors $\mathbf{F}_i^v(\mathbf{U})$, and the source vector \mathbf{S} are defined as

$$\mathbf{F}_i(\mathbf{U}) = u_i \mathbf{U} + p \begin{pmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{pmatrix}, \quad \mathbf{F}_i^v(\mathbf{U}) = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij}u_j - q_i \end{pmatrix}, \quad \mathbf{S} = \rho \begin{pmatrix} 0 \\ f_1 \\ f_2 \\ f_3 \\ f_i u_i \end{pmatrix}, \quad (2.2)$$

where p is the thermodynamic pressure, δ_{ij} is the Kronecker delta, τ_{ij} is the viscous stress tensor, q_j is the heat flux vector, and f_i is a body force per unit mass. The system of equations (2.1a) is closed by the following constitutive relations.

The fluid is assumed to be a Newtonian fluid, such that the shear stress is described by

$$\tau_{ij} = 2\mu(S_{ij} - \frac{\delta_{ij}}{3}S_{kk}), \quad (2.3)$$

where μ is the molecular viscosity coefficient, and $S_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$ is the strain-rate tensor.

The heat flux due to thermal conduction is computed using Fourier's law

$$q_i = -\kappa T_{,i}, \quad (2.4)$$

where κ is the thermal conductivity coefficient, and T is the absolute temperature.

The equation of state is used to relate the thermodynamic variables. It is assumed that the fluid is a perfect gas (i.e. intermolecular forces are negligible) so that

$$p = (\gamma - 1)\rho i, \quad (2.5)$$

where $\gamma = c_p/c_v$ with c_p and c_v the specific heat at constant pressure and constant volume, respectively. This equation is referred to as the thermal equation of state.

If c_p and c_v are assumed constant, then the internal energy can be related to the absolute temperature by

$$i = c_v T, \quad (2.6)$$

which is referred to as the caloric equation of state. For a calorically perfect gas, the fluid properties (viscosity and thermal conductivity coefficients) depend on temperature only. For low-speed flows with small temperature variations, the fluid properties can be assumed constant with negligible loss of accuracy. This assumption is used for all results presented here.

The Navier-Stokes equations are solved subject to appropriate initial and boundary conditions. These are specified for each specific problem in subsequent chapters.

2.2 Non-dimensionalization

The Navier-Stokes equations can be cast in non-dimensional form by defining the following dimensionless variables:

$$\begin{aligned} x^* &= x/L_r, & u_i^* &= u_i/U_r, & t^* &= tU_r/L_r, & \rho^* &= \rho/\rho_r, \\ p^* &= p/\rho_r U_r^2, & T^* &= Tc_v/U_r^2, & f_i^* &= f_i L_r/U_r^2, \end{aligned} \quad (2.7)$$

where L_r , U_r and ρ_r are the predefined reference length, velocity and density, respectively. Substitution of these relations into the Navier-Stokes equations leads to its non-dimensional form, which is equal to (2.1a) – (2.2), with the dimensional quantities replaced by their non-dimensional counterparts (2.7). The constitutive relations in non-dimensional form then become

$$\tau_{ij}^* = \frac{2}{\text{Re}} (S_{ij}^* - \frac{\delta_{ij}}{3} S_{kk}^*), \quad (2.8)$$

$$p^* = (\gamma - 1) \rho^* T^*, \quad (2.9)$$

$$q_i^* = -\frac{\gamma}{\text{RePr}} T_{,i}^*, \quad (2.10)$$

$$i^* = T^*, \quad (2.11)$$

where $\text{Re} = \rho_r U_r L_r / \mu$ is the Reynolds number, and $\text{Pr} = \mu c_p / \kappa$ is the Prandtl number. In the present results the fluid is assumed to be air for which $\text{Pr} = 0.7$ [69]. Additionally, the non-dimensionalization gives rise to the Mach number, defined by $\text{M} = U_r / a_r$, where $a_r = (\gamma p_r / \rho_r)^{1/2}$ is the reference speed of sound. Note that with the present non-dimensionalization (2.7), the Mach number does not explicitly appear in the non-dimensional Navier-Stokes equations, however, it enters the simulation through the specification of the boundary conditions. For example, a constant wall temperature T_w^* can be specified as follows

$$T_w^* = \frac{T c_v}{U_r^2} = \frac{1}{\gamma(\gamma - 1)\text{M}^2}, \quad (2.12)$$

where we have used the relations $c_v = \mathcal{R}/(\gamma - 1)$ and $a^2 = \gamma \mathcal{R}T$, with \mathcal{R} is the gas constant.

2.3 Quasi-linear form

A useful form of the Navier-Stokes equations is the quasi-linear form, which is defined as

$$\mathbf{U}_{,t} + \mathbf{A}_i \mathbf{U}_{,i} - (\mathbf{K}_{ij} \mathbf{U}_{,j})_{,i} = \mathbf{S}, \quad (2.13)$$

where $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$ are the inviscid flux Jacobians and \mathbf{K}_{ij} are the diffusivity matrices, satisfying $\mathbf{K}_{ij} \mathbf{U}_{,j} = \mathbf{F}_i^v$. The coefficients for both are given in Appendix A.

As discussed in [21], the Navier-Stokes equations in conservation form can be solved using any set of variables by introducing a transformation $\mathbf{U} = \mathbf{U}(\mathbf{Y})$, where \mathbf{Y} is a new set of variables. Using this transformation (2.13) can be rewritten in terms of \mathbf{Y} as

$$\tilde{\mathbf{A}}_0 \mathbf{Y}_{,t} + \tilde{\mathbf{A}}_i \mathbf{Y}_{,i} - (\tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j})_{,i} = \mathbf{S}, \quad (2.14)$$

where

$$\tilde{\mathbf{A}}_0 = \mathbf{U}_{,\mathbf{Y}}, \quad (2.15)$$

$$\tilde{\mathbf{A}}_i = \mathbf{A}_i \tilde{\mathbf{A}}_0, \quad (2.16)$$

$$\tilde{\mathbf{K}}_{ij} = \mathbf{K}_{ij} \tilde{\mathbf{A}}_0. \quad (2.17)$$

It can be shown [71] that a Galerkin finite-element solution using entropy variables satisfies the Clausius-Duhem inequality, which is the basic nonlinear stability condition for the compressible Navier-Stokes equations. This means that such a solution automatically inherits the entropy-production property of the compressible Navier-Stokes equations. However, the advantage of this property in terms of the accuracy of numerical solutions is not completely clear. Le Beau et al. [4] investigated stabilized finite-element formulations of the compressible Euler equations using entropy and conservative variables. Solutions were compared for several problems in the subsonic, transonic and supersonic range. It was found that solutions obtained from both sets of variables were very close and in some cases indistinguishable. A comparative study of different sets of variables for the compressible Navier-Stokes equations was also performed by Hauke et al. [21]. In addition to entropy and conservative variables the use of primitive pressure variables, i.e. $\mathbf{Y} = \{p, \mathbf{u}, T\}^T$, and primitive density variables, i.e. $\mathbf{Y} = \{\rho, \mathbf{u}, T\}^T$, was investigated in a stabilized finite-element formulation. It was found that conservative and entropy variables yielded the least accurate results for low-speed channel flows, whereas the use of both primitive variables were more accurate. It was noted that the differences are noticeable for coarse grids only. With sufficient refinement all methods achieve similar accuracy. An advantage of using primitive variables is that it simplifies the specification of certain types of boundary conditions. For example, an isothermal solid wall boundary condition (i.e. $u_i = 0, T = T_w$) for the Navier-Stokes equations as commonly used for channel flows can be directly imposed in terms of the primitive variables. The coefficient matrices (i.e. the flux Jacobians and diffusivity matrices) for conservative variables, as well as

for pressure- and density-primitive variables, are given in Appendix A.

Chapter 3

Finite-element formulation

In this chapter, we describe the finite-element formulations which will be investigated for application to LES. First, in Section 3.1 we define the space-time topology that is used in the present simulations. The present space-time formulation allows for different types of time discretizations, which can be either continuous or discontinuous in time. Here, we consider two time-continuous methods and one time-discontinuous method, as described in Section 3.2. Next, in Section 3.3 we present the spatial finite-element basis which is used to formulate the present VMS method. Finally, in Section 3.4 we consider an application to the linear convection-diffusion equation, where we compare the computational efficiency of the different spatial and temporal discretizations.

3.1 Space-time topology

The fixed spatial domain for the problem is denoted by Ω , which is an open, bounded subset of \mathbb{R}^d , with $d \in \{1, 2, 3\}$ the spatial dimension, with boundary $\partial\Omega$. Let the time interval of interest $I = (0, T)$ be partitioned into N time slabs $I_n = (t_n, t_{n+1})$.

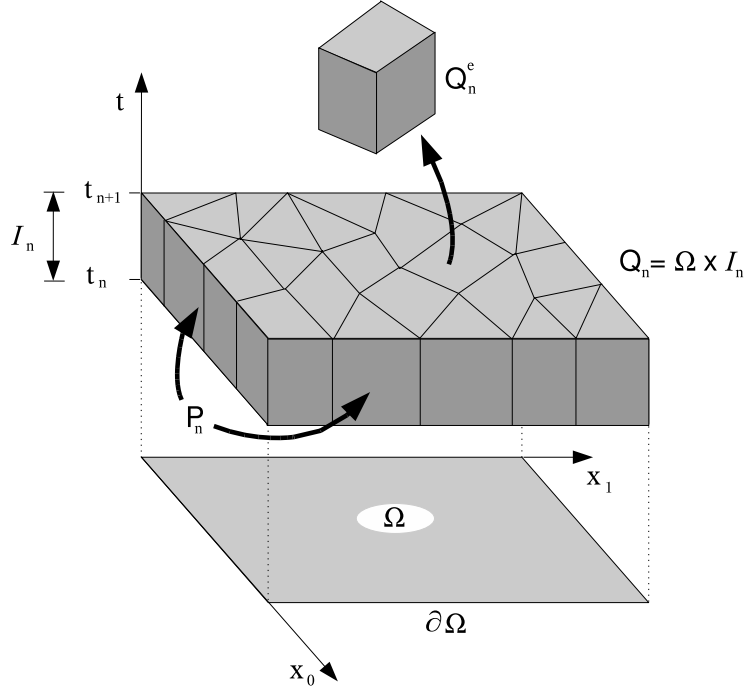


Figure 3.1: Sketch of a space-time slab

A space-time slab is then defined as

$$Q_n = \Omega \times I_n, \quad (3.1)$$

with lateral boundary

$$P_n = \partial\Omega \times I_n, \quad (3.2)$$

as shown in Figure 3.1.

Let the spatial domain Ω for the n^{th} space-time slab be subdivided into $(n_{el})_n$ elements Ω_n^e , $e = 1, \dots, (n_{el})_n$. The space-time elements for the n^{th} slab are then defined as

$$Q_n^e = \Omega_n^e \times I_n. \quad (3.3)$$

Let $\mathcal{P}_k(\hat{Q})$ be the space of polynomials of degree $\leq k$ defined on the master element

$\hat{\mathcal{Q}} = (-1, 1)^{d+1}$, then

$$\mathcal{P}_k(\mathcal{Q}^e) = \left\{ \phi \mid \phi = \hat{\phi} \circ \mathcal{T}, \hat{\phi} \in \mathcal{P}_k(\hat{\mathcal{Q}}) \right\}, \quad (3.4)$$

where $\mathcal{T} : \hat{\mathcal{Q}} \mapsto \mathcal{T}(\hat{\mathcal{Q}})$ denotes the mapping from the master element to physical space.

3.2 Time-discretization methods

The space-time finite-element formulation provides many options for representing the evolution of the solution in time. Time-discontinuous formulations can be very accurate, however, they are typically expensive. Alternatively, a good compromise might be obtained from a time-continuous formulation. Therefore, in the present work we consider two time-continuous discretizations, and one time-discontinuous discretization.

To clarify the structure and order of accuracy of the different time discretizations, we consider the following ordinary differential equation

$$y'(t) = f(y, t), \quad (3.5a)$$

$$y(0) = y_0, \quad (3.5b)$$

where the prime denotes differentiation with respect to time. Multiplying (3.5a) by the weighting function and integrating the result over the time interval $[t_n, t_{n+1}]$, yields the space-time variational formulation

$$w(t_{n+1})y(t_{n+1}) - w(t_n)y(t_n) = \int_{t_n}^{t_{n+1}} w(t)f(y, t) + w'(t)y(t) dt, \quad (3.6)$$

where we have applied integration by parts. In the following sections, we consider the time discretizations obtained by choosing the appropriate solution approximation and weighting functions.

3.2.1 Time-continuous Galerkin

The first time discretization we consider is a time-continuous Galerkin (TCG) method, which employs equal piecewise-linear solution approximation and weighting functions on the time interval $[t_n, t_{n+1}]$

$$y^h(t) = \pi_n^-(t)y_n + \pi_n^+(t)y_{n+1}, \quad (3.7)$$

$$w^h(t) = \pi_n^-(t)w_n + \pi_n^+(t)w_{n+1}, \quad (3.8)$$

where y_n and y_{n+1} are the solution coefficients at times t_n and t_{n+1} , and

$$\pi_n^-(t) = \frac{t_{n+1} - t}{\Delta t}, \quad \pi_n^+(t) = \frac{t - t_n}{\Delta t}, \quad (3.9)$$

with $\Delta t = t_{n+1} - t_n$, are the temporal basis functions. Since this method is continuous in time, the solution at time t_n corresponds to a Dirichlet condition, such that $w_n = 0$. Note that this ensures an equivalent number of equations and unknowns. For a linear ODE (3.5a) with $f(y, t) = \alpha y$, substitution of (3.7) and (3.8) into the variational formulation (3.6) gives the following discrete relation

$$y_{n+1} = \alpha \left(\frac{h}{6} y_n + \frac{h}{3} y_{n+1} \right) + \frac{1}{2} (y_n + y_{n+1}). \quad (3.10)$$

Substituting the Taylor series expansion for y_{n+1}

$$y_{n+1} = y_n + hy' + \frac{h^2}{2}y'' + h.o.t., \quad (3.11)$$

where *h.o.t.* refers to higher-order terms, into (3.10) leads to the *modified equation*, given by

$$y' = \alpha y + h \left(\frac{2}{3} \alpha y' - \frac{1}{2} y'' \right) + h^2 \frac{\alpha}{3} y'' + h.o.t. \quad (3.12)$$

This expression shows that the method is only first-order accurate in time, despite the linear representation of the solution over the time interval. As discussed in [22], this method corresponds to the θ -method with $\theta = 2/3$.

3.2.2 Time-continuous Petrov-Galerkin

The second time discretization we consider is a time-continuous Petrov-Galerkin (TCPG) method, which employs a piecewise-linear solution approximation and a piecewise-constant weighting function on the time interval $[t_n, t_{n+1}]$

$$y^h(t) = \pi_n^-(t)y_n + \pi_n^+(t)y_{n+1}, \quad (3.13)$$

$$w^h = w_{n+1}, \quad (3.14)$$

As opposed to the TCG method discussed in the previous section, the weighting function does not vanish at time t_n . Note that constraining the solution at time t_n ensures an equivalent number of equations and degrees of freedom.

Assuming a linear ODE (3.5a) with $f(y, t) = \alpha y$ and substituting (3.13) and (3.14) into the variational formulation (3.6), where the second term in the integral vanishes since $w' = 0$, we obtain the following expression

$$y_{n+1} - y_n = \alpha \frac{h}{2} (y_n + y_{n+1}). \quad (3.15)$$

Note that this expression corresponds to the trapezoidal integration rule, which is second-order accurate [48]. This is easily verified by the modified equation, which is obtained by substituting (3.11) into (3.15)

$$y' = \alpha y + \frac{\alpha h^2}{4} y'' + h.o.t. \quad (3.16)$$

For general linear partial differential equations this method corresponds to the classical Crank-Nicolson scheme [70]. For a more detailed analysis of this time discretization, we refer to Aziz and Monk [2], Winther [82], Hulme [33], and Zienkiewicz [84].

3.2.3 Time-discontinuous Galerkin

The third time discretization we consider is the time-discontinuous Galerkin (TDG) method. Similarly to the TCG method, the TDG method employs equal piecewise-

linear solution approximation and weighting functions on the time interval $[t_n, t_{n+1}]$

$$y^h(t) = \pi_n^-(t)y_n + \pi_n^+(t)y_{n+1}, \quad (3.17)$$

$$w^h(t) = \pi_n^-(t)w_n + \pi_n^+(t)w_{n+1}, \quad (3.18)$$

however, without the restriction the solution is constrained at time t_n , i.e. $w_n \neq 0$. Instead, the solution is discontinuous across the time intervals at times $t = t_n$, $n \in \{1, 2, \dots, N\}$, and the solution at t_n has to be determined as part of the solution. To advance the solution from one time interval to the next, a *jump-term* is added to the left-hand side of the variational statement (3.6), defined by

$$w^h(t_n^+) \llbracket y(t_n) \rrbracket, \quad (3.19)$$

where

$$\llbracket y(t_n) \rrbracket = y(t_n^+) - y(t_n^-) \quad \text{and} \quad w^h(t_n^\pm) = \lim_{\epsilon \rightarrow 0^\pm} w^h(t_n + \epsilon). \quad (3.20)$$

The jump-term provides a weak initial condition for the space-time slab. Therefore, the mesh and/or solution representation can be easily changed from one time step to the next. However, this method is inherently computationally expensive as it employs twice the number of degrees of freedom compared to the continuous methods discussed above. It can be shown that this method is third-order accurate, however, this is an elaborate task and we refer to [71] for the details.

3.3 Finite-element basis

In this section, we describe the spatial discretization that is used in the present work. Traditionally, finite-element methods make use of Lagrangian approximation bases, which consist of a set of (nodal) polynomial basis functions of equal degree. Consequently, these bases do not readily lend themselves to scale decomposition, such as required by the VMS method. A more appropriate basis for scale decomposition is a so-called hierarchical basis, in which the low-order basis functions are a subset of

the high-order basis functions. In addition to the capability of scale decomposition, we also require the basis to be applicable to complex domains. Therefore, we consider a hierarchical high-order polynomial basis, as presented in the following section.

3.3.1 Full tensor-product basis

In order to meet the criteria stated above, we consider a spatially continuous high-order spectral element method based on the modal p -type basis (Karniadakis and Sherwin [38]). This basis and the corresponding solution approximation on a one-dimensional master element ($-1 < \xi < 1$) are defined as

$$\phi_p(\xi) \mapsto \psi_p(\xi) = \begin{cases} \left(\frac{1-\xi}{2}\right) & p = 0, \\ \left(\frac{1-\xi}{2}\right) \left(\frac{1+\xi}{2}\right) P_{p-1}^{\alpha,\beta}(\xi) & 0 < p < P, \\ \left(\frac{1+\xi}{2}\right) & p = P, \end{cases} \quad (3.21)$$

$$u^h = \sum_{p=0}^P \phi_p(\xi) u_p \quad (3.22)$$

where $P_{p-1}^{\alpha,\beta}(\xi)$ are Jacobi polynomials of degree $p-1$, for example Legendre polynomials ($\alpha = \beta = 0$) or Chebychev polynomials ($\alpha = \beta = -\frac{1}{2}$). Although in fact any polynomial can be used, here we employ Jacobi polynomials with $\alpha = \beta = 1.0$ as they maintain a high degree of orthogonality [38] which is advantageous for the conditioning of the resulting system of equations. Jacobi polynomials can be constructed using a three-term recursive relation as described in Appendix B, along with miscellaneous formulas for their differentiation.

Note that this basis consist of a hierarchy of polynomials of increasing degree, as shown in Figure 3.2 for $P = 5$. As the polynomial degree increases, smaller-scale behavior can be captured by the corresponding basis functions. Therefore, we base the scale decomposition in the present VMS approach on the polynomial degree, where low polynomial degrees represent large-scale behavior, and the remaining high

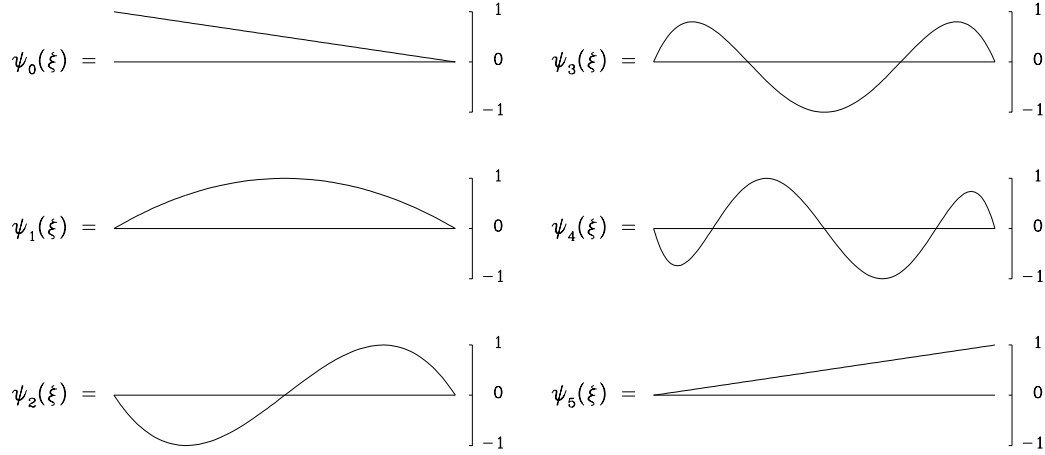


Figure 3.2: Shape of the one-dimensional modal expansion for a polynomial degree $P = 5$ and $\alpha = \beta = 1.0$. (From Karniadakis and Sherwin [38], reproduced with permission.)

polynomial degrees represent small-scale behavior. The details of the implementation within the VMS formulation are presented in Chapter 5.

For rectangular elements, the multi-dimensional modal bases can be simply constructed by a product of the one-dimensional bases in each of the coordinate directions. For example, the finite element basis functions and corresponding solution approximation on a two-dimensional master element ($-1 < \xi_1, \xi_2 < 1$) can be written as

$$\phi_{pq}(\xi_1, \xi_2) = \psi_p(\xi_1)\psi_q(\xi_2), \quad (3.23)$$

$$u^h(\xi_1, \xi_2) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \phi_{pq}(\xi_1, \xi_2)u_{pq}, \quad (3.24)$$

where P_2 is the maximum polynomial degree of the basis functions in the ξ_2 -direction, and u_{pq} is the finite-element approximation coefficient. Figure 3.3 demonstrates the basis functions obtained for a two-dimensional basis in a quadrilateral master element with $P_1 = P_2 = 4$. In cases where the same polynomial degree in each spatial direction

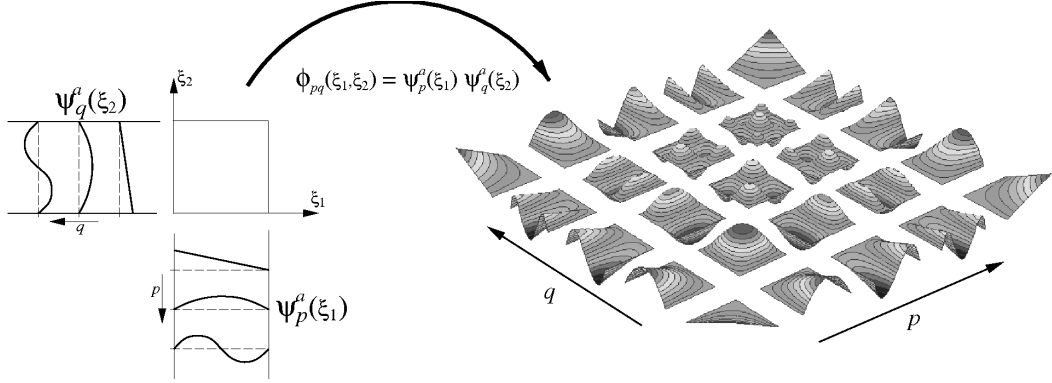


Figure 3.3: Construction of a two-dimensional modal expansion basis from the product of two one-dimensional expansions of degree $P_1 = P_2 = 4$. (From Karniadakis and Sherwin [38], reproduced with permission.)

is used, we define $P_1 = P_2 = P$.

It is convenient to write the multi-dimensional elemental solution approximation as a single summation over the element basis functions. For example, the two-dimensional elemental solution approximation can be written as

$$u^h = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \phi_{pq} u_{pq} = \sum_{a=1}^{n_{eb}} N_a u_a, \quad (3.25)$$

where N_a represents the a^{th} elemental basis function, such that the index a corresponds to a unique combination of p and q , and $n_{eb} = (P_1 + 1)(P_2 + 1)$ is the number of basis functions defined on the element. In the following, we always use lower case indices for local (elemental) orderings.

Likewise, the space-time basis functions are obtained by the tensor product of the spatial basis functions with the temporal basis functions, as defined in Section (3.2). Note that for three spatial dimensions, we obtain four-dimensional space-time finite element basis.

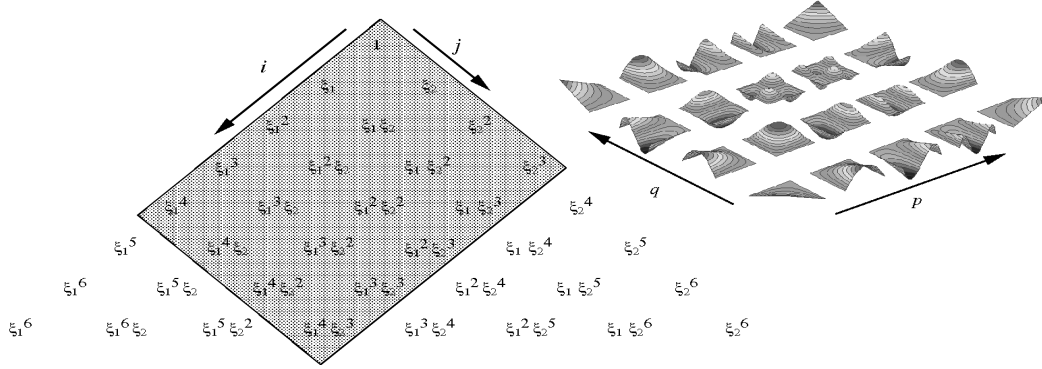


Figure 3.4: Polynomial space in terms of the Pascal triangle of the full tensor product of a quadrilateral modal expansion with $P_1 = 4$ and $P_2 = 3$. (From Karniadakis and Sherwin [38], reproduced with permission.)

3.3.2 Serendipity basis

In addition to the modal basis obtained from the full tensor product, we also consider its corresponding serendipity expansion basis. This is an interesting variant since it allows for a reduced number of basis functions compared to the full modal basis, while maintaining the same order of accuracy.

The polynomial space spanned by the full modal basis on a two-dimensional master element ($\hat{\Omega}$) can be expressed as

$$\phi_{pq}(\xi_1, \xi_2) \subseteq \mathcal{P}_k(\hat{\Omega}) = \text{span}\{\xi_1^i \xi_2^j\}_{(i,j) \in \mathcal{I}_{2D}} \quad (3.26)$$

$$\mathcal{I}_{2D} = \{(i, j) \mid 0 \leq i \leq P_1, 0 \leq j \leq P_2\}. \quad (3.27)$$

Pascal's triangle for this basis with $P_1 = 4$ and $P_2 = 3$, shown in Figure 3.4, demonstrates that terms are included with a higher polynomial degree than strictly necessary to span a two-dimensional polynomial basis of a given degree. In fact, only the terms up to the corresponding horizontal line in the Pascal triangle are required. The serendipity basis consists of only those modes necessary for a given degree of the

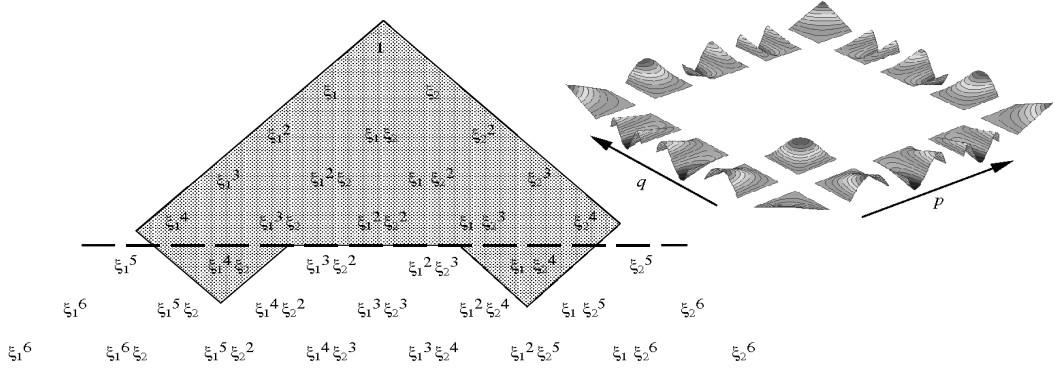


Figure 3.5: Polynomial space in terms of the Pascal triangle of a modal serendipity basis for $P = 4$. (From Karniadakis and Sherwin [38], reproduced with permission.)

polynomial space. Here, we only consider serendipity bases in the spatial directions with a uniform polynomial degree P . The two-dimensional serendipity basis is given by (3.26) together with

$$\mathcal{I}_{2D} = \{(i, j) \mid 0 \leq (i, j) \leq P, i + j \leq P\} \quad (3.28)$$

Figure 3.5 shows an example of the Pascal triangle and modes for the $P = 4$ serendipity basis. Note that a strict horizontal line in the Pascal triangle cannot be obtained, because the boundary modes corresponding to the $\xi_1^4 \xi_2$ and $\xi_1 \xi_2^4$ have to be retained for completeness. The three-dimensional serendipity basis is constructed in a similar fashion. Likewise, once the spatial serendipity basis is defined, the corresponding space-time basis is simply obtained by employing the tensor-product of the spatial basis functions with that of the temporal basis functions.

3.3.3 Modal connectivity

In the present study we are interested in approximations of the governing equations which are globally \mathcal{C}_0 -continuous within each space-time slab. Recall that the modal expansions discussed in the previous sections are all local as they are defined on a single (master) element. The modal basis functions can be decomposed into boundary

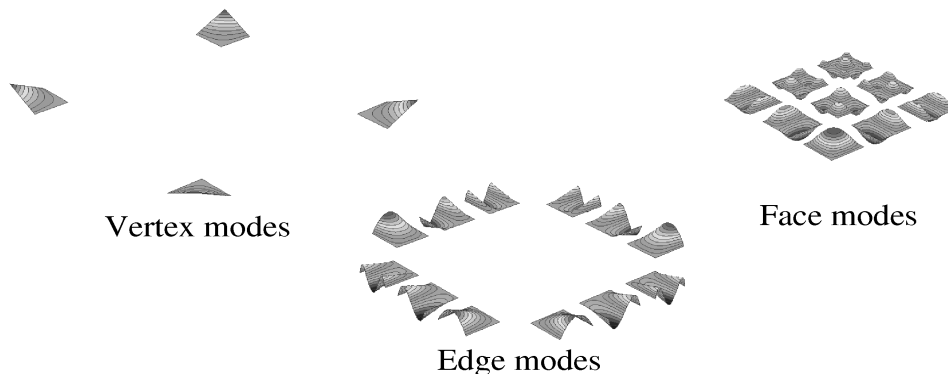


Figure 3.6: Boundary/interior decomposition of the modal expansion for $P = 4$, as shown in Figure 3.3. (From Karniadakis and Sherwin [38], reproduced with permission.)

and interior modes. Boundary modes are defined as all the modes that have nonzero support on the element's boundary, while interior modes are all the modes that are zero on all boundaries. Figure 3.6 shows an example of such decomposition for a two-dimensional basis, where the vertex modes and edge modes constitute the boundary modes, while the face modes constitute the interior modes. In order to construct a solution approximation which is \mathcal{C}_0 -continuous within the entire space-time slab, the elemental expansions need to be coupled between neighboring elements. Note that this requires only the boundary modes to be coupled, since only these have non-zero magnitude along the element boundaries. An example of this is shown in Figure 3.7, where \mathcal{C}_0 -continuity on a two-dimensional domain is obtained by matching the vertex modes and edge modes of similar shape.

In implementations of finite-element methods operations such as integration and differentiation are typically performed in a local fashion within each element. Consequently, to obtain the global system of equations a mapping is needed which assembles the global system from the local systems of all elements. In Appendix C a detailed description is given how this procedure is handled in the present implementation.

In the remainder of the thesis, we typically write the global finite element approx-

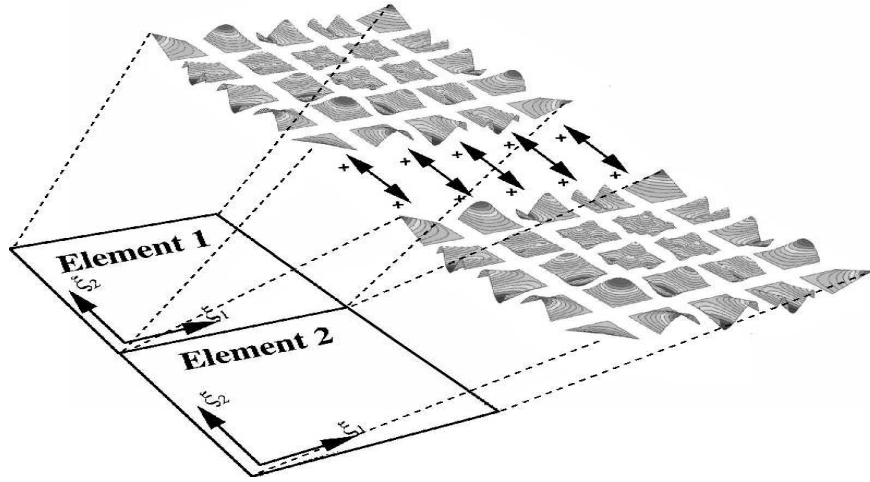


Figure 3.7: Illustration of the construction of a global \mathcal{C}^0 polynomial space from two local modal expansions of degree $P_1 = P_2 = 4$. The \mathcal{C}^0 continuity condition can be ensured by matching the vertex and boundary modes of similar shape. (From Karniadakis and Sherwin [38], reproduced with permission.)

imations using upper case indices, for example

$$u^h = \sum_{A=1}^{n_{bf}} N_A y_A, \quad (3.29)$$

where N_A represents the A^{th} global expansion mode (e.g. obtained by coupling matching boundary modes), y_A is its corresponding global solution approximation coefficient, and n_{bf} is the number of global basis functions.

3.4 Application to convection-diffusion

In this section, we present an evaluation of the different finite-element discretizations described in the previous sections using the linear convection-diffusion equation. The latter represents a simple model problem for the fluid flow equations. To gain insight

into relative accuracy and computational efficiency in terms of degrees of freedom, we investigate both temporal and spatial convergence behavior.

The linear time-dependent convection-diffusion equation is defined as

$$\mathcal{L}(u) = u_{,t} + \lambda_i u_{,i} - \nu u_{,ii} = f \quad \text{in } \Omega, \quad (3.30)$$

where u is the dependent variable representing a scalar quantity, λ_i is the constant convection velocity, $\nu (> 0)$ is the constant diffusion coefficient, and f is a prescribed source term. Equation (3.30) must be solved subject to appropriate initial and boundary conditions, as described in subsequent sections.

In the following sections, we investigate the convergence behavior of the numerical solutions u^h . For this purpose we consider the L_2 -norm of the error normalized by the exact solution, defined as

$$E = \left[\frac{\int_{\Omega} (u(x, T) - u^h(x, T))^2 \, d\Omega}{\int_{\Omega} u(x, T)^2 \, d\Omega} \right]^{1/2}, \quad (3.31)$$

where T the final time level of the computation. For consistent methods, the discretization error E can be written as

$$E = Ch^p + h.o.t., \quad (3.32)$$

where h represents the mesh spacing, C is a constant independent of h and p is the order of accuracy of the discretization. The latter can be estimated using systematic mesh refinement, as described in the following. Let E_1 denote the error obtained using mesh size h , and E_2 the error obtained using mesh size h/r , then

$$E_1 \approx Ch^p, \quad (3.33)$$

$$E_2 \approx C(h/r)^p, \quad (3.34)$$

where r is the mesh refinement ratio. The observed order of accuracy of the numerical method can then be calculated from

$$p = \log \left(\frac{E_1}{E_2} \right) / \log(r). \quad (3.35)$$

Next, we describe the discrete forms of the convection-diffusion equation (3.30) corresponding to the different time-discretization methods considered.

3.4.1 Time-continuous Galerkin

The finite-element approximation and weighting space for the TCG method are defined as

$$\mathcal{V}_n^h = \{u^h | u^h \in \mathcal{C}^0(\mathbf{Q}_n), u^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n)), u^h = g(t) \text{ on } \Gamma_D\}, \quad (3.36)$$

$$\mathcal{W}_n^h = \{w^h | w^h \in \mathcal{C}^0(\mathbf{Q}_n), w^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n)), w^h = 0 \text{ on } \Gamma_D\}, \quad (3.37)$$

where Γ_D is the portion of the space-time slab boundary where Dirichlet conditions $g(t)$ are specified. The variational form of the convection-diffusion equation (3.30) can then be stated as follows: *For every slab \mathbf{Q}_n find $u^h \in \mathcal{V}_n^h$ such that $\forall w^h \in \mathcal{W}_n^h$*

$$\begin{aligned} \int_{\mathbf{Q}_n} (-w_t^h u^h - \lambda_i w_{,i}^h u^h + \nu w_{,i}^h u_{,i}^h) d\mathbf{Q} + \int_{\mathbf{P}_n} (w^h \lambda_i u^h - w^h \nu u_{,i}^h) n_i d\mathbf{P} \\ + \int_{\Omega} w^h(t_{n+1}) u^h(t_{n+1}) d\Omega - \int_{\mathbf{Q}_n} w^h f d\mathbf{Q} = 0, \end{aligned} \quad (3.38)$$

where we have performed integration-by-parts in both space and time, and n_i denotes the spatial unit outward normal. For this method, we set the weighting functions to zero at times t_n , i.e. $w^h(t_n) = 0$, which guarantees an equal number of equations and unknowns.

The global solution and weighting function within the n^{th} space-time slab for the TCG method can be written as

$$u^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) u_{A;(n)} + \pi_n^+(t) u_{A;(n+1)}), \quad (3.39)$$

$$w^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) w_{A;(n)} + \pi_n^+(t) w_{A;(n+1)}), \quad (3.40)$$

where $N_A(\mathbf{x})$ are the A^{th} global spatial mode, $u_{A;(n)}$ and $u_{A;(n+1)}$ are its corresponding solution approximation coefficients at times t_n and t_{n+1} , likewise, $w_{A;(n)}$ and $w_{A;(n+1)}$

are the corresponding weighting coefficients, n_{bf} is the total number of global spatial basis functions, and $\pi_n^\pm(t)$ are defined in (3.9). A discussion on global ordering, indicated here by upper case indices, is provided in Appendix C. Substituting (3.39) and (3.40) with $w_{A;(n)} = 0$ into (3.38) yields

$$\begin{aligned}
& \sum_{A=1}^{n_{bf}} w_{A;(n+1)} \left\{ \int_{\mathcal{Q}_n} \left(-N_A \pi_{n,t}^+ \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) - \right. \right. \\
& \quad \lambda_i N_{A,i} \pi_n^+ \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) + \\
& \quad \left. \left. \nu N_{A,i} \pi_n^+ \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) d\mathcal{Q} \right. \\
& + \int_{\mathcal{P}_n} \left(N_A \pi_n^+ \lambda_i \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) + \right. \\
& \quad \left. N_A \pi_n^+ \nu \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) n_i d\mathcal{P} \\
& \left. + \int_{\Omega} N_A \sum_{B=1}^{n_{bf}} N_B u_{B;(n+1)} d\Omega - \int_{\mathcal{Q}_n} N_A \pi_n^+ f d\mathcal{Q} = 0 \right\}. \quad (3.41)
\end{aligned}$$

This equation can be written as

$$\sum_{A=1}^{n_{bf}} w_{A;(n+1)} G_A(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}), \quad (3.42)$$

where

$$\mathbf{u}_{(n+1)} = \{u_{1;(n+1)}, \dots, u_{n_{bf};(n+1)}\}^T, \quad (3.43)$$

$$\mathbf{u}_{(n)} = \{u_{1;(n)}, \dots, u_{n_{bf};(n)}\}^T, \quad (3.44)$$

and G_A are linear functions of the unknowns $\mathbf{u}_{(n+1)}$ and given $\mathbf{u}_{(n)}$. Since (3.42) must hold for all $w_{A;(n+1)}$, we obtain

$$G_A(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}) = 0 \quad \text{for } A = 1, \dots, n_{bf}. \quad (3.45)$$

These equations form a linear system of equations

$$\mathbf{G}(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}) \equiv \{G_1, G_2, \dots, G_{n_{bf}}\}^T = 0. \quad (3.46)$$

having n_{bf} equations for n_{bf} unknowns.

3.4.2 Time-continuous Petrov-Galerkin

The finite-element approximation and weighting space for the TCPG method are defined as

$$\mathcal{V}_n^h = \{u^h | u^h \in \mathcal{C}^0(\mathbf{Q}_n), u^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n)), u^h = g(t) \text{ on } \Gamma_D\} \quad (3.47)$$

$$\mathcal{W}_n^h = \{w^h | w^h \in \mathcal{C}^0(\mathbf{Q}_n), w^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_0(I_n))\}, \quad (3.48)$$

The variational form of the convection-diffusion equation (3.30) can then be stated as follows: *For every slab \mathbf{Q}_n find $u^h \in \mathcal{V}_n^h$ such that $\forall w^h \in \mathcal{W}_n^h$*

$$\begin{aligned} \int_{\mathbf{Q}_n} (-\lambda_i w_{,i}^h u^h + \nu w_{,i}^h u_{,i}^h) d\mathbf{Q} + \int_{\mathbf{P}_n} (w^h \lambda_i u^h - w^h \nu u_{,i}^h) n_i d\mathbf{P} \\ + \int_{\Omega} w^h(t_{n+1}) u^h(t_{n+1}) - w^h(t_n) u^h(t_n) d\Omega - \int_{\mathbf{Q}_n} w^h f d\mathbf{Q} = 0, \end{aligned} \quad (3.49)$$

where we have performed integration-by-parts in both space and time, and n_i denotes the spatial unit outward normal. Note that constraining the solution at time t_n , guarantees an equivalent number of equations and unknowns.

The global solution and weighting function within the n^{th} space-time slab for the TCPG method can be written as

$$u^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) u_{A;(n)} + \pi_n^+(t) u_{A;(n+1)}), \quad (3.50)$$

$$w^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) w_{A;(n+1)}, \quad (3.51)$$

Substituting (3.50) and (3.51) into (3.49) yields

$$\begin{aligned}
& \sum_{A=1}^{n_{bf}} w_{A;(n+1)} \left\{ \int_{\mathcal{Q}_n} \left(-\lambda_i N_{A,i} \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) + \right. \right. \\
& \quad \left. \left. \nu N_{A,i} \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) d\mathcal{Q} \right. \\
& + \int_{\mathcal{P}_n} \left(N_A \lambda_i \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) - \right. \\
& \quad \left. N_A \nu \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) n_i d\mathcal{P} \\
& + \int_{\Omega} \left(N_A \sum_{B=1}^{n_{bf}} N_B u_{B;(n+1)} - N_A \sum_{B=1}^{n_{bf}} N_B u_{B;(n)} \right) d\Omega \\
& \left. - \int_{\mathcal{Q}_n} N_A f d\mathcal{Q} = 0 \right\}. \tag{3.52}
\end{aligned}$$

With the definitions of the previous section, this equation can be written as

$$\sum_{A=1}^{n_{bf}} w_{A;(n+1)} G_A(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}), \tag{3.53}$$

where G_A are linear functions of the unknown $\mathbf{u}_{(n+1)}$ and given $\mathbf{u}_{(n)}$. Since (3.53) must hold for all $w_{A;(n+1)}$, we obtain the following linear system of equations

$$\mathbf{G}(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}) \equiv \{G_1, G_2, \dots, G_{n_{bf}}\}^T = 0, \tag{3.54}$$

which constitutes n_{bf} equations for n_{bf} unknowns.

3.4.3 Time-discontinuous Galerkin

The finite-element approximation and weighting space for the TDG method are defined as

$$\mathcal{V}_n^h = \{u^h | u^h \in \mathcal{C}^0(\mathcal{Q}_n), u^h|_{\mathcal{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n)), u^h = g(t) \text{ on } \Gamma_D\}, \tag{3.55}$$

$$\mathcal{W}_n^h = \{w^h | w^h \in \mathcal{C}^0(\mathcal{Q}_n), w^h|_{\mathcal{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n)), w^h = 0 \text{ on } \Gamma_D\}, \tag{3.56}$$

The variational form of the convection-diffusion equation (3.30) can be stated as follows: *For every slab Q_n find $u^h \in \mathcal{V}_n^h$ such that $\forall w^h \in \mathcal{W}_n^h$*

$$\begin{aligned} \int_{Q_n} (-w_{,t}^h u^h - \lambda_i w_{,i}^h u^h + \nu w_{,i}^h u_{,i}^h) dQ + \int_{P_n} (w^h \lambda_i u^h - w^h \nu u_{,i}^h) n_i dP \\ + \int_{\Omega} w^h(t_{n+1}^-) u^h(t_{n+1}^-) - w^h(t_n^+) u^h(t_n^-) d\Omega \\ - \int_{Q_n} w^h f dQ = 0, \end{aligned} \quad (3.57)$$

where the second line is obtained from adding the jump condition

$$w^h(t_n)(u^h(t_n^+) - u^h(t_n^-)) \quad (3.58)$$

to the terms resulting from integration-by-parts of the time term.

The global solution and weighting function within the n^{th} space-time slab for the TDG method can be written as

$$u^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) u_{A;(n)} + \pi_n^+(t) u_{A;(n+1)}), \quad (3.59)$$

$$w^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) w_{A;(n)} + \pi_n^+(t) w_{A;(n+1)}), \quad (3.60)$$

Let

$$\mathbf{w}_A = \{w_{A;(n)}, w_{A;(n+1)}\}^T \quad \text{and} \quad \bar{\pi}_n = \{\pi_n^-(t), \pi_n^+(t)\}^T, \quad (3.61)$$

then the discrete variational form, obtained by substituting (3.59),(3.60) into (3.57)

can be written as

$$\begin{aligned}
& \sum_{A=1}^{n_{bf}} \mathbf{w}_A^T \left\{ \int_{\mathbf{Q}_n} \left(-N_A \bar{\pi}_{n,t} \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) - \right. \right. \\
& \quad \lambda_i N_{A,i} \bar{\pi}_n \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n u_{B;(n+1)}) + \\
& \quad \left. \left. \nu N_{A,i} \bar{\pi}_n \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) d\mathbf{Q} \right. \\
& + \int_{\mathbf{P}_n} \left(N_A \bar{\pi}_n \lambda_i \sum_{B=1}^{n_{bf}} N_B (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) - \right. \\
& \quad \left. N_A \bar{\pi}_n \nu \sum_{B=1}^{n_{bf}} N_{B,i} (\pi_n^- u_{B;(n)} + \pi_n^+ u_{B;(n+1)}) \right) n_i d\mathbf{P} \\
& + \int_{\Omega} \left(N_A \sum_{B=1}^{n_{bf}} N_B u_{B;(n+1)} - N_A \sum_{B=1}^{(n_{bf})(n-1)} N_B^{(n-1)} u_{B;(n)}^- \right) d\Omega \\
& \left. - \int_{\mathbf{Q}_n} N_A \bar{\pi}_n f d\mathbf{Q} = 0 \right\}. \tag{3.62}
\end{aligned}$$

where $N_B^{(n-1)}$ is the B^{th} global spatial basis function employed within the slab \mathbf{Q}_{n-1} , $u_{B;(n)}^-$ is its corresponding solution coefficient at time t_n^- , and $(n_{bf})(n-1)$ is the number of global spatial basis functions within slab \mathbf{Q}_{n-1} . Equation (3.62) can be written as

$$\sum_{A=1}^{n_{bf}} \mathbf{w}_A^T \mathbf{G}_A \left(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}, \mathbf{u}_{(n)}^- \right), \tag{3.63}$$

where

$$\mathbf{u}_{(n)}^- = \left\{ u_{1;(n)}^-, \dots, u_{(n_{bf})(n-1);(n)}^- \right\}^T, \tag{3.64}$$

and \mathbf{G}_A are 2×1 vectors whose components are linear functions of the unknowns $\mathbf{u}_{(n+1)}$ and $\mathbf{u}_{(n)}$, and given $\mathbf{u}_{(n)}^-$. Since (3.63) must hold for all \mathbf{w}_A , it follows

$$\mathbf{G}_A(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}, \mathbf{u}_{(n)}^-) = 0 \quad \text{for } A = 1, \dots, n_{bf}. \tag{3.65}$$

In the absence of constraints, these equations form a linear system of equations

$$\mathbf{G}(\mathbf{u}_{(n+1)}; \mathbf{u}_{(n)}, \mathbf{u}_{(n)}^-) \equiv \{\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n_{bf}}\}^T = 0. \tag{3.66}$$

constituting $2 \cdot n_{bf}$ equations for $2 \cdot n_{bf}$ unknowns.

3.4.4 Temporal convergence

In this section we investigate the temporal convergence behavior of the different time-discretization schemes discussed in the previous sections. To this end, we consider the one-dimensional form of (3.30) defined on $\Omega = (0, 1)$, with the diffusion term set to zero. The initial condition $u_0(x)$ is given by the Gaussian distribution

$$u_0(x) = e^{-\sigma(x-x_0)^2}, \quad (3.67)$$

where $\sigma(> 0)$ represents a measure for the extent of the distribution and x_0 is the center of the distribution. For the present calculations we set $\sigma = 400.0$ and $x_0 = 0.25$. The analytic solution for this problem is a simple translation of the initial condition, given by

$$u(x) = u_0(x - \lambda t) = e^{-\sigma(x-x_0-\lambda t)^2}. \quad (3.68)$$

Since we are interested in time-accurate solutions, the computations are carried out at constant Courant number close to unity, i.e. $\lambda \Delta t / \Delta x \approx 1$. Therefore, as the time-step is reduced, the spatial mesh is refined correspondingly. To isolate temporal convergence behavior as much as possible, effects of spatial discretization errors must be minimized. Therefore, the spatial mesh size was chosen to be four times smaller than the time-step size when using linear ($P = 1$) approximation functions in space. To obtain a unit Courant number, the convection velocity is set equal to $\lambda = 0.25$. Tables 3.1 – 3.3 show the computed L_2 -errors as well as the observed order of accuracy for the different time-discretization methods. The observed orders of accuracy are in good agreement with their theoretical values.

The upper graph of Figure 3.8 shows the L_2 -error of the TCG, TCPG and TDG methods versus the time-step size. For the range of time-step sizes considered, the TDG method is approximately at least three orders of magnitude more accurate

Δt	dofs	E	order
1.6e-2	256	1.439e-02	-
8.0e-3	512	7.300e-03	0.98
4.0e-3	1024	3.674e-03	0.99
2.0e-3	2048	1.842e-03	1.00
1.0e-3	4096	9.225e-04	1.00
5.0e-4	8192	4.616e-04	1.00

Table 3.1: Temporal convergence parameters for the TCG method

Δt	dofs	E	order
1.6e-2	256	2.576e-03	-
8.0e-3	512	6.566e-04	1.98
4.0e-3	1024	1.650e-04	1.99
2.0e-3	2048	4.129e-05	2.00
1.0e-3	4096	1.033e-05	2.00
5.0e-4	8192	2.582e-06	2.00

Table 3.2: Temporal convergence parameters for the TCPG method

than the TCPG method. In turn, the TCPG method is approximately one order of magnitude more accurate than the TCG method. Moreover, the curves do not cross over the range of time-step sizes, indicating that there is clear advantage of using a higher-order time discretization. The lower graph of Figure 3.8 shows the error versus degrees of freedom. The TDG method is clearly most accurate in terms of degrees of freedom.

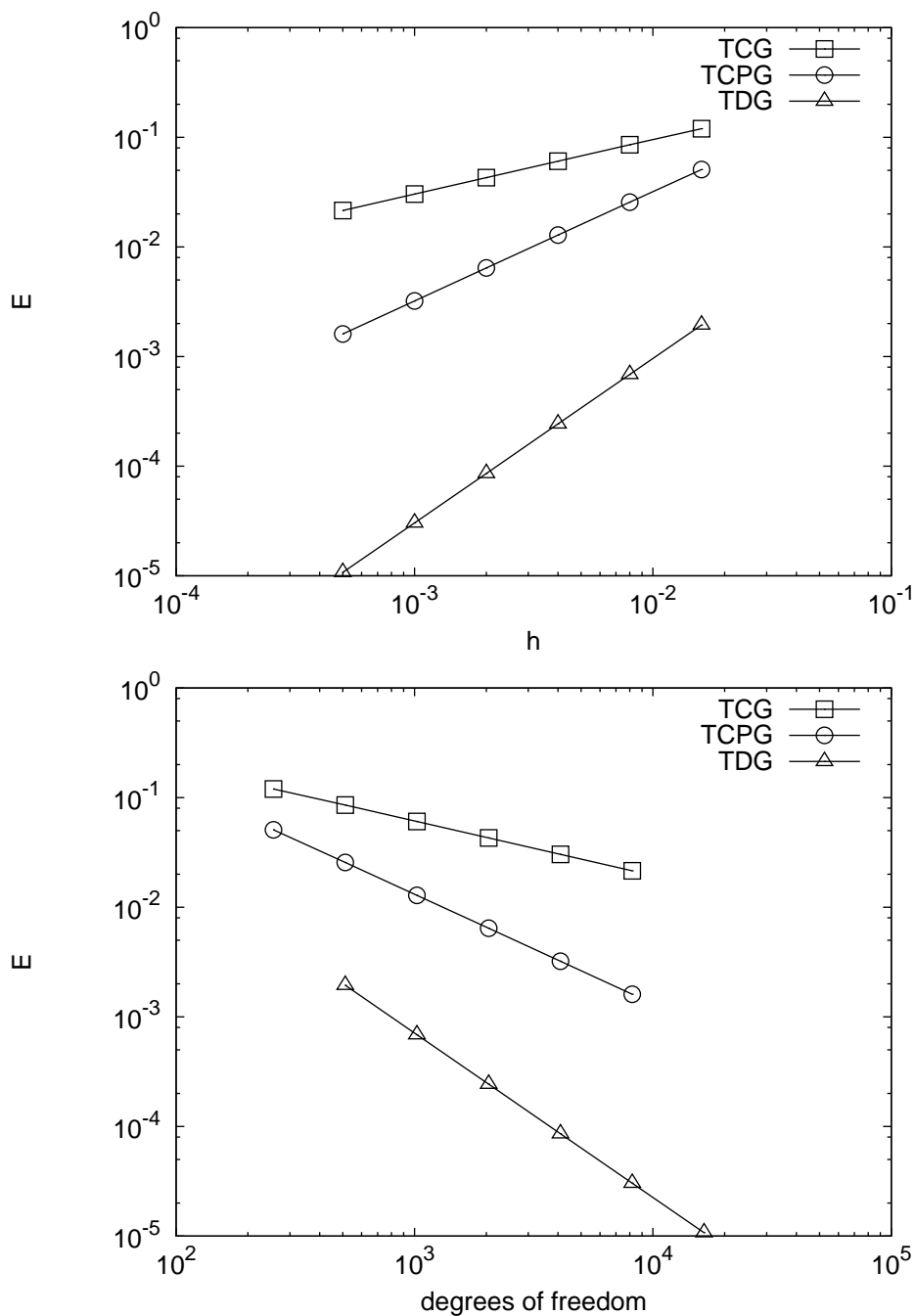


Figure 3.8: Temporal convergence using a unit Courant number.

Δt	dofs	E	order
1.6e-2	512	3.775e-06	-
8.0e-3	1024	4.717e-07	3.00
4.0e-3	2048	5.896e-08	3.00
2.0e-3	4096	7.370e-09	3.00
1.0e-3	8192	9.212e-10	3.00
5.0e-4	16384	1.152e-10	3.00

Table 3.3: Temporal convergence parameters for the TDG method

3.4.5 Spatial convergence

Here, we investigate the convergence behavior of the spatial discretization. We consider different spatial polynomial degrees for both the full-tensor basis and the serendipity basis for one, two and three-dimensional versions of the convection-diffusion equation.

Since we are interested in the convergence behavior of the spatial discretization, we consider a steady solution of the convection-diffusion equation. Such solution can be easily generated using the *Method of Manufactured Solutions* (MMS), as described by Roache [65]. The basic idea of the MMS is to specify an exact solution and force the differential equation to have this solution using a source term. The source term that produces any desired solution $u^e(x_i, t)$ of (3.30) is obtained by operating on $u^e(x_i, t)$ with the differential operator

$$f(x_i, t) = \mathcal{L}(u^e(x_i, t)) = u_{,t}^e + \lambda_i u_{,i}^e - \nu u_{,ii}^e. \quad (3.69)$$

If this source term is included in the differential equation as follows

$$\mathcal{L}(u) = f \equiv \mathcal{L}(u^e), \quad (3.70)$$

then it is clear that the solution to this equation must be $u^e(x_i, t)$.

The MMS technique is a useful tool in the verification of numerical codes. For this purpose arbitrary complex manufactured solutions can be chosen to make sure all terms in the code are excited and implemented properly. Here, we chose the following manufactured solutions

$$u^e(x_i) = \prod_i^d \sin\left(\frac{2\pi}{L_i}x_i\right), \quad (3.71)$$

where L_i is the length of the spatial domain in the i^{th} direction. Note that this corresponds to a periodic solution in each direction, so that boundary effects are excluded.

Steady solutions are obtained by using any one of the time-discretization schemes discussed in Section 3.2. A solution is considered steady when the difference in the L_2 -error norm (E) of two consecutive time steps is smaller than a specified tolerance, that is $(E_{n+1} - E_n) < 10^{-12}$.

Table 3.4 shows the error and observed order of accuracy for spatial polynomial degree $P = 1, \dots, 4$ obtained from the one-dimensional model problem on $\Omega = (0, 1)$ with $\lambda = 1.0$ and $\nu = 0.0003125$. It can be seen that the observed orders of accuracy are in good agreement with their theoretical values, $P + 1$. In Figure 3.9 the error is plotted versus the mesh size and degrees of freedom. Note that the curves do not cross, indicating the advantage of using a higher-order approximation basis.

In Table 3.5 the results are shown for spatial polynomial degrees $P = 1, \dots, 4$ obtained from the two-dimensional model problem defined on $\Omega = (0, 1)^2$ with $\lambda_i = 0.7$ and $\nu = 0.0003125$. It can be seen that all the observed orders of accuracy are close to the theoretical values. The upper graph of Figure 3.10 shows the error versus the spatial mesh-size. Except for $P = 2$, the results obtained using the serendipity basis are less accurate than their full-basis counterparts for a given mesh size. Although some modes are not strictly necessary to retain a given order of accuracy, these modes do contribute to the absolute accuracy of the discretization. Note that again the curves do not cross. The lower graph of Figure 3.10 shows the error versus degrees of

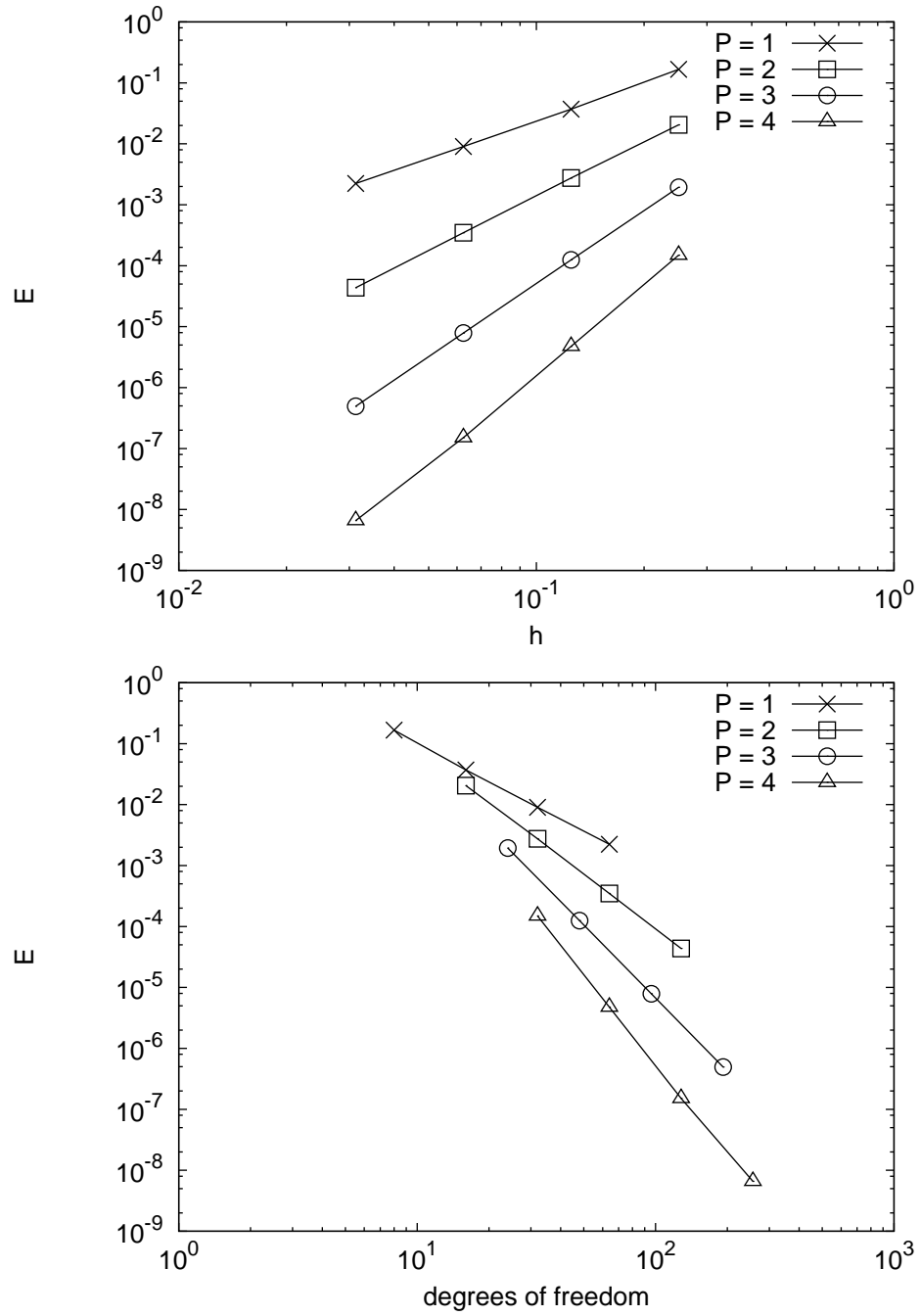


Figure 3.9: Spatial convergence for one-dimensional convection-diffusion using different polynomial degrees.

mesh	P	E	order	mesh	P	E	order
4	1	1.67e-01	-	4	2	2.05e-02	-
8	1	3.70e-02	2.17	8	2	2.74e-03	2.90
16	1	9.01e-03	2.04	16	2	3.47e-04	2.98
32	1	2.24e-03	2.01	32	2	4.35e-05	3.00
mesh	P	E	order	mesh	P	E	order
4	3	1.94e-03	-	4	4	1.49e-04	-
8	3	1.25e-04	3.96	8	4	4.74e-06	4.96
16	3	7.87e-06	3.99	16	4	1.51e-07	4.98
32	3	4.95e-07	3.99	32	4	6.59e-09	4.51

Table 3.4: Observed spatial order of accuracy for 1D convection-diffusion.

freedom. For $P = 2$, the serendipity basis is slightly more efficient in terms of degrees of freedom than the full basis. However, for $P = 3$ both methods are very similar, while for $P = 4$ the full basis appears to be more efficient.

The results for spatial polynomial degrees $P = 1, 2, 3$ of the three-dimensional model problem defined on $\Omega = (0, 1)^3$ with $\lambda_i = 0.7$ and $\nu = 0.0003125$ are shown in Table 3.6. Again all observed orders of accuracy are in good agreement with the theoretical values. Figure 3.11 shows behavior very similar to that of the two-dimensional case shown in Figure 3.10. As the spatial polynomial degree is increased, the serendipity expansion becomes less efficient in terms of degrees of freedom than to the full basis.

mesh	P	E	order
4×4	1	2.25e-01	-
8×8	1	5.33e-02	2.08
16×16	1	1.32e-02	2.02
32×32	1	3.28e-03	2.00

mesh	P	E	order	mesh	P	E	order
4×4	2	2.05e-02	-	4×4	2*	3.14e-02	-
8×8	2	2.74e-03	2.90	8×8	2*	3.92e-03	3.00
16×16	2	3.47e-04	2.98	16×16	2*	4.92e-04	3.00
32×32	2	4.35e-05	3.00	32×32	2*	6.16e-05	3.00

mesh	P	E	order	mesh	P	E	order
4×4	3	2.70e-03	-	4×4	3*	1.54e-02	-
8×8	3	1.76e-04	3.94	8×8	3*	6.40e-04	4.59
16×16	3	1.11e-05	3.98	16×16	3*	3.61e-05	4.15
32×32	3	6.98e-07	3.99	32×32	3*	2.20e-06	4.04

Table 3.5: Observed spatial order of accuracy for 2D convection-diffusion, the asterisks on the order indicate a serendipity expansion.

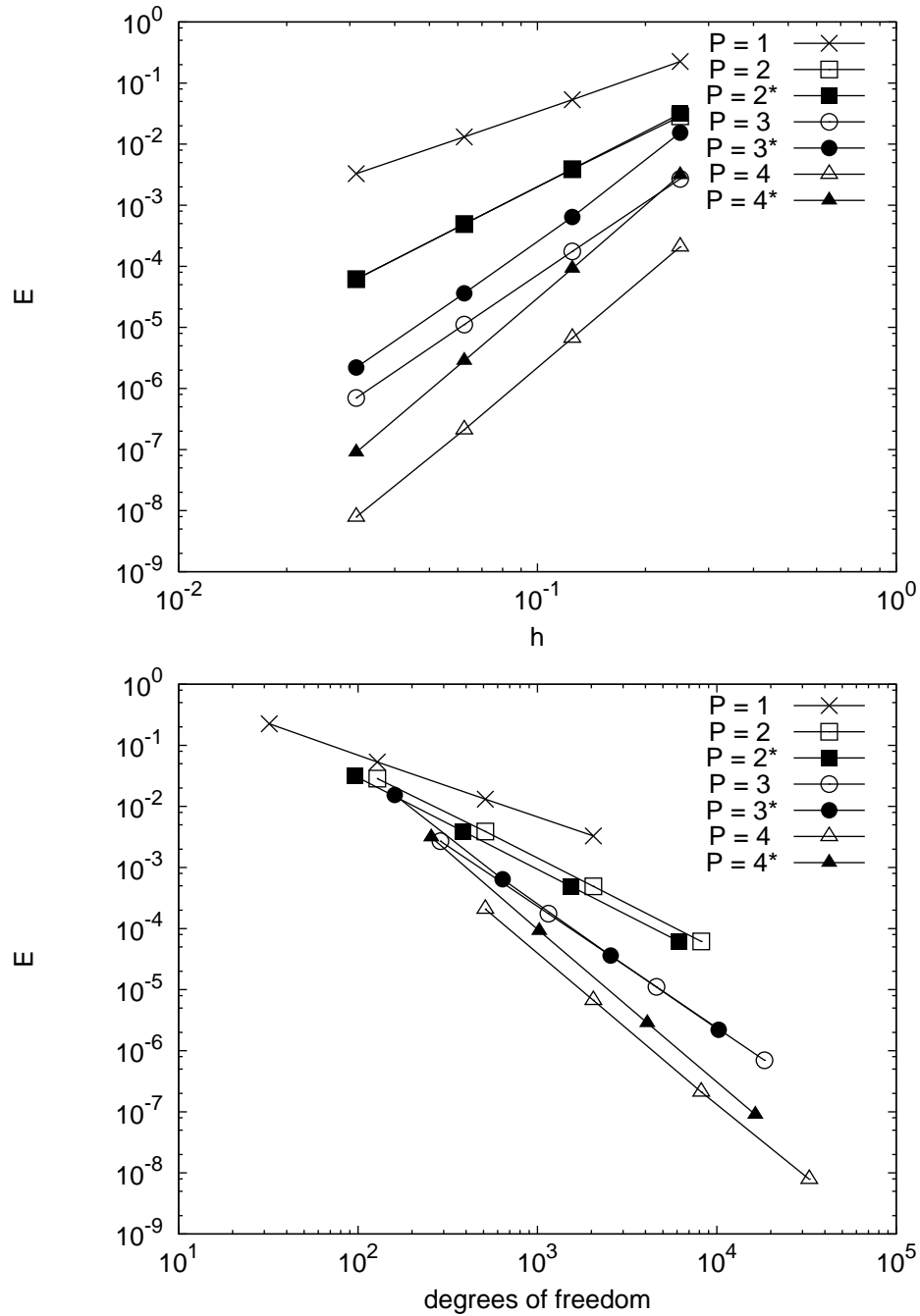


Figure 3.10: Spatial convergence for two-dimensional convection-diffusion using different polynomial degrees. The asterisks indicate a serendipity expansion.

mesh	P	E	order
$4 \times 4 \times 4$	1	2.56e-01	-
$8 \times 8 \times 8$	1	5.92e-02	2.11
$16 \times 16 \times 16$	1	1.45e-02	2.03

mesh	P	E	order	mesh	P	E	order
$4 \times 4 \times 4$	2	3.40e-02	-	$4 \times 4 \times 4$	2*	4.11e-02	-
$8 \times 8 \times 8$	2	4.73e-03	2.84	$8 \times 8 \times 8$	2*	4.82e-03	3.09
$16 \times 16 \times 16$	2	6.01e-04	2.98	$16 \times 16 \times 16$	2*	6.04e-04	3.00

mesh	P	E	order	mesh	P	E	order
$4 \times 4 \times 4$	3	3.25e-03	-	$4 \times 4 \times 4$	3*	2.66e-02	-
$8 \times 8 \times 8$	3	2.14e-04	3.93	$8 \times 8 \times 8$	3*	1.09e-03	4.61
$16 \times 16 \times 16$	3	1.36e-05	3.97	$16 \times 16 \times 16$	3*	6.10e-05	4.16

Table 3.6: Observed spatial order of accuracy for 3D convection-diffusion, the asterisks on the order indicate a serendipity expansion.

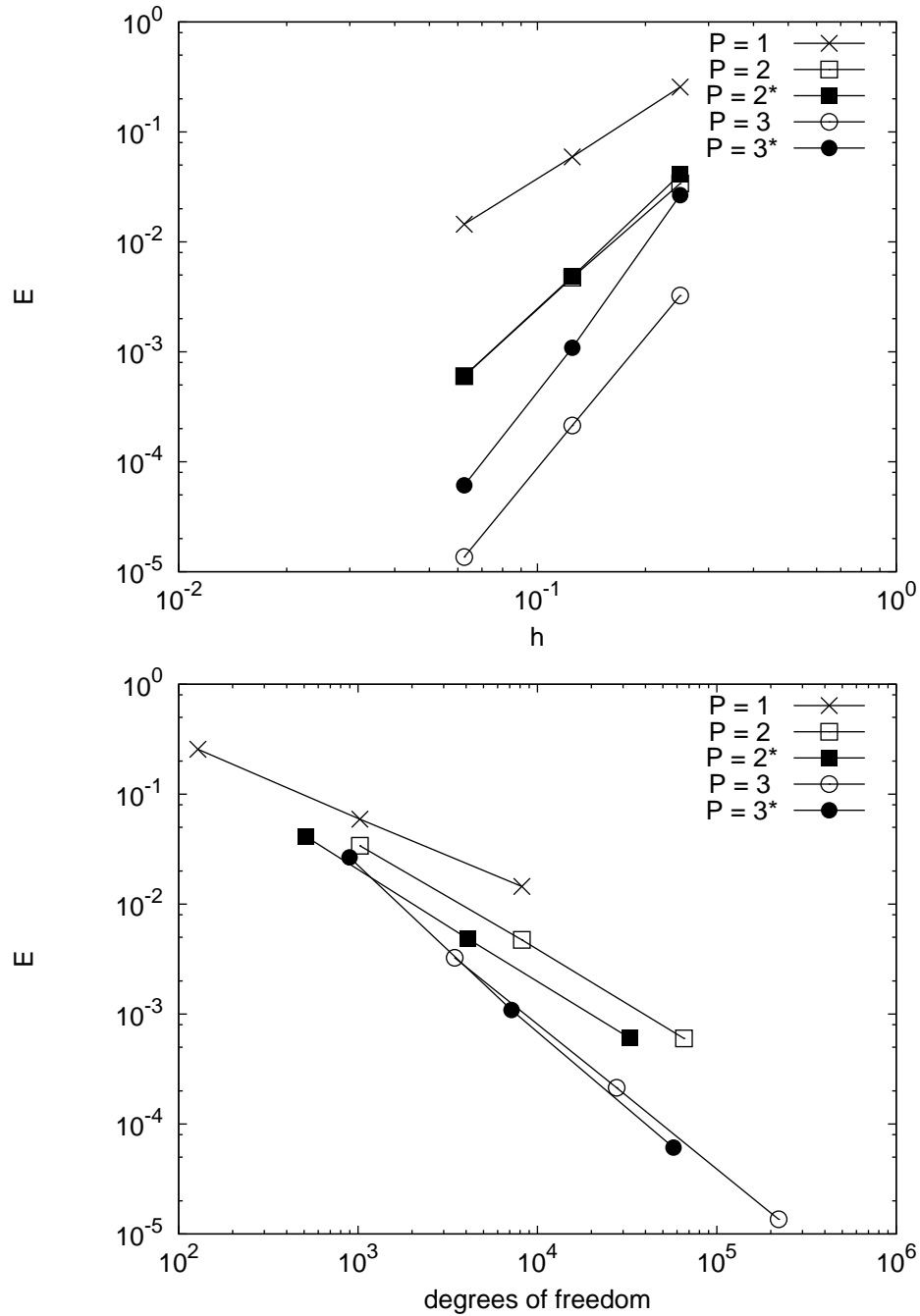


Figure 3.11: Spatial convergence for three-dimensional convection-diffusion using different polynomial orders. The asterisks indicate a serendipity expansion.

Chapter 4

Numerical solution of the Navier-Stokes equations

In this chapter, we describe the solution techniques that are used to solve the Navier-Stokes equations numerically. In Sections 4.1, 4.2 and 4.3 we present the variational formulations and the resulting non-linear systems of equations for the time-continuous Galerkin method, the time-continuous Petrov-Galerkin method and time-discontinuous Galerkin method. Next, in Section 4.4 we describe the predictor multi-corrector algorithm which is used to solve the non-linear systems. Then, in Section 4.5 we discuss the assembly procedure for the linear systems of equations that arise in the predictor multi-corrector algorithm. Subsequently, in Section 4.6 we give a brief overview of the iterative technique that is used to solve the linear systems. In Section 4.7 we conclude with an application to laminar channel flow.

4.1 Time-continuous Galerkin

4.1.1 Space-time variational form

The finite-element approximation and weighting spaces for the TCG method are defined as

$$\mathcal{Y}_n^h = \{ \mathbf{Y}^h | \mathbf{Y}^h \in (\mathcal{C}^0(\mathcal{Q}_n))^m, \mathbf{Y}^h|_{\mathcal{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n))^m, \mathbf{Y}^h = \mathbf{g}(t) \text{ on } \Gamma_D \}, \quad (4.1)$$

and

$$\mathcal{W}_n^h = \{ \mathbf{W}^h | \mathbf{W}^h \in (\mathcal{C}^0(\mathcal{Q}_n))^m, \mathbf{W}^h|_{\mathcal{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n))^m, \mathbf{W}^h = \mathbf{0} \text{ on } \Gamma_D \}, \quad (4.2)$$

where $m = d + 2$ is the number of conservative variables, \mathbf{g} is the vector of Dirichlet conditions and Γ_D is the portion of the space-time slab boundary where Dirichlet boundary conditions are specified. The variational formulation of the compressible Navier-Stokes equations (2.1a) can then be stated as follows: *Within each slab \mathcal{Q}_n , find $\mathbf{Y}^h \in \mathcal{Y}_n^h$ such that $\forall \mathbf{W}^h \in \mathcal{W}_n^h$*

$$\begin{aligned} & \int_{\mathcal{Q}_n} \left(-\mathbf{W}_{,t}^h \cdot \mathbf{U}(\mathbf{Y}^h) - \mathbf{W}_{,i}^h \cdot \mathbf{F}_i(\mathbf{Y}^h) + \mathbf{W}_{,i}^h \cdot \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^h \right) d\mathcal{Q} \\ & + \int_{\mathcal{P}_n} \mathbf{W}^h \cdot (\mathbf{F}_i(\mathbf{Y}^h) - \mathbf{F}_i^v(\mathbf{Y}^h)) n_i d\mathcal{P} \\ & + \int_{\Omega} \mathbf{W}^h(t_{n+1}) \cdot \mathbf{U}(\mathbf{Y}^h(t_{n+1})) d\Omega \\ & = \int_{\mathcal{Q}_n} \mathbf{W}^h \cdot \mathbf{S} d\mathcal{Q}, \end{aligned} \quad (4.3)$$

where the both the spatial and time flux terms are in integration-by-parts form, which results in conservation of fluxes under inexact quadrature rules [71], whereas the non-integrated-by-parts form results in a loss of conservation.

The initial condition is strongly enforced by constraining the finite-element solution at time t_n of the current slab \mathbf{Q}_n to the solution at time t_n of previous slab \mathbf{Q}_{n-1} . Consequently, in order to obtain a square system of equations, the weighting functions must be zero at time t_n , i.e. $\mathbf{W}^h(t_n) = \mathbf{0}$.

4.1.2 Non-linear system of equations

The solution approximation and weighting functions for the time-continuous Galerkin method within the n^{th} space-time slab can be written as

$$\mathbf{Y}^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) \mathbf{y}_{A;(n)} + \pi_n^+(t) \mathbf{y}_{A;(n+1)}), \quad (4.4)$$

$$\mathbf{W}^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) \mathbf{w}_{A;(n)} + \pi_n^+(t) \mathbf{w}_{A;(n+1)}), \quad (4.5)$$

for $\mathbf{x} \in \Omega$, $t \in I_n$. In these relations, $N_A(\mathbf{x})$ is the A^{th} global spatial basis function, $\mathbf{y}_{A;(n)}$ and $\mathbf{y}_{A;(n+1)}$ are the corresponding $m \times 1$ vectors of solution approximation coefficients at times t_n and t_{n+1} , and $\mathbf{w}_{A;(n)}$ and $\mathbf{w}_{A;(n+1)}$ are the $m \times 1$ vectors of weighting function coefficients, n_{bf} is the total number of global spatial approximation functions, and $\pi_n^\pm(t)$ are the temporal basis functions defined in (3.9). Note that we use upper case indices to indicate global ordering, see Appendix C for the global ordering definition.

Substituting (4.4) and (4.5) with $\mathbf{w}_{A;(n)} = \mathbf{0}$ into the space-time variational statement (4.3) yields

$$\begin{aligned}
& \sum_{A=1}^{n_{bf}} \mathbf{w}_{A;(n+1)}^T \left\{ \int_{\mathcal{Q}_n} \left(-N_A \pi_{n,t}^+ \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) - \right. \right. \\
& \quad N_{A,i} \pi_n^+ \mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) + \\
& \quad \left. \left. N_{A,i} \pi_n^+ \tilde{\mathbf{K}}_{ij} \sum_{B=1}^{n_{bf}} N_{B,j} (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+(t) \mathbf{y}_{B;(n+1)}) - N_A \pi_n^+ \mathbf{S} \right) d\mathcal{Q} \right. \\
& + \int_{\mathcal{P}_n} N_A \pi_n^+ \left(\mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) - \right. \\
& \quad \left. \mathbf{F}_i^v \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) \right) n_i d\mathcal{P} \\
& \left. + \int_{\Omega} N_A \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B \mathbf{y}_{B;(n+1)} \right) d\Omega \right\} = \mathbf{0}, \tag{4.6}
\end{aligned}$$

Define

$$\mathbf{y}_{(n+1)} = \{ \mathbf{y}_{1;(n+1)}^T, \mathbf{y}_{2;(n+1)}^T, \dots, \mathbf{y}_{n_{bf};(n+1)}^T \}^T, \tag{4.7}$$

$$\mathbf{y}_{(n)} = \{ \mathbf{y}_{1;(n)}^T, \mathbf{y}_{2;(n)}^T, \dots, \mathbf{y}_{n_{bf};(n)}^T \}^T, \tag{4.8}$$

$$\mathbf{w} = \{ \mathbf{w}_{1;(n+1)}^T, \dots, \mathbf{w}_{n_{bf};(n+1)}^T \}^T. \tag{4.9}$$

Then (4.6) can be written as

$$\mathbf{w}^T \mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}) = \mathbf{0}, \tag{4.10}$$

and since (4.10) must hold for all \mathbf{w} , we obtain

$$\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}) = \mathbf{0}, \tag{4.11}$$

where $\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)})$ is an $(n_{bf} \cdot m) \times 1$ system of non-linear algebraic equations, which must be solved at every time step.

4.2 Time-continuous Petrov-Galerkin

4.2.1 Space-time variational form

The finite element approximation and weighting spaces for the time-continuous Petrov-Galerkin method are defined as

$$\mathcal{Y}_n^h = \{ \mathbf{Y}^h | \mathbf{Y}^h \in (\mathcal{C}^0(\mathbf{Q}_n))^m, \mathbf{Y}^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n))^m, \mathbf{Y}^h = \mathbf{g}(t) \text{ on } \Gamma_D \}, \quad (4.12)$$

and

$$\mathcal{W}_n^h = \{ \mathbf{W}^h | \mathbf{W}^h \in (\mathcal{C}^0(\mathbf{Q}_n))^m, \mathbf{W}^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_0(I_n))^m \}, \quad (4.13)$$

The variational formulation of the compressible Navier-Stokes equations (2.1a) can then be stated as follows: *Within each slab \mathbf{Q}_n , find $\mathbf{Y}^h \in \mathcal{Y}_n^h$ such that $\forall \mathbf{W}^h \in \mathcal{W}_n^h$*

$$\begin{aligned} & \int_{\mathbf{Q}_n} \left(\mathbf{W}_{,i}^h \cdot \mathbf{F}_i(\mathbf{Y}^h) + \mathbf{W}_{,i}^h \cdot \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^h \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n} \mathbf{W}^h \cdot (\mathbf{F}_i(\mathbf{Y}^h) - \mathbf{F}_i^v(\mathbf{Y}^h)) n_i d\mathbf{P} \\ & + \int_{\Omega} (\mathbf{W}^h(t_{n+1}) \cdot \mathbf{U}(\mathbf{Y}^h(t_{n+1})) - \mathbf{W}^h(t_n) \cdot \mathbf{U}(\mathbf{Y}^h(t_n))) d\Omega \\ & = \int_{\mathbf{Q}_n} \mathbf{W}^h \cdot \mathbf{S} d\mathbf{Q}. \end{aligned} \quad (4.14)$$

Note that, in order to obtain an equal number of equations and unknowns, the solution must be constrained at time t_n . However, as opposed to the TCG method, the weighting functions are not zero at time t_n .

4.2.2 Non-linear system of equations

The solution approximation and weighting functions within the n^{th} space-time slab can be written as

$$\mathbf{Y}^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) (\pi_n^-(t) \mathbf{y}_{A;(n)} + \pi_n^+(t) \mathbf{y}_{A;(n+1)}), \quad (4.15)$$

$$\mathbf{W}^h(\mathbf{x}) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x}) \mathbf{w}_{A;(n+1)}, \quad (4.16)$$

Substituting (4.15) and (4.16) into the space-time variational statement (4.14) yields

$$\begin{aligned} & \sum_{A=1}^{n_{bf}} \mathbf{w}_{A;(n+1)}^T \left\{ \int_{\mathcal{Q}_n} \left(N_{A,i} \mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) + \right. \right. \\ & \quad \left. \left. N_{A,i} \tilde{\mathbf{K}}_{ij} \sum_{B=1}^{n_{bf}} N_{B,j} (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) - N_A \mathbf{S} \right) d\mathcal{Q} \right. \\ & + \int_{\mathcal{P}_n} N_A \left(\mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) - \right. \\ & \quad \left. \mathbf{F}_i^v \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) \right) n_i d\mathcal{P} \\ & \left. + \int_{\Omega} N_A \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B \mathbf{y}_{B;(n+1)} \right) - N_A \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B \mathbf{y}_{B;(n)} \right) d\Omega \right\} = \mathbf{0}, \quad (4.17) \end{aligned}$$

With (4.7) – (4.9), (4.17) can be written as

$$\mathbf{w}^T \mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}) = \mathbf{0}, \quad (4.18)$$

and since (4.18) must hold for all \mathbf{w} , we obtain

$$\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}) = \mathbf{0}, \quad (4.19)$$

where $\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)})$ is an $(n_{bf} \cdot m) \times 1$ system of non-linear algebraic equations, which has to be solved at each time step.

4.3 Time-discontinuous Galerkin

4.3.1 Space-time variational form

The finite element approximation and weighting spaces for the time-discontinuous Galerkin method are defined as

$$\mathcal{Y}_n^h = \{ \mathbf{Y}^h | \mathbf{Y}^h \in (\mathcal{C}^0(\mathbf{Q}_n))^m, \mathbf{Y}^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n))^m, \mathbf{Y}^h = \mathbf{g}(t) \text{ on } \Gamma_D \}, \quad (4.20)$$

and

$$\mathcal{W}_n^h = \{ \mathbf{W}^h | \mathbf{W}^h \in (\mathcal{C}^0(\mathbf{Q}_n))^m, \mathbf{W}^h|_{\mathbf{Q}_n^e} \in (\mathcal{P}_k(\Omega_n^e) \times \mathcal{P}_1(I_n))^m, \mathbf{W}^h = \mathbf{0} \text{ on } \Gamma_D \}, \quad (4.21)$$

The variational formulation of the compressible Navier-Stokes equations (2.1a) can then be stated as follows: *Within each slab \mathbf{Q}_n , find $\mathbf{Y}^h \in \mathcal{Y}_n^h$ such that $\forall \mathbf{W}^h \in \mathcal{W}_n^h$*

$$\begin{aligned} & \int_{\mathbf{Q}_n} \left(-\mathbf{W}_{,i}^h \cdot \mathbf{U}(\mathbf{Y}^h) - \mathbf{W}_{,i}^h \cdot \mathbf{F}_i(\mathbf{Y}^h) + \mathbf{W}_{,i}^h \cdot \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^h \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n} \mathbf{W}^h \cdot (\mathbf{F}_i(\mathbf{Y}^h) - \mathbf{F}_i^v(\mathbf{Y}^h)) n_i d\mathbf{P} \\ & + \int_{\Omega} (\mathbf{W}^h(t_{n+1}^-) \cdot \mathbf{U}(\mathbf{Y}^h(t_{n+1}^-)) - \mathbf{W}^h(t_n^+) \cdot \mathbf{U}(\mathbf{Y}^h(t_n^+))) d\Omega \\ & = \int_{\mathbf{Q}_n} \mathbf{W}^h \cdot \mathbf{S} d\mathbf{Q}. \end{aligned} \quad (4.22)$$

The third line in (4.22) is obtained by adding the *jump condition*

$$\int_{\Omega_n} \mathbf{W}^h(t_n^+) \cdot \llbracket \mathbf{U}(\mathbf{Y}^h(t_n)) \rrbracket d\Omega \quad (4.23)$$

to the terms resulting from the integration-by-parts of the time terms. For time-discontinuous Galerkin methods, the jump condition provides the mechanism by which information is propagated from one slab to the next. In other words, it imposes a weakly-enforced boundary condition. Additionally, as discussed by Shakib [71], the

jump condition adds a consistent high-order numerical dissipation which enhances the stability of the solution.

4.3.2 Non-linear system of equations

The solution approximation and weighting functions for the time-discontinuous Galerkin method within the n^{th} space-time slab can be written as

$$\mathbf{Y}^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x})(\pi_n^-(t)\mathbf{y}_{A;(n)} + \pi_n^+(t)\mathbf{y}_{A;(n+1)}), \quad (4.24)$$

$$\mathbf{W}^h(\mathbf{x}, t) = \sum_{A=1}^{n_{bf}} N_A(\mathbf{x})(\pi_n^-(t)\mathbf{w}_{A;(n)} + \pi_n^+(t)\mathbf{w}_{A;(n+1)}), \quad (4.25)$$

Let

$$\mathbf{w}_A = \{\mathbf{w}_{A;(n)}^T, \mathbf{w}_{A;(n+1)}^T\}^T \quad \text{and} \quad \bar{\pi}_n = \{\pi_n^-(t), \pi_n^+(t)\}^T, \quad (4.26)$$

then substituting (4.24),(4.25) and (4.26) into the space-time variational statement (4.22) yields

$$\begin{aligned} & \sum_{A=1}^{n_{bf}} \mathbf{w}_A^T \left\{ \int_{\mathcal{Q}_n} \left(-N_A \bar{\pi}_{n,t} \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) - \right. \right. \\ & \quad N_{A,i} \bar{\pi}_n \mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) + \\ & \quad \left. \left. N_{A,i} \bar{\pi}_n \tilde{\mathbf{K}}_{ij} \sum_{B=1}^{n_{bf}} N_{B,j} (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) - N_A \bar{\pi}_n \mathbf{S} \right) d\mathcal{Q} \right. \\ & + \int_{\mathcal{P}_n} N_A \bar{\pi}_n \left(\mathbf{F}_i \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) - \right. \\ & \quad \left. \mathbf{F}_i^v \left(\sum_{B=1}^{n_{bf}} N_B (\pi_n^- \mathbf{y}_{B;(n)} + \pi_n^+ \mathbf{y}_{B;(n+1)}) \right) \right) n_i d\mathcal{P} \\ & \left. + \int_{\Omega} N_A \mathbf{U} \left(\sum_{B=1}^{n_{bf}} N_B \mathbf{y}_{B;(n+1)} \right) - N_A \mathbf{U} \left(\sum_{B=1}^{(n_{bf})(n-1)} N_B^{(n-1)} \mathbf{y}_{B;(n)}^- \right) d\Omega \right\} = \mathbf{0}, \quad (4.27) \end{aligned}$$

where $N_B^{(n-1)}$ is the B^{th} global spatial mode of slab \mathbf{Q}_{n-1} , $\mathbf{y}_{B;(n)}^-$ is its corresponding vector of solution coefficients at time t_n^- , and $(n_{bf})_{(n-1)}$ is the number of global spatial basis functions of slab \mathbf{Q}_{n-1} . In the present research $(n_{bf})_{(n-1)} = n_{bf}$. Define

$$\mathbf{y}_{(n)}^- = \{\mathbf{y}_{1;(n)}^{-T}, \dots, \mathbf{y}_{(n_{bf})_{(n-1);(n)}}^{-T}\}^T \quad (4.28)$$

$$\mathbf{w} = \{\mathbf{w}_{1;(n)}^T, \dots, \mathbf{w}_{n_{bf};(n)}^T, \mathbf{w}_{1;(n+1)}^T, \dots, \mathbf{w}_{n_{bf};(n+1)}^T\}^T, \quad (4.29)$$

then using (4.7) and (4.8), (4.27) can be written as

$$\mathbf{w}^T \mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}, \mathbf{y}_{(n)}^-) = \mathbf{0}, \quad (4.30)$$

which must hold for all \mathbf{w} , so that

$$\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}, \mathbf{y}_{(n)}^-) = \mathbf{0}, \quad (4.31)$$

where $\mathbf{G}(\mathbf{y}_{(n+1)}; \mathbf{y}_{(n)}, \mathbf{y}_{(n)}^-)$ is an $(2n_{bf} \cdot m) \times 1$ system of non-linear algebraic equations. Note that since the method is discontinuous in time, the degrees of freedom $y_{(n)}$ are determined as part of the solution.

4.4 Predictor multi-corrector algorithm

To simplify notation, we write the non-linear systems of equations for the different time discretizations, i.e. (4.11), (4.19) and (4.31) as

$$\mathbf{G}(\mathbf{y}) = \mathbf{0}. \quad (4.32)$$

This system has to be solved within each time step, which is done by means of a Newton method. In general, each iteration of the Newton method can be considered as a pass of a predictor multi-corrector algorithm, where the predictor is given by the solution of the previous time step. The Newton method is derived from a first-order Taylor expansion about a current point $\mathbf{y}^{(k)}$:

$$\mathbf{G}(\mathbf{y}^{(k+1)}) = \mathbf{G}(\mathbf{y}^{(k)}) + \frac{\partial \mathbf{G}(\mathbf{y}^{(k)})}{\partial \mathbf{y}} (\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}), \quad (4.33)$$

where k denotes the index for the k^{th} -iteration. Setting the right hand side to zero yields an iterative method (given an initial guess $\mathbf{y}^{(0)}$) over a sequence of linear systems

$$\mathbf{J}^{(k)} \delta \mathbf{y}^{(k)} = -\mathbf{R}^{(k)}, \quad (4.34a)$$

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \delta \mathbf{y}^{(k)}, \quad k = 0, 1, \dots \quad (4.34b)$$

where

$$\delta \mathbf{y}^{(k)} = \mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}, \quad (4.35)$$

$$\mathbf{J}^{(k)} = \frac{\partial \mathbf{G}(\mathbf{y}^{(k)})}{\partial \mathbf{y}}, \quad (4.36)$$

$$\mathbf{R}^{(k)} = \mathbf{G}(\mathbf{y}^{(k)}), \quad (4.37)$$

where $\mathbf{J}^{(k)}$ is the tangent or Jacobian matrix, and $\mathbf{R}^{(k)}$ is the non-linear residual vector. The iteration is terminated based on a required drop in the norm of the residual of each of the components (i.e. mass, momentum, total energy)

$$\frac{\|\mathbf{R}_i^{(k)}\|}{\|\mathbf{R}_i^{(0)}\|} < \varepsilon_{res}, \quad i \in \{1, \dots, m\}, \quad (4.38)$$

and/or a sufficiently small Newton update for each of the components

$$\frac{\|\delta \mathbf{y}_i^{(k)}\|}{\|\mathbf{y}_i^{(k)}\|} < \varepsilon_{update}, \quad i \in \{1, \dots, m\}. \quad (4.39)$$

Forming the Jacobian matrix can be a very-time consuming operation. In practice, however, one need not update the Jacobian matrix each Newton iteration, or even time step. When the Jacobian is held constant in such a manner, the method falls into a class known as modified Newton methods. In general, when the Jacobian is not exact, quadratic non-linear convergence cannot be guaranteed. Therefore, the

constant Jacobian approach requires a check on the convergence rate to ensure the Jacobian is updated with sufficient frequency.

4.5 Assembly of the linear systems

The assembly of $\mathbf{J}^{(k)}$ and $\mathbf{R}^{(k)}$ can be represented as

$$\mathbf{J}^{(k)} = \mathbf{A} \sum_{e=1}^{n_{el}} \mathbf{J}^{e;(k)}, \quad (4.40)$$

$$\mathbf{R}^{(k)} = \mathbf{A} \sum_{e=1}^{n_{el}} \mathbf{R}^{e;(k)}, \quad (4.41)$$

where \mathbf{A} is the finite-element assembly operator, and $\mathbf{J}^{e;(k)}$ and $\mathbf{R}^{e;(k)}$ are the element contributions to the global Jacobian matrix and residual vector at the k^{th} iteration.

4.5.1 Time-continuous Galerkin

The solution approximation of the TCG method within the e^{th} element at the k^{th} Newton iteration can be expressed as

$$\mathbf{Y}^{e;(k)}(\mathbf{x}, t) = \sum_{a=1}^{n_{eb}} N_a^e(\mathbf{x}) \left(\pi_n^-(t) \mathbf{y}_{a;(n)} + \pi_n^+(t) \mathbf{y}_{a;(n+1)}^{(k)} \right) \quad (4.42)$$

for $\mathbf{x} \in \Omega^e, t \in I_n$, where n_{eb} is the number of (local) element basis functions, $N_a^e(\mathbf{x})$ is the a^{th} elemental spatial basis function, $\mathbf{y}_{a;(n+1)}^{(k)}$ is the corresponding vector of solution coefficients at time t_{n+1} and $\mathbf{y}_{a;(n)}$ is the initial condition. Note that lower case indices denote local ordering, as described in Appendix C. The elemental residual vector and

Jacobian matrix can be written as

$$\mathbf{R}^{e;(k)} = \{ \mathbf{R}_a^{e;(k)} \} \quad a = 1, \dots, n_{eb} \quad (4.43)$$

$$\begin{aligned} \mathbf{R}_a^{e;(k)} = & \int_{\mathbf{Q}_n^e} \left(-N_a^e \pi_{n,t}^+ \mathbf{U}(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) - N_{a,i}^e \pi_n^+ \mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) + \right. \\ & \left. N_{a,i}^e \pi_n^+ \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^{e;(k)}(\mathbf{x}, t) - N_a^e \pi_n^+ \mathbf{S} \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n^e} N_a^e \pi_n^+ \left(\mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) - \mathbf{F}_i^v(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) \right) n_i d\mathbf{P} \\ & + \int_{\Omega^e} N_a^e \mathbf{U}(\mathbf{Y}^{e;(k)}(\mathbf{x}, t_{n+1})) d\Omega, \end{aligned} \quad (4.44)$$

and

$$\mathbf{J}^{e;(k)} = \left[\mathbf{J}_{ab}^{e;(k)} \right] \quad a, b = 1, \dots, n_{eb} \quad (4.45)$$

$$\begin{aligned} \mathbf{J}_{ab}^{e;(k)} = & \int_{\mathbf{Q}_n^e} \left(-N_a^e \pi_{n,t}^+ \tilde{\mathbf{A}}_0 N_b^e - N_{a,i}^e \pi_n^+ \tilde{\mathbf{A}}_i N_b^e + N_{a,i}^e \pi_n^+ \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n^e} N_a^e \pi_n^+ \left(\tilde{\mathbf{A}}_i N_b^e - \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) n_i d\mathbf{P} \\ & + \int_{\Omega_{n+1}^e} N_a^e \pi_n^+ \tilde{\mathbf{A}}_0 N_b^e d\Omega \end{aligned} \quad (4.46)$$

4.5.2 Time-continuous Petrov-Galerkin

The solution approximation of the TCPG method within the e^{th} element at the k^{th} Newton iteration can be expressed as

$$\mathbf{Y}^{e;(k)}(\mathbf{x}, t) = \sum_{a=1}^{n_{eb}} N_a^e(\mathbf{x}) \left(\pi_n^-(t) \mathbf{y}_{a;(n)} + \pi_n^+(t) \mathbf{y}_{a;(n+1)}^{(k)} \right) \quad (4.47)$$

for $\mathbf{x} \in \Omega^e, t \in I_n$. The elemental residual vector and Jacobian matrix can then be written as

$$\mathbf{R}^{e;(k)} = \{\mathbf{R}_a^{e;(k)}\} \quad a = 1, \dots, n_{eb} \quad (4.48)$$

$$\begin{aligned} \mathbf{R}_a^{e;(k)} &= \int_{\mathbf{Q}_n^e} \left(-N_{a,i}^e \mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) + N_{a,i}^e \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^{e;(k)}(\mathbf{x}, t) - N_a^e \mathbf{S} \right) d\mathbf{Q} \\ &+ \int_{\mathbf{P}_n^e} N_a^e (\mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) - \mathbf{F}_i^v(\mathbf{Y}^{e;(k)}(\mathbf{x}, t))) n_i d\mathbf{P} \\ &+ \int_{\Omega^e} N_a^e \mathbf{U}(\mathbf{Y}^{e;(k)}(\mathbf{x}, t_{n+1})) - N_a^e \mathbf{U}(\mathbf{Y}^e(\mathbf{x}, t_n)) d\Omega, \end{aligned} \quad (4.49)$$

and

$$\mathbf{J}^{e;(k)} = [\mathbf{J}_{ab}^{e;(k)}] \quad a, b = 1, \dots, n_{eb} \quad (4.50)$$

$$\begin{aligned} \mathbf{J}_{ab}^{e;(k)} &= \int_{\mathbf{Q}_n^e} \left(-N_{a,i}^e \tilde{\mathbf{A}}_i N_b^e + N_{a,i}^e \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) d\mathbf{Q} \\ &+ \int_{\mathbf{P}_n^e} N_a^e \left(\tilde{\mathbf{A}}_i N_b^e - \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) n_i d\mathbf{P} \\ &+ \int_{\Omega_{n+1}^e} N_a^e \tilde{\mathbf{A}}_0 N_b^e d\Omega \end{aligned} \quad (4.51)$$

4.5.3 Time-discontinuous Galerkin

The solution approximation of the TDG method within the e^{th} element at the k^{th} Newton iteration can be expressed as

$$\mathbf{Y}^{e;(k)}(\mathbf{x}, t) = \sum_{a=1}^{n_{eb}} N_a^e(\mathbf{x}) \left(\pi_n^-(t) \mathbf{y}_{a;(n)}^{(k)} + \pi_n^+(t) \mathbf{y}_{a;(n+1)}^{(k)} \right) \quad (4.52)$$

for $\mathbf{x} \in \Omega^e, t \in I_n$. The elemental residual vector and Jacobian matrix can then be written as

$$\mathbf{R}^{e;(k)} = \{ \mathbf{R}_a^{e;(k)} \} \quad a = 1, \dots, n_{eb} \quad (4.53)$$

$$\begin{aligned} \mathbf{R}_a^{e;(k)} = & \int_{\mathbf{Q}_n^e} \left(-N_a^e \bar{\pi}_{n,t} \mathbf{U}(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) - N_{a,i}^e \bar{\pi}_n \mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) + \right. \\ & \left. N_{a,i}^e \bar{\pi}_n \tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j}^{e;(k)}(\mathbf{x}, t) - N_a^e \bar{\pi}_n \mathbf{S} \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n^e} N_a^e \bar{\pi}_n (\mathbf{F}_i(\mathbf{Y}^{e;(k)}(\mathbf{x}, t)) - \mathbf{F}_i^v(\mathbf{Y}^{e;(k)}(\mathbf{x}, t))) n_i d\mathbf{P} \\ & + \int_{\Omega^e} N_a^e \mathbf{U}(\mathbf{Y}^{e;(k)}(\mathbf{x}, t_{n+1})) - N_a^e \mathbf{U}(\mathbf{Y}^e(\mathbf{x}, t_n^-)) d\Omega, \end{aligned} \quad (4.54)$$

and

$$\mathbf{J}^{e;(k)} = \left[\mathbf{J}_{ab}^{e;(k)} \right] \quad a, b = 1, \dots, n_{eb} \quad (4.55)$$

$$\begin{aligned} \mathbf{J}_{ab}^{e;(k)} = & \int_{\mathbf{Q}_n^e} \left(-N_a^e \bar{\pi}_{n,t} \tilde{\mathbf{A}}_0 N_b^e - N_{a,i}^e \bar{\pi}_n \tilde{\mathbf{A}}_i N_b^e + N_{a,i}^e \bar{\pi}_n \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) d\mathbf{Q} \\ & + \int_{\mathbf{P}_n^e} N_a^e \bar{\pi}_n \left(\tilde{\mathbf{A}}_i N_b^e - \tilde{\mathbf{K}}_{ij} N_{b,j}^e \right) n_i d\mathbf{P} \\ & + \int_{\Omega_{n+1}^e} N_a^e \tilde{\mathbf{A}}_0 N_b^e d\Omega. \end{aligned} \quad (4.56)$$

4.6 Solution method for linear systems

As described in Section 4.4 the non-linear solution algorithm requires the solution of a sequence of linear systems within each time step. Solution methods for solving linear systems of equations can be divided into direct and iterative solution techniques. Typically, direct solution techniques make use of Gaussian elimination by means of

Lower-Upper (LU) decompositions. These methods are advantageous in that the work they require is not directly related to the conditioning of the system. However, for the current discretization direct methods are very unattractive. The use of high-order finite-element expansions results in increased bandwidths of the Jacobian matrices (\mathbf{J}) compared to lower-order discretizations. For this reason, the lower and upper factorizations contain many more non-zero entries (called *fill-ins*) than present in the original Jacobian matrix. For large problems, this may result in excessive memory requirements and poor scaling of the required number of operations with the matrix dimension. Therefore, one has to resort to iterative solution techniques, such as Krylov subspace methods.

4.6.1 Iterative solution techniques

An attractive class of iterative solution techniques are Krylov methods, which are specifically designed for solving large linear systems of equations [67]. These are generalized projection methods which create a Krylov subspace \mathcal{K}_j from which the solution is constructed. For the present system of equations (4.34a), the Krylov subspace is defined as

$$\mathcal{K}_j = \text{span}\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, \mathbf{J}^2\mathbf{r}_0, \dots, \mathbf{J}^{j-1}\mathbf{r}_0\}, \quad (4.57)$$

where $\mathbf{r}_0 = -\mathbf{R} - \mathbf{J}\delta\mathbf{y}_0$ is the initial linear residual given an suitably chosen initial guess $\delta\mathbf{y}_0$, and $\mathbf{J}^j\mathbf{r}_0$, $j = 0, \dots, j-1$, are so-called Krylov vectors. Krylov methods require only matrix-vector products to carry out the iteration. This is a key feature for their use in a Jacobian-free Newton method, as described in Section 4.6.2. A widely used Krylov method suitable for non-symmetric systems is the Generalized Minimal Residual (GMRES) algorithm [66]. This algorithm has been successfully employed for many compressible flow problems [21, 36, 71], and is also adopted in the present simulations.

In the absence of a good initial guess for the Krylov method, $\delta\mathbf{y}_0$ is typically set to zero. This seems a reasonable guess in the context of a non-linear Newton method, since the Newton iterates $\delta\mathbf{y}_j$ should approach zero in late iterations. The j^{th} iteration of the GMRES algorithm minimizes the residual $\|\mathbf{R} + \mathbf{J} \delta\mathbf{y}_j\|$ within a subspace of small dimension relative to the total number of unknowns of the system. The j^{th} iterate $\delta\mathbf{y}_j$ is constructed from a linear combination of the Krylov vectors

$$\delta\mathbf{y}_j = \delta\mathbf{y}_0 + \sum_{i=1}^{j-1} \beta_i \mathbf{J}^i \mathbf{r}_0, \quad (4.58)$$

where the scalars β_i minimize the residual. In practice, $\delta\mathbf{y}_j$ is determined as a linear combination of the orthonormalized Krylov vectors. The solution of the linear system is considered converged based on a required drop in the norm of the residual relative to the initial residual

$$\frac{\|\mathbf{R} + \mathbf{J}\mathbf{y}_j\|}{\|\mathbf{R} + \mathbf{J}\mathbf{y}_0\|} < \varepsilon_{gmres}. \quad (4.59)$$

For a detailed description of the GMRES algorithm (including preconditioning) the reader is referred to standard works such as that of Saad [67].

4.6.2 Jacobian-free Newton Krylov

As discussed in the previous section, Krylov methods use matrix-vector products to determine the approximate solution of the linear system. Specifically, the GMRES method uses one matrix-vector product of the form $\mathbf{J}\mathbf{v}$ per iteration, where \mathbf{v} represents a Krylov vector. Since \mathbf{J} is a Jacobian matrix, this product can be approximated by

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{y} + \epsilon\mathbf{v}) - \mathbf{R}(\mathbf{y})}{\epsilon}, \quad (4.60)$$

where ϵ is a small parameter. Note that (4.60) is simply a first-order Taylor expansion approximation for the Jacobian times a vector. In this way, the product $\mathbf{J}\mathbf{v}$ can be approximated without actually forming and storing the Jacobian matrix \mathbf{J} . However, this approach requires additional residual evaluations, $\mathbf{R}(\mathbf{y} + \epsilon\mathbf{v})$, one for each

GMRES iteration. This approach is referred to as a Jacobian-Free Newton-Krylov (JFNK) method.

The choice of the perturbation parameter ϵ in (4.60) is as much of an art as a science [42]. If ϵ is too large, (4.60) is a poor approximation due to the truncation error from omitting higher-order terms in the Taylor expansion. On the other hand, if ϵ is too small, the result will be contaminated by floating-point roundoff error. Therefore, the choice for ϵ should attempt to optimize the balance between both sources of error. A simple definition for ϵ that has been successfully used for compressible flow problems [36, 42, 49, 59, 76] is

$$\epsilon = \frac{1}{n\|\mathbf{v}\|} \left[\sum_{i=1}^n (b|y_i|) + b \right], \quad (4.61)$$

where n is the dimension of the linear system and b is a constant whose magnitude is within a few orders of the square root of machine roundoff (typically 10^{-6} for 64 bit double precision).

4.6.3 Preconditioning

Lack of robustness is a widely recognized weakness of iterative solution techniques (such as GMRES), compared to direct solvers. This drawback prevents these methods from being used more widely in industrial applications, despite their inherent appeal for solving very large linear systems of equations. Both the efficiency and robustness can be improved by *preconditioning*. Preconditioning is a means of transforming the original system of equations into one which has the same solution, but which is easier to solve with an iterative solver. In general, the reliability of iterative solvers depends much more on the quality of the preconditioner than on the particular iterative solver itself [67].

A linear preconditioner can be applied on the left of the system (rescaling the matrix rows and the right-hand side) or on the right (rescaling the matrix rows and the

solution vector), or on both if necessary. Left preconditioning changes the norm of the residual by which convergence of the iterative solution method is generally measured. Therefore, right preconditioning is often preferred when comparing performance of different preconditioning strategies.

For linear system 4.34a right-preconditioning constitutes

$$(\mathbf{JP}^{-1})(\mathbf{P}\delta\mathbf{y}) = -\mathbf{R}, \quad (4.62)$$

where \mathbf{P} is the preconditioning matrix which approximates, in some sense, the matrix \mathbf{J} . Right preconditioning is realized in a two-step process, i.e. first solve

$$(\mathbf{JP}^{-1})\mathbf{w} = -\mathbf{R}, \quad (4.63)$$

for \mathbf{w} , and then reconstruct the actual solution \mathbf{y} by

$$\delta\mathbf{y} = \mathbf{P}^{-1}\mathbf{w}. \quad (4.64)$$

Note that, while the matrix \mathbf{P} is referred to as the preconditioning matrix, operationally only the action of its inverse, \mathbf{P}^{-1} , is required.

One of the most common preconditioners are the class of Incomplete Lower-Upper (ILU) factorizations. This type of preconditioner has been successfully applied to many compressible flow problems [9, 19] and is also adopted in the present simulations. An incomplete factorization discards many, or even all fill-ins (Section 4.6), so that it is relatively easy to compute and requires approximately the same memory as the Jacobian matrix itself. The product of triangular factors no longer equals the Jacobian matrix, however it can still be used as an effective preconditioner, given by

$$\mathbf{P} = \mathbf{LU} \Leftrightarrow \mathbf{P}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}, \quad (4.65)$$

In the present simulations all fill-ins are discarded. This variant is often called ILU0. Incomplete factorizations are sequential in nature and difficult to implement efficiently on a parallel computer. For parallel computations we employ a block variant

of the incomplete factorization, where the incomplete factorization for each partition is used to precondition the corresponding partition's system of equations. In general, the performance of this parallel implementation is less than its single-processor counterpart, as will be demonstrated in Chapter 6.

It might be hoped that effective preconditioners can typically be simpler than the strict Jacobian of the system. For a low-order finite element discretization of the compressible Navier-Stokes equations, Johan [36] obtained good performance by using a preconditioner based on a block-diagonal variants of the Jacobian matrix. However, for the present high-order discretizations this approach was found to be far less effective than the ILU preconditioner.

Note that right or left preconditioning may also be employed in a Jacobian-free method. To do so, the right-preconditioned version of the required matrix-vector product (4.60) can be written

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{R}(\mathbf{y} + \epsilon\mathbf{P}^{-1}\mathbf{v}) - \mathbf{R}(\mathbf{y})}{\epsilon}. \quad (4.66)$$

This operation must be performed once per GMRES iteration, and is actually done in two steps:

1. preconditioning: solve $\mathbf{P}\mathbf{u} = \mathbf{v}$ for \mathbf{u}
2. perform Jacobian-free product: $\mathbf{J}\mathbf{u} \approx [\mathbf{R}(\mathbf{y} + \epsilon\mathbf{u}) - \mathbf{R}(\mathbf{y})]/\epsilon$

The use of a fixed Jacobian (computed from an earlier step) for the preconditioner may be advantageous in the context of a JFNK method. This approach is different from that for a traditional modified Newton method, in which both the Jacobian and its preconditioner are held constant over multiple Newton iterations. The modified Newton approach has much weaker non-linear convergence properties [42] than the JFNK approach. In the latter, the true Jacobian (to within truncation error of (4.60)) is felt at every iteration through the matrix-vector product approximation. Note that this approach is not truly Jacobian-free since Jacobians are still formed

and stored for preconditioning purposes. Moreover, the JFNK method requires an additional residual vector evaluation for each GMRES iteration, which is typically more expensive than a matrix-vector product.

4.7 Application to laminar channel flow

We consider laminar viscous flow between two parallel horizontal plates of infinite width as illustrated in Figure 4.1, where $x = x_1$ is the streamwise direction and $y = x_2$ is the wall-normal direction. This problem is the laminar counterpart of the large eddy simulation problem which will be considered in the next chapters. Assuming the viscosity to be independent of the temperature allows a closed-form exact solution to be derived for the velocity and temperature profiles

$$u_1 = u_{max} \left[1 - \left(\frac{y}{\delta} \right)^2 \right], \quad v = 0, \quad (4.67)$$

$$T = T_w + \frac{\mu u_{max}^2}{3\kappa} \left[1 - \left(\frac{y}{\delta} \right)^4 \right], \quad (4.68)$$

$$u_{max} = -\frac{\delta}{2\mu} \left(\frac{dp}{dx} \right), \quad (4.69)$$

where δ is the half-height of the channel, u_{max} is the centerline velocity of the channel (i.e. at $y = 0$), T_w is the prescribed wall temperature, and (4.69) relates the centerline velocity to the pressure gradient that drives the flow. If the viscosity is assumed constant throughout the flow, this solution holds for both incompressible and compressible flow [34].

The flow is simulated using periodic boundary conditions in the streamwise direction, along with a constant wall temperature $T = T_w$ and the no-slip condition $u = v = 0$ on the upper and lower walls of the channel. The periodic boundary conditions enforce all variables of the inflow plane to be identical to those of the outflow plane. Consequently, the flow cannot be driven by a pressure gradient. Instead, its

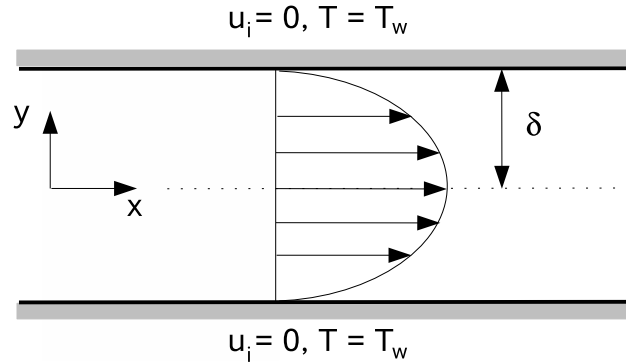


Figure 4.1: Laminar channel flow geometry and boundary conditions.

effect is mimicked by including a body force f_1 in the x -momentum equation, with a value equal to that of the pressure gradient

$$f_1 = -\frac{dp}{dx} = 2\frac{u_{max}\delta}{\mu}, \quad (4.70)$$

where the minus sign comes from the fact that a negative pressure gradient forces the flow in positive x -direction.

A convergence study using primitive density variables was performed for spatial polynomial degrees $P = 1, 2$ and 3 . We considered the L_2 -error in the temperature profile, computed from (3.31). The error versus the mesh size in wall-normal direction is shown in Figure 4.2. The observed orders of convergence, shown in Tables 4.1 to 4.3, are in good agreement with theory for the polynomial degrees considered. Figure 4.3 shows the error versus the number of degrees of freedom in the wall-normal direction. For this relatively smooth problem, the higher-order methods have higher accuracy for a given number of degrees of freedom. Note that the convergence behavior for this problem is very similar to that observed for the linear convection-diffusion problem described in the previous chapter.

The present results do not necessarily imply that higher-order solutions are computationally more efficient. For example, for a given number of degrees of freedom, higher-order methods typically lead to fuller matrices compared to lower-methods,

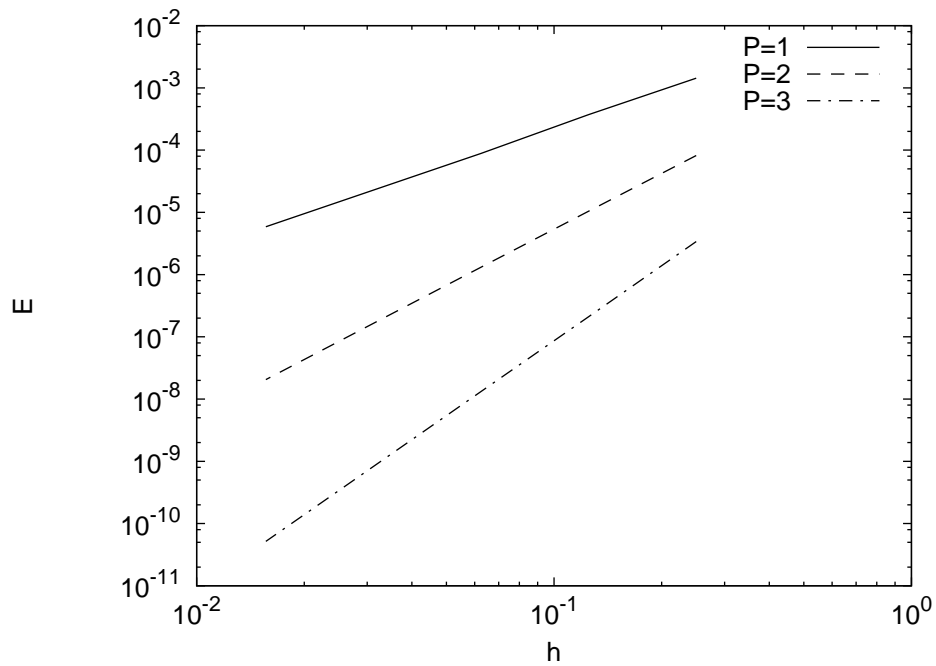


Figure 4.2: Convergence of temperature profile in laminar channel flow; error versus the element size in wall-normal direction.

making the linear solution technique much more expensive. A more detailed relative cost analysis is given in Chapter 6 for large eddy simulations of turbulent channel flows.

mesh	P	d.o.f.	error	order
1×4	1	28	4.5e-3	-
1×8	1	60	3.7e-4	3.61
1×16	1	124	8.8e-5	2.07
1×32	1	252	2.2e-5	1.96
1×64	1	508	5.9e-6	1.96

Table 4.1: Simulation parameters for laminar channel flow using $P = 1$.

mesh	P	d.o.f.	error	order
1×4	2	120	8.1e-5	-
1×8	2	248	1.0e-5	2.97
1×16	2	504	1.3e-6	2.99
1×32	2	1016	1.6e-7	3.00
1×64	2	2040	2.1e-8	3.00

Table 4.2: Simulation parameters for laminar channel flow using $P = 2$.

mesh	P	d.o.f.	error	order
1×4	3	276	3.4e-6	-
1×8	3	564	2.1e-7	4.00
1×16	3	1140	1.3e-8	4.00
1×32	3	2292	8.3e-10	4.00
1×64	3	4596	5.2e-11	4.00

Table 4.3: Simulation parameters for laminar channel flow using $P = 3$.

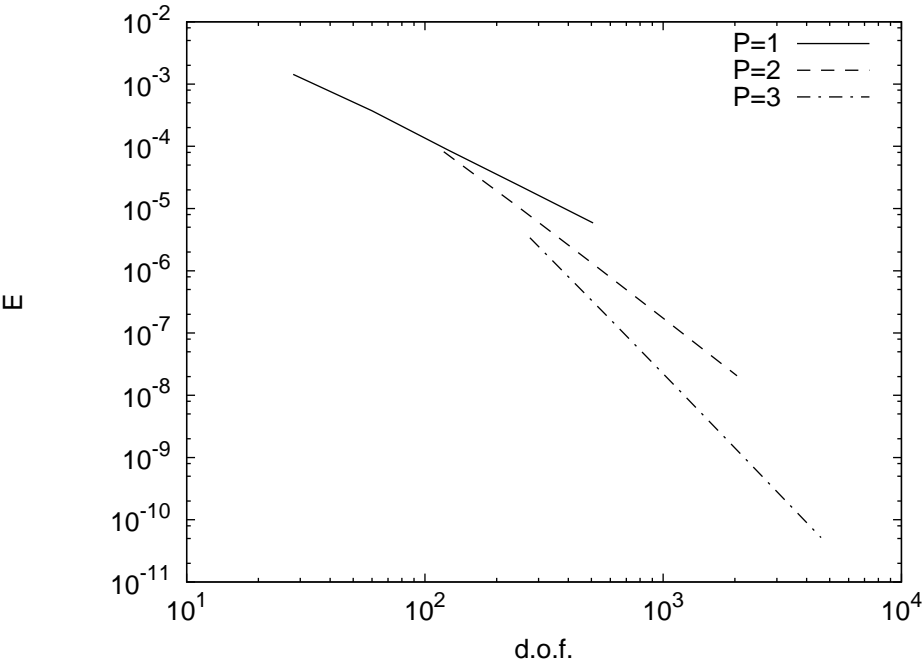


Figure 4.3: Convergence of temperature profile in laminar channel flow; error versus the number of degrees of freedom.

Chapter 5

The variational multiscale method

In this chapter, we examine the ability of the proposed discretization to exploit the advantages of the VMS method. To do so, we consider the effects of the spatial discretization and subgrid-scale modeling parameters on the quality of the computed turbulence statistics for turbulent channel flow. In Section 5.1 a brief outline is given of the VMS method and its implementation using the present hierarchical finite-element basis. Section 5.2 gives a detailed description of the turbulent channel flow problem, including the initial conditions used for the simulations, and the calculation of the turbulent statistics. The numerical results are presented in Section 5.3, where we focus on resolution requirements and subgrid-scale modeling parameters. In particular, we compare results obtained using the VMS method to those obtained using no explicit SGS modeling and a traditional LES formulation using a similar SGS model as the VMS method, but on all scales.

5.1 Variational multiscale formulation

Here, we give a brief outline of the VMS method. A more detailed description can be found in Collis [10] and Hughes *et al.* [29]. We follow the three-level multiscale formulation introduced by Collis [10], in which both the solution and weighting functions

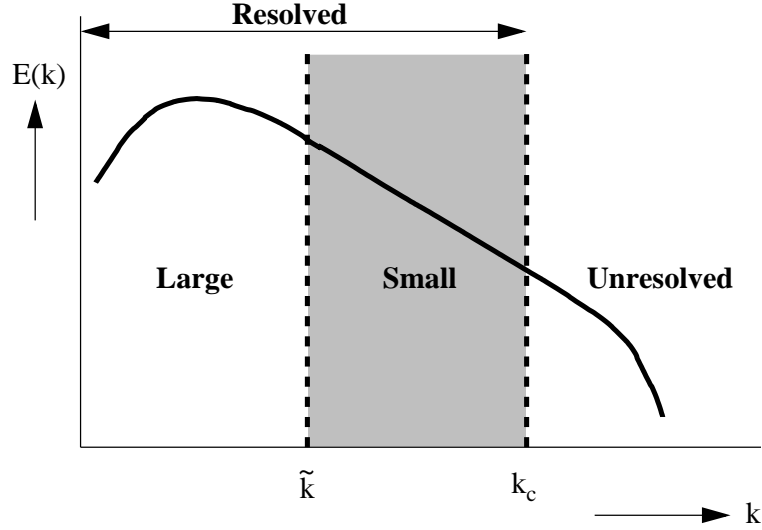


Figure 5.1: Identification of scales in the VMS method

are decomposed as

$$\mathbf{U} = \overline{\mathbf{U}} + \tilde{\mathbf{U}} + \hat{\mathbf{U}}, \quad \mathbf{W} = \overline{\mathbf{W}} + \tilde{\mathbf{W}} + \hat{\mathbf{W}}, \quad (5.1)$$

where $\overline{\mathbf{U}}$ are the large scales, $\tilde{\mathbf{U}}$ are the small scales and $\hat{\mathbf{U}}$ are the unresolved scales. A similar decomposition is used for the weighting functions. Note that both the large and small scales are part of the resolved scales. Intuitively, these scales may be thought of as Fourier modes in wavenumber space [12] where the large and small scales are represented by the low and medium wavenumber components, and the unresolved scales by the remaining high wavenumber components, as shown in Figure 5.1.

To simplify notation, we introduce the following compact notation for the variational forms of the Navier-Stokes equations

$$B(\mathbf{W}, \mathbf{U}) = (\mathbf{W}, \mathbf{S}), \quad (5.2)$$

where $B(\mathbf{W}, \mathbf{U})$ represents the left-hand side of the variational forms of the different methods, i.e. (4.3), (4.14) or (4.22) and (\mathbf{W}, \mathbf{S}) represents the corresponding right-hand sides. Substituting (5.1) into (5.2) leads to the equations describing the

dynamics of each scale range

$$\text{Large:} \quad B(\overline{\mathbf{W}}, \overline{\mathbf{U}} + \tilde{\mathbf{U}} + \hat{\mathbf{U}}) = (\overline{\mathbf{W}}, \mathbf{S}), \quad (5.3)$$

$$\text{Small:} \quad B(\widetilde{\mathbf{W}}, \overline{\mathbf{U}} + \tilde{\mathbf{U}} + \hat{\mathbf{U}}) = (\widetilde{\mathbf{W}}, \mathbf{S}), \quad (5.4)$$

$$\text{Unresolved:} \quad B(\widehat{\mathbf{W}}, \hat{\mathbf{U}} + \tilde{\mathbf{U}} + \hat{\mathbf{U}}) = (\widehat{\mathbf{W}}, \mathbf{S}), \quad (5.5)$$

which is a coupled set of equations, referred to as the large, small, and unresolved-scale equations. If a non-orthogonal finite element basis is employed, the large-scale equation can be written as

$$\begin{aligned} B(\overline{\mathbf{W}}, \overline{\mathbf{U}}) + B'(\overline{\mathbf{W}}, \overline{\mathbf{U}}, \tilde{\mathbf{U}}) - R(\overline{\mathbf{W}}, \tilde{\mathbf{U}}) &= (\overline{\mathbf{W}}, \mathbf{S}) \\ - B'(\overline{\mathbf{W}}, \overline{\mathbf{U}}, \hat{\mathbf{U}}) + R(\overline{\mathbf{W}}, \hat{\mathbf{U}}) + C(\overline{\mathbf{W}}, \tilde{\mathbf{U}}, \hat{\mathbf{U}}), & \end{aligned} \quad (5.6)$$

and the small-scale equation is

$$\begin{aligned} B'(\widetilde{\mathbf{W}}, \overline{\mathbf{U}}, \tilde{\mathbf{U}}) - R(\widetilde{\mathbf{W}}, \tilde{\mathbf{U}}) &= -[B(\widetilde{\mathbf{W}}, \overline{\mathbf{U}}) - (\widetilde{\mathbf{W}}, \mathbf{S})] \\ - B'(\widetilde{\mathbf{W}}, \overline{\mathbf{U}}, \hat{\mathbf{U}}) + R(\widetilde{\mathbf{W}}, \hat{\mathbf{U}}) + C(\widetilde{\mathbf{W}}, \tilde{\mathbf{U}}, \hat{\mathbf{U}}). & \end{aligned} \quad (5.7)$$

In these expressions, $B'(\overline{\mathbf{W}}, \overline{\mathbf{U}}, \mathbf{U}')$ is the operator $B(\overline{\mathbf{W}}, \overline{\mathbf{U}})$ linearized about $\overline{\mathbf{U}}$ for a linear perturbation \mathbf{U}' . For example, the linearized operator for the TDG method (4.22) is defined as

$$\begin{aligned} B'(\overline{\mathbf{W}}, \overline{\mathbf{U}}, \mathbf{U}') &= \int_{\mathbf{Q}_n} (-\overline{\mathbf{W}}_{,t} \cdot \mathbf{U}' - \overline{\mathbf{W}}_{,i} \cdot \mathbf{F}_i(\overline{\mathbf{U}}, \mathbf{U}') + \overline{\mathbf{W}}_{,i} \cdot \mathbf{F}_i^v(\overline{\mathbf{U}}, \mathbf{U}')) \, d\mathbf{Q} \\ &+ \int_{\mathbf{P}_n} \overline{\mathbf{W}} \cdot (\mathbf{F}_i(\overline{\mathbf{U}}, \mathbf{U}') - \mathbf{F}_i^v(\overline{\mathbf{U}}, \mathbf{U}')) \, n_i \, d\mathbf{P} \\ &+ \int_{\Omega} (\overline{\mathbf{W}}(t_{n+1}^-) \cdot \mathbf{U}'(t_{n+1}^-) - \overline{\mathbf{W}}(t_n^+) \cdot \mathbf{U}'(t_n^-)) \, d\Omega, \end{aligned} \quad (5.8)$$

where

$$\mathbf{F}_i(\bar{\mathbf{U}}, \mathbf{U}') = \bar{u}_i \mathbf{U}' + u'_i \bar{\mathbf{U}} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \bar{p}u'_i + p'\bar{u}_i \end{pmatrix}, \quad \mathbf{F}_i^v(\bar{\mathbf{U}}, \mathbf{U}') = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \bar{\tau}_{ij}u'_j + \tau'_{ij}\bar{u}_j \end{pmatrix}, \quad (5.9)$$

$C(\bar{\mathbf{W}}, \tilde{\mathbf{U}}, \hat{\mathbf{U}})$ is the generalized cross stress projection onto the large scales, defined as

$$\begin{aligned} C(\bar{\mathbf{W}}, \tilde{\mathbf{U}}, \hat{\mathbf{U}}) &= \int_{\mathbf{Q}_n} \left(\bar{\mathbf{W}}_{,i} \cdot \mathbf{F}_i(\tilde{\mathbf{U}}, \hat{\mathbf{U}}) - \bar{\mathbf{W}}_{,i} \cdot \mathbf{F}_i^v(\tilde{\mathbf{U}}, \hat{\mathbf{U}}) \right) d\mathbf{Q} \\ &\quad - \int_{\mathbf{P}_n} \bar{\mathbf{W}} \cdot \left(\mathbf{F}_i(\tilde{\mathbf{U}}, \hat{\mathbf{U}}) - \mathbf{F}_i^v(\tilde{\mathbf{U}}, \hat{\mathbf{U}}) \right) n_i d\mathbf{P}, \end{aligned} \quad (5.10)$$

and $R(\bar{\mathbf{W}}, \tilde{\mathbf{U}})$ is the generalized Reynolds stress projection onto the large scales, defined as

$$\begin{aligned} R(\bar{\mathbf{W}}, \tilde{\mathbf{U}}) &= \int_{\mathbf{Q}_n} \left(\bar{\mathbf{W}}_{,i} \cdot \mathbf{F}_i(\tilde{\mathbf{U}}) - \bar{\mathbf{W}}_{,i} \cdot \mathbf{F}_i^v(\tilde{\mathbf{U}}) \right) d\mathbf{Q} \\ &\quad - \int_{\mathbf{P}_n} \bar{\mathbf{W}} \cdot \left(\mathbf{F}_i(\tilde{\mathbf{U}}) - \mathbf{F}_i^v(\tilde{\mathbf{U}}) \right) n_i d\mathbf{P}. \end{aligned} \quad (5.11)$$

For incompressible flows, $C(\bar{\mathbf{W}}, \tilde{\mathbf{U}}, \hat{\mathbf{U}})$ and $R(\bar{\mathbf{W}}, \tilde{\mathbf{U}})$ are identical to the projection of the cross stress and Reynolds stress onto the large scales [10]. For compressible flows, these terms are considerably more complicated due to additional terms arising from the variable density and terms in the energy equation. See Refs. [17, 43] for a more detailed description of these terms.

5.1.1 Subgrid-scale modeling

Here, we make the following subgrid-scale (SGS) modeling assumptions:

- (1) The unresolved scales have negligible direct influence on the dynamics of the large scales. This is a reasonable assumption, provided there is sufficient scale separation between the large and unresolved scales. As a result, terms involving the unresolved scales in the large-scale equation (i.e. the second line in (5.6)) are set to zero, so that the large scale equation becomes

$$B(\overline{\mathbf{W}}, \overline{\mathbf{U}}) - B'(\overline{\mathbf{W}}, \overline{\mathbf{U}}, \tilde{\mathbf{U}}) - R(\overline{\mathbf{W}}, \tilde{\mathbf{U}}) = (\overline{\mathbf{W}}, \mathbf{S}). \quad (5.12)$$

- (2) The small and unresolved scales, on the other hand, are closer in scale and thus the effect of the unresolved scales on the small scales must be modeled. Consequently, all terms involving the unresolved scales in the small-scale equation (i.e. the second line in (5.7)) are replaced by a model $\mathbf{M}(\tilde{\mathbf{W}}, \tilde{\mathbf{U}})$

$$B'(\tilde{\mathbf{W}}, \overline{\mathbf{U}}, \tilde{\mathbf{U}}) - R(\tilde{\mathbf{W}}, \tilde{\mathbf{U}}) = -[B(\tilde{\mathbf{W}}, \overline{\mathbf{U}}) - (\tilde{\mathbf{W}}, \mathbf{S})] + \mathbf{M}(\tilde{\mathbf{W}}, \tilde{\mathbf{U}}). \quad (5.13)$$

The above modeling assumptions do not imply that the large scales do not feel the influence of the model. In fact, the large scales are still indirectly influenced by the model through its interaction with the small scales. However, the coupling terms in the large-scale equation which provide this interaction are still in their exact form. Moreover, since the large-scale equation does not have explicit modeling terms, it allows full-rate of convergence of the underlying numerical discretization. This key feature is referred to as the consistency property of the large scales with the Navier-Stokes equations [10, 29].

A thorough discussion on subgrid-scale modeling for compressible flows is provided by Martin *et al.* [50]. Here, we make the assumption that if the unresolved scales are sufficiently small, their effect can be reasonably modeled using an eddy-viscosity assumption, as is commonly done in LES. This concept stems from the analogy between turbulent stresses and viscous stresses, so that the model term can be written as

$$\mathbf{M}(\tilde{\mathbf{W}}, \tilde{\mathbf{U}}) = - \int_{\mathbf{Q}_n} \tilde{\mathbf{W}}_{,i} \cdot \mathbf{F}_i^m(\tilde{\mathbf{U}}) \, d\mathbf{Q} + \int_{\mathbf{P}_n} \tilde{\mathbf{W}} \cdot \mathbf{F}_i^m(\tilde{\mathbf{U}}) n_i \, d\mathbf{P}, \quad (5.14)$$

where $\mathbf{F}_i^m(\tilde{\mathbf{U}})$ is the model flux vector

$$\mathbf{F}_i^m(\tilde{\mathbf{U}}) = \begin{pmatrix} 0 \\ \tau_{1i}^m \\ \tau_{2i}^m \\ \tau_{3i}^m \\ \tau_{ij}^m \tilde{u}_j - q_i^m \end{pmatrix}, \quad (5.15)$$

with τ_{ij}^m the SGS stress tensor, and q_i^m the SGS heat transfer.

The VMS method employing the constant-coefficient Smagorinsky model [73] has proven to be successful in prior VMS implementations [30, 31, 35, 61, 62, 63, 64]. Therefore, as a first investigation of the present VMS method, this model is also used here. Since the model acts directly on the small scales, the SGS stress tensor is defined as

$$\tau_{ij}^m = 2 \mu_t \left(\tilde{S}_{ij} - \frac{\delta_{ij}}{3} \tilde{S}_{kk} \right), \quad \mu_t = (C_s \Delta)^2 \rho |\tilde{S}|, \quad (5.16)$$

where μ_t is the eddy-viscosity coefficient, and $\tilde{S}_{ij} = \frac{1}{2}(\tilde{u}_{i,j} + \tilde{u}_{j,i})$ is the symmetric part of the small-scale velocity gradient tensor, C_s is the Smagorinsky constant, Δ is a length scale representative of the unresolved scales, and $|\tilde{S}| = (2\tilde{S}_{ij}\tilde{S}_{ij})^{1/2}$. Note that we have only used the small-scale solution in the SGS model, which therefore corresponds to the so-called small-small variant of the Smagorinsky model [29]. Note that if the small-scale components in (5.16) are replaced by all the resolved scales, the classical Smagorinsky model is recovered. The SGS heat flux in (5.15) is computed from

$$q_i^m = \kappa_t \frac{\partial \tilde{T}}{\partial x_i}, \quad \kappa_t = \frac{\mu_t c_p}{\text{Pr}_t}, \quad (5.17)$$

where $\text{Pr}_t = 0.9$ is the turbulent Prandtl number [69].

The parameters C_s and Δ of the Smagorinsky model may be determined by a procedure due to Lilly [46] (see Hughes *et al.* [29] for a brief overview). In Lilly's analysis turbulent kinetic energy production and dissipation are balanced. In the

inertial subrange, the spectral amplitude of the turbulent kinetic energy can be described by

$$E(k) = \alpha \epsilon^{2/3} k^{-5/3}, \quad (5.18)$$

where α is the Kolmogorov constant, ϵ is the kinetic energy dissipation, and k is the radius of the wavenumber vector in wavenumber space. For a traditional LES formulation with the Smagorinsky model acting directly on all resolved scales, $|S|$ is derived from the relation

$$\frac{1}{2}|S|^2 = \int_0^{k_c} k^2 E(k) dk, \quad (5.19)$$

where k_c is the resolution limit, such as the cut-off wavenumber in a spectral method. Note that since the model acts on all the scales, the integration interval starts from zero. Setting the turbulent kinetic energy production equal to the dissipation leads to an expression for the length scale of the Smagorinsky model when used in a traditional LES formulation

$$C_s \Delta = \left(\frac{2}{3\alpha} \right)^{3/4} k_c^{-1}. \quad (5.20)$$

Assuming $\Delta = h = \pi k_c^{-1}$, with h the mesh spacing, leads to a value of $C_s \approx 0.18$. This value has proven satisfactory for homogeneous isotropic turbulent flows. However, in the presence of mean shear, such as that present in turbulent channel flows, lower values such as $C_s = 0.1$ are often required [11]. In order to obtain the Smagorinsky coefficient C_s for the VMS method, Lilly's analysis can again be employed, using the small-scale counterpart of (5.19)

$$\frac{1}{2}|\tilde{S}|^2 = \int_{\tilde{k}}^{k_c} k^2 E(k) dk, \quad (5.21)$$

where \tilde{k} is the wavenumber that separates the large scales from the small, as shown in Figure 5.1. A similar analysis gives the following relation

$$\tilde{C}_s \Delta = \left(\frac{2}{3\alpha} \right)^{3/4} \tilde{k}^{-1} [(k_c/\tilde{k})^{4/3} - 1]^{-3/4}, \quad (5.22)$$

where the tilde in \tilde{C}_s is used to indicate that the model is acting on the small scales only. If an appropriate value for C_s is found then values for \tilde{C}_s follow from the ratio \tilde{C}_s/C_s given by

$$\tilde{C}_s/C_s = (k_c/\tilde{k})[(k_c/\tilde{k})^{4/3} - 1]^{-3/4}. \quad (5.23)$$

In principle, the length scale in the Smagorinsky model should vary in different flow regimes. When approaching a solid wall, the turbulent kinetic energy decreases until it reaches the viscous sublayer where the flow is no longer turbulent. Thus the value of the eddy-viscosity used in the outer region is no longer appropriate. To account for this effect, Kim *et al.* [39] have proposed a modification of the length scale ($\ell = C_s\Delta$) in the Smagorinsky model with a van Driest wall-damping function, such that the length scale becomes a function of the distance to the wall

$$\ell = \ell(y^+) = C_s\Delta \left(1 - e^{-y^+/A^+}\right), \quad (5.24)$$

where y^+ represents the non-dimensional distance normal to the wall (see Section 5.2), and $A^+ = 25$ is the van Driest constant. Note that with van Driest wall damping the length scale, and therefore the SGS stress tensor, goes to zero at the wall. The length scale Δ in (5.24) is computed from

$$\Delta = \left(\frac{\Delta_1}{P_1} \frac{\Delta_2}{P_2} \frac{\Delta_3}{P_3}\right)^{1/3}, \quad (5.25)$$

where Δ_i represents the element size in the i^{th} direction and P_i is the polynomial degree of the finite-element approximation in the i^{th} direction. This formulation takes both mesh size and the polynomial degree into account.

5.1.2 Implementation

As discussed in Section 3.3.1, in the present VMS formulation we base the partition of the resolved scales into large and small scales on the polynomial degree. The large scales are then described by the basis functions with low polynomial degrees, while the

small scales are described by the remaining basis functions with higher polynomial degrees. We do not consider temporal scale separation, so that in the following scale decomposition refers to spatial directions only. In three dimensions, the spatial component of the solution approximation on a master element ($-1 < \xi_1, \xi_2, \xi_3 < 1$) takes the form

$$\mathbf{Y}^e(\xi_1, \xi_2, \xi_3) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \sum_{r=0}^{P_3} \phi_{pqr}(\xi_1, \xi_2, \xi_3) \mathbf{y}_{pqr} \quad (5.26)$$

where P_1 , P_2 and P_3 are the polynomial degrees of the basis in each spatial direction. Note that the maximum indices, $p = P_1$, $q = P_2$ and $r = P_3$, correspond to linear vertex modes, not to be confused with modes of degree P_1 . We define the elemental basis functions representing the small scales within the VMS formulation as

$$\phi_{pqr}(\xi_1, \xi_2, \xi_3) = \tilde{\phi}_{pqr}(\xi_1, \xi_2, \xi_3) \quad \text{if} \quad \begin{cases} \tilde{P}_1 \leq p < P_1 & \text{and/or} \\ \tilde{P}_2 \leq q < P_2 & \text{and/or} \\ \tilde{P}_3 \leq r < P_3 \end{cases} \quad (5.27)$$

where \tilde{P}_1 , \tilde{P}_2 , and \tilde{P}_3 define the large/small partition in each of the coordinate directions. Note that whenever a basis function ϕ_{pqr} contains a small-scale component in one of the coordinate directions, it is considered a small-scale basis function. In this work we only consider equal polynomial degrees in all spatial directions, and we define $\tilde{P}_1 = \tilde{P}_2 = \tilde{P}_3 = \tilde{P}$. Having defined the small-scale basis functions leaves the large-scale basis functions as the remaining modes

$$\bar{\phi}_{pqr} = \{\phi_{pqr} | \phi_{pqr} \neq \tilde{\phi}_{pqr}, 0 \leq p, q, r \leq P\} \quad (5.28)$$

With the previous definitions the elemental large and small-scale solution approximations can be written as

$$\bar{\mathbf{U}}(\bar{\mathbf{Y}}^e) = \bar{\mathbf{U}} \left(\sum_{a=1}^{n_L} \bar{N}_a^e (\pi_n^-(t) \bar{\mathbf{y}}_{a;(n)} + \pi_n^+(t) \bar{\mathbf{y}}_{a;(n+1)}) \right), \quad (5.29)$$

$$\tilde{\mathbf{U}}(\tilde{\mathbf{Y}}^e) = \tilde{\mathbf{U}} \left(\sum_{a=1}^{n_S} \tilde{N}_a^e (\pi_n^-(t) \tilde{\mathbf{y}}_{a;(n)} + \pi_n^+(t) \tilde{\mathbf{y}}_{a;(n+1)}) \right), \quad (5.30)$$

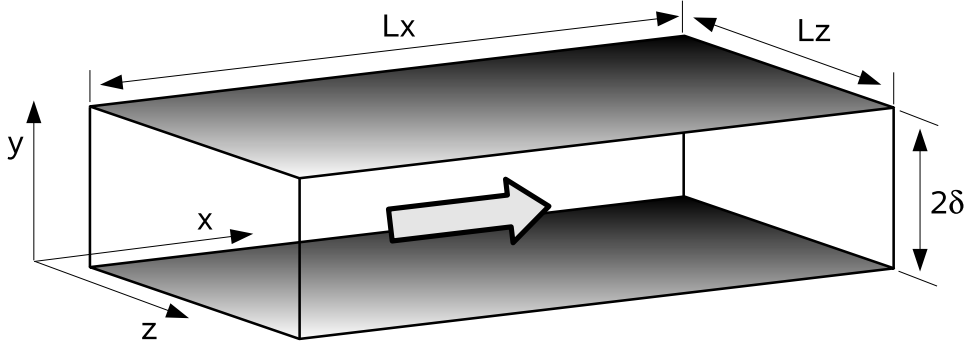


Figure 5.2: Planar channel flow geometry. The arrow indicates the direction of the mean flow, and the shaded regions represent the channel walls.

where n_L and n_S are the number of spatial large and small-scale elemental basis functions respectively, \bar{N}_a^e and \tilde{N}_a^e are the a^{th} large and small-scale elemental basis functions corresponding to $\bar{\phi}_{pqr}$ and $\tilde{\phi}_{pqr}$ such that each index a represents a unique combination of p, q and r corresponding to a large (5.28) or small scale (5.27) basis function, and \bar{y}_a and \tilde{y}_a are the large and small-scale solution coefficients.

5.2 Turbulent channel flow

5.2.1 Problem description

Planar channel flow has been a widely used test case for the research of turbulence in the presence of mean shear. The flow is assumed to be passing between two horizontal plates separated by a distance 2δ as shown in Figure 5.2, where $x = x_1$ is the streamwise direction, $y = x_2$ is the wall-normal direction, and $z = x_3$ is the spanwise direction. Results for wall bounded turbulent flows are often presented in wall-units, defined as $x_i^+ = x_i u_\tau / \nu$ and $u^+ = u / u_\tau$, where $\nu = \mu / \rho$, and $u_\tau \equiv (\tau_w / \rho)^{1/2}$ is the friction velocity, with τ_w the average wall shear stress.

5.2.2 Physical aspects

Both experiments and dimensional reasoning have indicated the existence of three distinct layers in the wall-normal direction, known as the viscous sublayer, the buffer layer, and the logarithmic layer. The inner-most layer or viscous sublayer ($0 < y^+ < 5$) is a region where turbulent effects are minimal and viscous forces dominate. The velocity profile in this region can be expressed by the linear profile $u^+ = y^+$. Above this region, the turbulent and viscous effects are of similar order of magnitude. This is known as the buffer layer, which extends roughly from $y^+ = 5$ to 30. Beyond this region comes the logarithmic layer in which the mean velocity profile can be described by $u^+ = 1/k \ln y^+ + B$, where $k = 0.4$ is the von Karman constant and $B \approx 5.5$ is a constant determined by the intersection with the buffer layer.

Numerous physical and numerical experiments [39, 41] have shown the existence of *low-speed streaks*, which are regions in the near-wall vicinity of the flow where the velocity is relatively low compared to their direct surrounding (Figure 1.2). Near the wall, streaks consist of well-defined structures independent of the Reynolds number, having a mean streamwise spacing of $\Delta x^+ \approx 400$ wall-units, and a mean spanwise separation of $\Delta z^+ \approx 100$ wall-units [11].

Near-wall streaks fall into the category of so-called coherent structures [58], which play an important role in the dynamics of wall-bounded turbulence. In particular, near-wall streaks are responsible for a sequence of sweep and ejection events, that bring higher-momentum flow towards the wall, and at the same time lift lower-momentum flow to the outer layer. It is this process which is believed to be responsible for the self-sustained process of turbulent energy production in wall-bounded flow.

Near-wall streaks also play an important role in the analysis of the present numerical results. Prior investigations of the VMS method using a Galerkin Fourier-spectral method [11, 63] have demonstrated a close relation between the large/small partition and the typical size of the near-wall streaks. By applying the VMS scale decom-

position only in one of the coordinate directions, it was shown that the large/small partition must be chosen such that the large-scale space includes the minimum range of scales required to capture the dynamically important features of the near-wall coherent structures. It was concluded that in the streamwise direction the critical wavelength is $\Delta x^+ \approx 200$, which corresponds to the length over which the spanwise and wall-normal velocities become decorrelated. In the spanwise direction the critical wavelength is $\Delta z^+ \approx 50$, which corresponds to the mean spanwise spacing of the streaks. It has been found that spanwise scale separation in the VMS method leads to sharper estimates of the required partition [11]. This is due to the near-wall streaks being more consistently spaced in the spanwise direction than in the streamwise direction. As discussed by [62], the solution quality for LES [8] and DNS [39] is most sensitive to the spanwise resolution. Therefore, in the following when considering near-wall streaks we focus on the spanwise direction.

5.2.3 Boundary conditions and mesh geometry

We assume the flow to be homogeneous in horizontal planes parallel to the walls, so that periodic boundary conditions can be applied in the streamwise and spanwise directions. As a consequence of the periodic boundary condition in the streamwise direction, the flow cannot be driven by a pressure gradient. Therefore, we apply an external body force, which mimics the role of the pressure gradient. It can be shown [54] that for a horizontal homogeneous turbulent channel flow to be in equilibrium

$$\frac{dp}{dx} = -\frac{\rho u_\tau^2}{\delta}, \quad (5.31)$$

so that the pressure gradient is balanced by the shear stress at the walls. Note that, if the reference length scale and velocity are chosen to be δ and u_τ the non-dimensional pressure gradient becomes unity. Two procedures are generally encountered in literature concerning the forcing of turbulent channel flow. The most simple one is applying a constant forcing term equal to the pressure gradient. An alternative ap-

proach is to keep the mass flow constant by dynamically adjusting the forcing term. The latter is more popular, since it leads to shorter transients in which the turbulence develops before reaching the statistically-stationary regime [56, 79].

In order to increase the resolution in the vicinity of the wall, we use a stretched mesh in the wall-normal direction, such that

$$y_j = \frac{\tanh(a_s(2j/N_y - 1))}{\tanh(a_s)} + 1, \quad j = 0, \dots, N_y, \quad (5.32)$$

where N_y is the number of elements in the wall-normal direction and a_s is a stretching factor, set equal to $a_s = 1.75$.

At the solid walls we apply the no-slip condition for the velocity components and a constant wall temperature. The latter allows internal energy created by viscous dissipation to be removed from the domain via heat transfer, so that a statistically steady state can be achieved.

5.2.4 Initial condition

All turbulent channel flow simulations were started using an incompressible DNS solution (obtained from an external code) as the basis for the initial condition. A compressible density and pressure field was obtained using an isentropic flow assumption. To do so, we assume all streamlines come from the same reservoir with total density, pressure and speed of sound given by

$$\rho_0 = \rho_{res} \left[1 + \frac{\gamma - 1}{2} M_{res}^2 \right]^{1/(\gamma-1)}, \quad (5.33)$$

$$p_0 = p_{res} \left[1 + \frac{\gamma - 1}{2} M_{res}^2 \right]^{\gamma/(\gamma-1)}, \quad (5.34)$$

$$a_0 = (\gamma p_0 / \rho_0)^{1/2}, \quad (5.35)$$

where ρ_{res} and p_{res} are the static density and pressure of an “imaginary” reservoir and M_{res} is the Mach number of the flow in the reservoir. When the total quantities

are known, the local density and pressure are calculated using the inverse relations

$$\rho = \rho_0 \left[1 + \frac{\gamma - 1}{2} \mathbf{M}^2 \right]^{1/(1-\gamma)}, \quad (5.36)$$

$$p = p_0 \left[1 + \frac{\gamma - 1}{2} \mathbf{M}^2 \right]^{\gamma/(1-\gamma)}, \quad (5.37)$$

where $\mathbf{M} = \|bu\|/a$ is the local Mach number, with a the local speed of sound given by

$$a = \left[a_0^2 - \frac{\gamma - 1}{2} V^2 \right]^{1/2}. \quad (5.38)$$

When the local density and pressure are known, the temperature can be computed from the perfect gas law (2.5).

Finally, in order to obtain the solution coefficients for the finite-element basis that is employed in the simulation, the initial compressible solution is projected onto the finite-element space used for the LES. Since this solution space is inherently coarse relative to that of the DNS, information will be lost. This is specifically noticeable as reduced turbulence intensities. Moreover, since we consider the compressible Navier-Stokes equations, a transient will develop in which the variables adjust to a turbulent state. After this transient phase the flow becomes statistically stationary. This is illustrated by the history of the average shear stress on the walls, as shown in Figure 5.3. From the statistically stationary regime the turbulence statistics can be computed, which is described in the next section.

5.2.5 Turbulence statistics

Turbulent flows are generally studied in terms of statistical quantities which give insight into the dynamics of the flow. To generate a statistical picture of the flow, it is assumed that instantaneous variables can be decomposed into an average plus a fluctuation, commonly known as the Reynolds decomposition. For example, the velocity components can be written as

$$u_i = \langle u_i \rangle + u'_i, \quad (5.39)$$

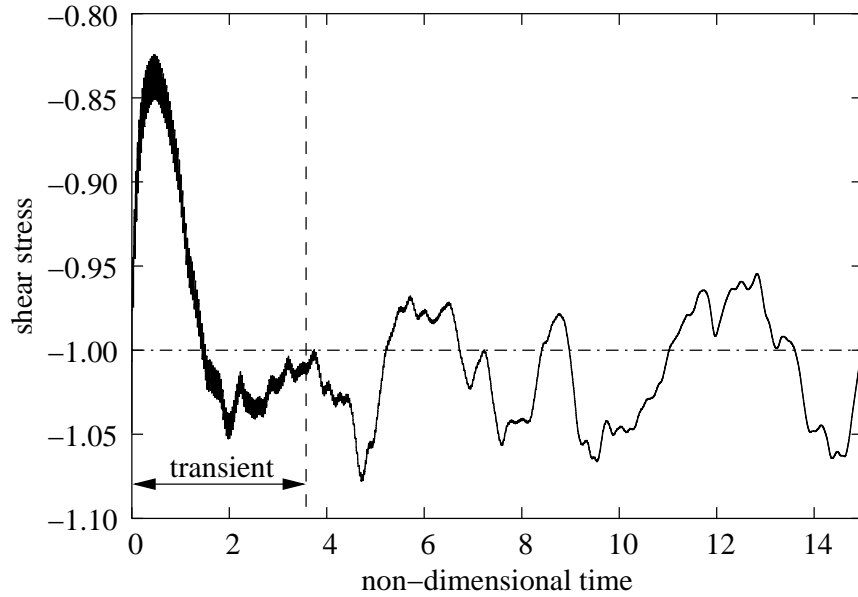


Figure 5.3: Shear stress history for turbulent channel flow at $\text{Re}_\tau = 180$, averaged over the upper and lower walls of the channel.

where the brackets denote the average or mean value, and the prime denotes the fluctuation. The averaged quantity is usually defined as a time average, given by

$$\langle u_i \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} u_i(t) dt, \quad (5.40)$$

where T is the time interval over which the average is computed. In numerical computations, this average is usually computed by performing an average over N discrete time steps, i.e.

$$\langle u_i \rangle = \frac{1}{N} \sum_{n=1}^N u_i(t_n), \quad (5.41)$$

where $u_i(t_n)$ is the numerical solution at time level t_n . In addition to temporal averaging, spatial averaging can be performed over homogeneous directions for flows that have such character. This increases the number of statistical samples lowering the number of time steps required to achieve statistically-steady behavior of the flow quantities of interest. Therefore, in turbulent channel flow, spatial averaging is

performed over the homogeneous xz -planes, i.e. in stream- and spanwise directions.

The difference in computing the statistics for a hierarchical basis compared to a Lagrangian basis is that, with the exception of the vertex modes, the solution coefficients do not correspond directly to a local value of the solution in space. Therefore, simply collecting and averaging the coefficients does not work when a hierarchical basis is used. The computation of statistics thus requires the evaluation of the interpolation functions for reconstructing the solution at the location of interest.

In case of the channel flow, spatial averaging of a quantity of interest at a given wall-normal coordinate is performed on two-dimensional structured sampling grids. In particular, the quantities of interest are computed at sampling points on a horizontal plane and averaged to a single point on a line in the wall-normal direction. Next, the temporal averages are computed from (5.41), with u_i corresponding to the spatial averages. As a final step, the number of samples are doubled by averaging over the upper and lower half of the channel.

5.3 Numerical results

In the present simulations, we choose the channel half-height δ as the reference length, and the friction velocity u_τ as the reference velocity. These reference quantities gives rise to a reference Reynolds number $\text{Re}_\tau = u_\tau \delta / \nu$.

As an initial investigation of the proposed VMS formulation, we perform simulations at $\text{Re}_\tau = 180$. The centerline Mach number was set to $\text{M} = 0.3$, so that the results can be directly compared [13, 14, 61] to the incompressible DNS data from Kim *et al.* [39]. The size of the computational domain is $(6\delta, 2\delta, 4\delta)$ in the x_1 , x_2 , and x_3 directions respectively. As shown by various investigations [31, 63, 78, 79], these dimensions are sufficiently large for this Reynolds number.

In order to minimize the effects of the time discretization, we employ the most accurate method in the simulations presented here, that is the TDG method (see

Section 3.2.3). All computations were performed with time-step size $\Delta t = 0.002$, which is comparable to that used by other authors employing implicit discretizations of the *incompressible* Navier-Stokes equations [61, 79]. (The effect of the time step on the low-order turbulence statistics will be investigated in Chapter 6.) For an explicit time discretization, the use of a compressible formulation leads to stricter convective stability limits compared to an incompressible formulation. This is due to the finite speed of sound in a compressible flow leading to higher convective speeds. For the present implicit scheme, however, all simulations performed for this section remained numerically stable. The capability of using relatively large time-step sizes is a clear advantage of the proposed time discretization.

All simulations were performed using primitive density variables, as defined in Appendix A. The convergence criteria for the non-linear solution algorithm correspond to $\varepsilon_{update} = \varepsilon_{res} = 0.001$, so that both the relative update and relative residual are checked for non-linear convergence. The convergence criterion for the solution of the linear system is set to $\varepsilon_{gmres} = 0.001$. The effects of these convergence criteria on the accuracy and cost of the simulation are investigated in Chapter 6.

In the following, we compare results obtained for a given *hp*-resolution using

- (i) no SGS model,
- (ii) using the SGS model on all resolved scales, which is referred to as LES,
- (iii) using the SGS model only on the small resolved scales, which is referred to as VMS.

All results for LES employ van Driest wall damping, while the VMS methods do not employ van Driest wall damping. The Smagorinsky constant was set to $C_s = 0.1$ for all computations, as this value has proven successful in prior VMS computations [11, 14, 23, 29, 30, 63, 64]. Here, we consider the following low-order turbulence statistics:

- mean streamwise velocity denoted by U ,
- root-mean-square (rms) of the velocity fluctuations denoted by $u_{rms}, v_{rms}, w_{rms}$, defined as $\langle u'^2 \rangle^{1/2}, \langle v'^2 \rangle^{1/2}, \langle w'^2 \rangle^{1/2}$,
- Reynolds stress, defined as $\langle -u'v' \rangle$,
- viscous stress, defined as $(1/\text{Re})dU/dy$,
- total stress, defined as the sum of the Reynolds and viscous stresses,
- One-dimensional mean energy spectra E_{ii} , defined as the discrete Fourier transform of the square of the velocity components u_i .

5.3.1 Coarse mesh results

We begin with an $(8 \times 8 \times 8)$ mesh with uniform polynomial degree $P = 2$. This h -resolution corresponds to a spacing of $\Delta x^+ = 135$ and $\Delta z^+ = 90$ in the streamwise and spanwise directions, and $\Delta y^+ \approx 14.6$ in the wall-normal direction. The simulation parameters are summarized in Table 5.1, including the total number of degrees of freedom per variable as well as the fraction of modes in the large and small-scale space, denoted by f_{large} and f_{small} for the VMS simulations.

The mean flow and rms profiles at this resolution are shown in Figure 5.7. Note that for this polynomial degree there is only one possibility for the VMS partition, that is $\tilde{P} = 2$. This means that the large scales are represented by the linear modes in each direction (see Section 5.1). The computed mean velocity profiles are shown in Figure 5.7(a). All results are in poor agreement with the DNS [39]. The no-model case overpredicts the wall-shear stress, as can be seen from the mean velocity profile that is low relative to the DNS profile. On the other hand, the LES and VMS results are very similar, both underpredicting the wall-shear stress. Now, consider the rms profiles for the above cases, as shown in Figure 5.7(b). The u -component of the no-model case is in reasonable agreement with the DNS [39], however both the v and

w components are significantly overpredicted. The LES is excessively dissipative, as can be seen from the spanwise and wall-normal components. Moreover, the peak in the streamwise rms is shifted away from the wall. The VMS results have a significant overprediction of the streamwise component, while the remaining components are in reasonable agreement with the DNS [39].

The Reynolds stress profiles are shown in Figure 5.8(a). The no-model case is slightly oscillatory and overpredicts the peak value. As expected, the LES severely underpredicts the Reynolds stress due to excessive dissipation inherent in the model. Meanwhile, the VMS is in reasonable agreement with the DNS [39]. The viscous stress profiles, shown in Figure 5.8(b), all deviate from the DNS [39] for approximately $y^+ < 0.25$. Moreover, the total stress shows oscillatory behavior. This occurs because we are trying to resolve large gradients close to the wall with a relatively coarse resolution, without using additional stabilization mechanisms such as SUPG [4] or GLS [71]. Recall that the near-wall resolution corresponds to $\Delta y_w^+ \approx 14.6$ based on the element size. Note that this value is significantly larger than is normally used in LES, that is $\Delta y_w^+ < 1.0$ [61].

Finally, compare the one-dimensional energy spectra in the streamwise and spanwise directions at $y^+ = 5.39$ shown in Figure 5.9 to that of the DNS [39]. The energy content in both spectra for the no-model case is significantly higher than that of the DNS [39]. This excess energy is mainly a result of the absence of viscous dissipation scales due to the coarse resolution. Consequently, when no model is employed to account for these missing scales, one can expect an energy content which is larger than usual. On the other hand, applying the Smagorinsky model to all the resolved scales, as in LES, leads to an energy content which is significantly lower than that of the DNS [39]. Overall, the large-scale spectra obtained using the VMS method are closest to the DNS [39]. Clearly, the coarse mesh resolution is too coarse for obtaining reasonable low-order statistics. Moreover, in both the LES and VMS results, the basic dynamics of the important coherent structures, i.e. the near-wall streaks, are

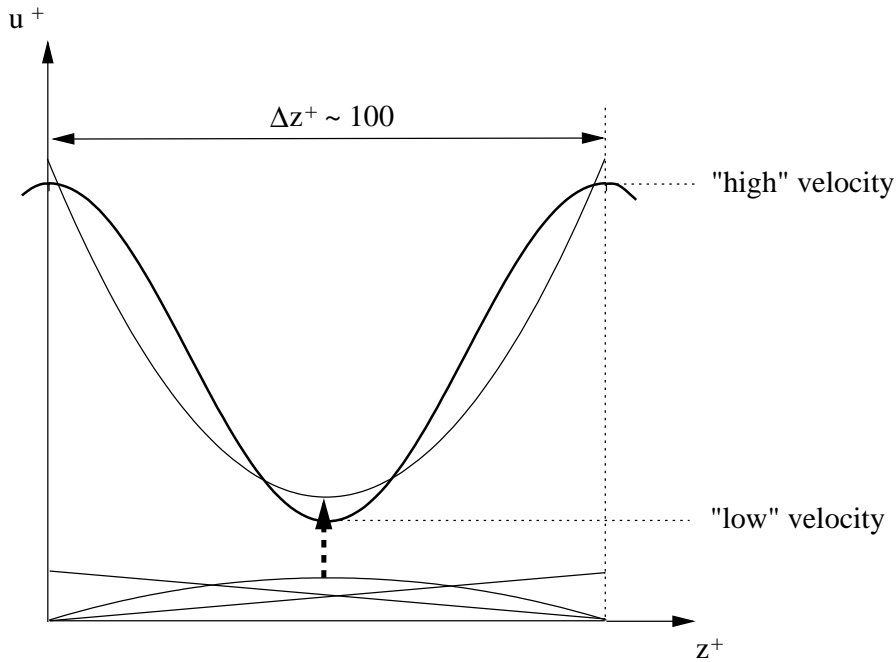


Figure 5.4: Illustration of the near-wall streamwise velocity distribution in spanwise direction due to low-speed streaks, relatively to a spanwise element size approximately equal to the streak separation. The modes at the bottom represent the finite-element basis functions.

directly influenced by the SGS model. This can be seen as follows. The spanwise element size ($\Delta z^+ = 90$) is approximately equal to the mean streak separation in spanwise direction. This means that the streamwise velocity profile in the spanwise direction varies from a high to a low and back to a high value again within one element, as sketched in Figure 5.4. Note that such variation can not be captured by the large-scale solution, which consist of only linear basis functions. The quadratic modes do have the capability of capturing such alternating velocity variation within the element, however, these modes are contained within the small-scale space. Consequently, the SGS model is acting directly on those scales which describe this basic feature.

5.3.2 Wall-normal h-refinement

The arguments of the previous section imply that an increased resolution in the wall-normal direction only will not significantly improve accuracy. To verify this, we employ an $(8 \times 16 \times 8)$ mesh with polynomial degree $P = 2$. This corresponds to $\Delta y^+ \approx 5.9$ in the wall-normal direction, while the resolution in the homogeneous directions is equivalent to that of the previous case. Therefore, when the VMS method is used, the large-scale structures are still directly affected by the SGS model.

The mean velocity profiles for the different model options, shown in Figure 5.10(a) are very similar to those obtained using the $(8 \times 8 \times 8)$ mesh. This is also the case for the rms profiles shown in Figure 5.10(b) and the Reynolds stress shown in Figure 5.11(a). Inspection of the viscous stresses shown in Figure 5.11(b) also shows very similar results. Note that the oscillatory behavior of the total stress is less severe than that of the coarser wall-normal resolution, indicating that it is indeed a coarse-resolution artifact. Finally, consider the streamwise and spanwise energy spectra shown in Figure 5.12. Again, similar trends are observed as for the coarser resolution.

Thus, increasing the resolution only in the wall-normal direction does not lead to a significant increase in accuracy. This is also to be expected since a turbulent flow is inherently three-dimensional. Especially at these coarse resolutions, one cannot expect that increasing the resolution in one direction leads to significantly improved representations of the physically important structures in the flow.

5.3.3 Uniform h-refinement

Next, we increase the resolution by refining the mesh in all three directions, i.e. we consider a $(16 \times 16 \times 16)$ mesh with $P = 2$. This mesh corresponds to a streamwise spacing of $\Delta x^+ = 67.5$, a spanwise spacing of $\Delta z^+ = 45$, and a wall-normal spacing of $\Delta y^+ \approx 5.9$. Before presenting the results we note that this is an interesting case for the VMS method (with $\tilde{P} = 2$). The element size in the spanwise direction now

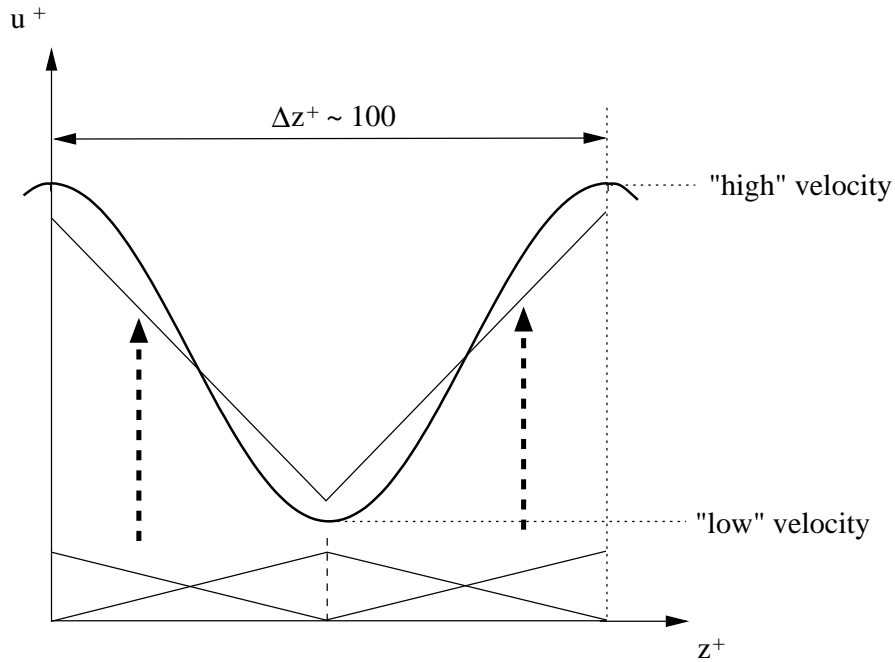


Figure 5.5: Illustration of the near-wall streamwise velocity distribution in spanwise direction due to low-speed streaks, relatively to a spanwise element size approximately equal to half the streak separation. The modes at the bottom represent the finite-element basis functions.

corresponds to half the typical streak spacing, or the streak size. As illustrated in Figure 5.5, the alternating spanwise velocity variation typical of near-wall streaks can now be captured by the linear modes on two neighboring elements. Since the linear modes are contained within the large-scale space, this feature of the near-wall streaks is not directly influenced by the SGS model. In Fourier analysis terms, only the lowest wavenumber component of the alternating velocity distribution is free from direct modeling terms.

The mean velocity profiles are shown in Figure 5.13(a). The mean flow profile obtained using the VMS method is in very good agreement with the DNS [39], it cannot be distinguished at this scale. The LES is in reasonable agreement with the DNS [39], while the no-model case still overpredicts the wall shear stress. The

rms profiles, shown in Figure 5.13(b), are best predicted by the VMS method. The Reynolds stresses shown in Figure 5.14(a), exhibit a similar trend. The Reynolds stress profiles of the no-model case and LES are comparable to those obtained at lower resolutions, while that of VMS method is very close to the DNS [39]. Likewise, the viscous and total stresses shown in Figure 5.14(b), are best predicted by the VMS method. Finally, the streamwise and spanwise energy spectra are shown in Figure 5.15. The spectra obtained using the VMS method are in reasonable agreement with the DNS [39] in the large scales, especially the streamwise energy spectrum (Figure 5.15(a)). In contrast, the energy content of the no-model case is again significantly higher than that of the DNS [39], indicating insufficient dissipative mechanisms. The spectra of the LES indicate excessive model dissipation as the energy content is considerably lower than that of the DNS [39].

Figure 5.6 shows an instantaneous contourplot of the streamwise velocity in a horizontal in the near-wall vicinity of the flow, obtained from the VMS calculation. This figure clearly demonstrates that the near-wall streaks are captured, and confirms the typical streak spacings in streamwise ($\Delta x^+ \approx 400$) and spanwise direction ($\Delta z^+ \approx 100$).

Remarkably, leaving just the lowest wavenumber component of the near-wall streak velocity distribution free from direct modeling terms leads to a significant increase in the quality of the low-order statistics. On the other hand, applying the model to all scales, such as in traditional LES formulations, leads to results that suffer from excessive dissipation and are qualitatively similar to those of coarser resolutions.

5.3.4 Uniform p-refinement

In this section we investigate the effect of SGS modeling at a higher polynomial degree for an $(8 \times 8 \times 8)$ mesh with uniform polynomial degree $P = 3$. Recall from Section 5.3.1 that the spanwise element size corresponds to the mean streak separation. Based

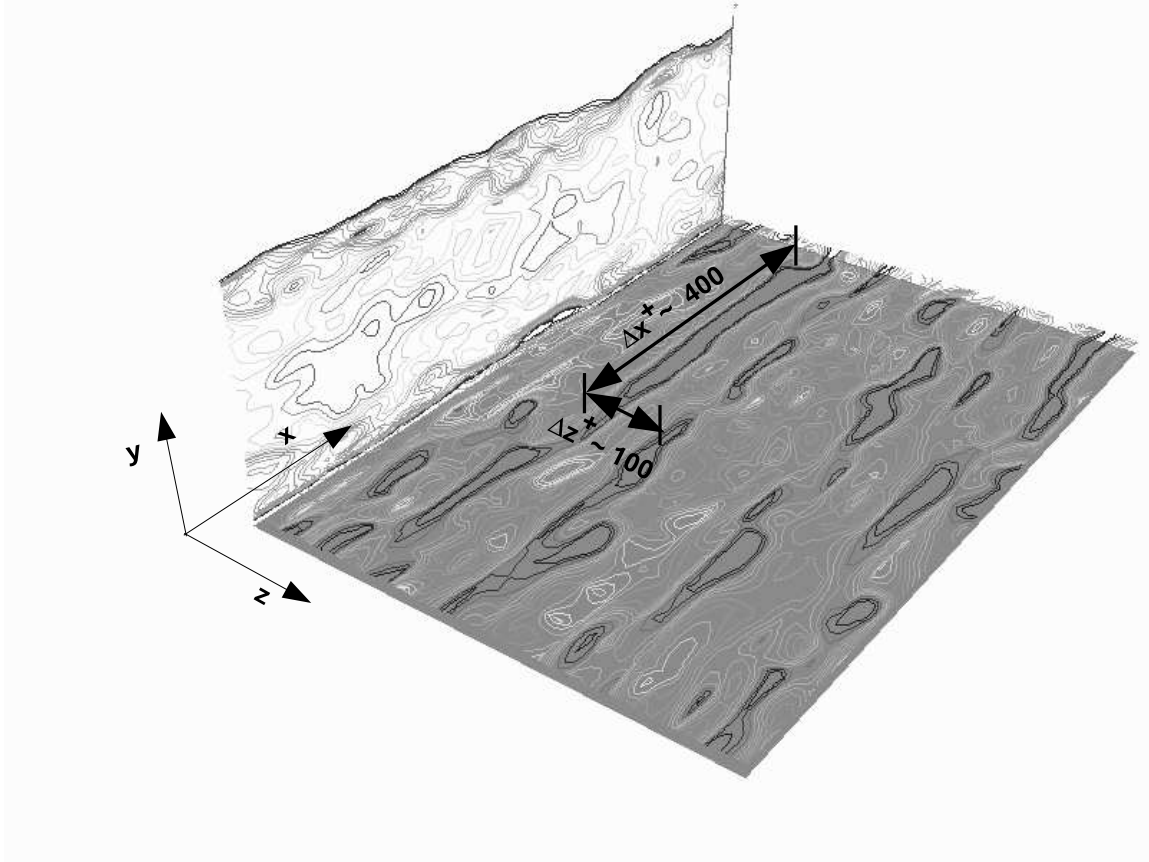


Figure 5.6: Instantaneous streamwise velocity contours in a turbulent channel flow at $\text{Re}_\tau = 180$, obtained from a VMS calculation on an $(16 \times 16 \times 16)$ mesh with $P = 2$ and $\tilde{P} = 2$.

on previous observations, we begin with a partition parameter $\tilde{P} = 3$, so that the linear and quadratic modes are in the large scale space and all modes that contain a cubic term are considered small scales. This ensures that the quadratic modes, which are capable of capturing the basic streak velocity variation within an element (see Figure 5.4), are not directly influenced by the SGS model.

The mean velocity profiles for the cases under consideration are shown in Figure 5.16(a). The mean flow profile obtained with the VMS method is in excellent agreement with the DNS [39]. In fact, they cannot be distinguished at this scale. The

no-model case significantly overpredicts the wall-shear stress, while the LES does not exhibit the correct logarithmic behavior. Now, consider the rms profiles shown in Figure 5.16(b). Again, the best agreement with the DNS [39] for all components of the turbulence intensities is obtained using the VMS method. As observed in the mean velocity profile, the remaining models show poor agreement with the DNS [39], similar to that observed for the lower polynomial degree. The LES is clearly over-dissipative as seen from the spanwise and wall-normal components, while the peak in the streamwise component is again shifted away from the wall. Meanwhile, the no-model case significantly overpredicts the turbulence intensities in the spanwise and wall-normal direction. Likewise, the Reynolds stress, viscous stress and total stress, shown in Figure 5.17, obtained using the VMS method are closest to that of the DNS [39]. The no-model case and LES are again qualitatively similar to that obtained using lower polynomial degree. Finally, we consider the energy spectra shown in Figure 5.18. The VMS model shows remarkably good agreement with the DNS [39] in the large scales for both the streamwise and spanwise energy spectra. Overall, the energy content obtained by the no-model case is too large compared to the DNS [39]. As expected, the LES is excessively dissipative leading to an overall energy content that is too low in comparison with the DNS [39].

Choosing large-scale space to capture the basic feature of the near-wall streaks, that is the alternating velocity variation in spanwise direction, proved a necessary condition in order to obtain reasonable low-order statistics using the VMS method. This ensured that modes corresponding to the lowest wavenumber components of the dynamically important scales are not contaminated by the SGS model. To further investigate this condition, we change the large/small partition for the $(8 \times 8 \times 8)$ mesh with $P = 3$ to $\tilde{P} = 2$. The modes containing quadratic and cubic terms were now contained in the small scale space. Recall from our previous discussion that the quadratic modes are the lowest order modes that are capable of capturing the streak velocity pattern within an element at this resolution. However, at this partition these

modes are directly affected by the model.

Next, we compare the results obtained using the VMS method using $\tilde{P} = 2$ to that of the VMS method with $\tilde{P} = 3$ and the LES method. As before, Figure 5.19 shows the mean flow and rms profiles, Figure 5.20 shows the Reynolds, viscous and total stresses, and Figure 5.21 shows the one-dimensional energy spectra in streamwise and spanwise directions. Overall, the results of the VMS method with $\tilde{P} = 2$ are in poor agreement with that of the DNS [39]. In fact, the results are of intermediate quality between the VMS method with $\tilde{P} = 3$ and the LES method. These results again emphasize the requirement that the modes representing the basic feature of the near-wall streaks must be left free from direct modeling terms.

5.3.5 Serendipity expansion

In this section we investigate if the efficiency in terms of degrees of freedom can be further increased by employing the serendipity basis, which consist of only those modes necessary for a given order of accuracy. We compare the results obtained using $(8 \times 8 \times 8)$ elements with a serendipity expansion for $P = 3$ and $P = 4$ to those employing the full tensor-product basis with $P = 3$. All results employ the VMS method with $\tilde{P} = 3$, which for the full basis showed a good agreement with the DNS [39] (see Section 5.3.4).

The mean velocity profiles are shown in Figure 5.22(a). Both results using the serendipity basis are in reasonable agreement with the DNS [39]. The serendipity basis for $P = 3$ is slightly less accurate than the full basis, which is indistinguishable from the DNS at this scale. Increasing the polynomial degree to $P = 4$ improves the mean velocity profile, however there is still a slight discrepancy with the DNS in the center of the channel.

The rms profiles shown are in 5.22(b). Both results obtained using the serendipity expansions are in very poor agreement with the DNS [39]. The turbulence intensities

are significantly overpredicted for all components, indicating insufficient dissipation. This is also confirmed by the one-dimensional energy spectra shown in Figure 5.23, from which a significant pile-up of energy is observed, especially at the large scales.

The reduced number of modes in the serendipity expansion leads to a reduction of both the large and small-scale space. Compared to the full basis, the serendipity basis misses all interior modes and various boundary modes having nonzero value along two and three-dimensional elemental regions. As a result of the diminished large-scale space the large energy containing structures cannot be resolved accurately. Likewise, the diminished small-scale space leads to reduced accuracy of the small scales, which in turn inappropriately affect the large scales through non-linear interactions. Additionally, the reduced small-scale space leads to insufficient model dissipation as clearly observed from the energy spectra. For all these reasons, the serendipity expansion does not produce reasonable turbulence statistics for the resolutions considered here.

These observations are consistent with those obtained for the convection-diffusion problem in Section 3.4.5. The modes that are discarded in the serendipity basis do have a contribution to the absolute accuracy of the solution. The results for turbulent channel flow, however, do not necessarily imply that the serendipity basis is less efficient in terms of degrees of freedom. Note that for a given number of degrees of freedom, the serendipity basis allows an increased number of elements and/or polynomial degree compared to the corresponding full basis. Moreover, increased polynomial degree offers additional freedom for the partition between large and small scales for the VMS method. On the other hand, an increased polynomial degree requires a higher quadrature order, which can seriously affect efficiency in terms of CPU time. We discuss the factors influencing the computational cost and accuracy in some more detail in Chapter 6. The capability of the serendipity is not further investigated in this work.

N_x	N_y	N_z	P	Δx^+	Δy^+	Δz_w^+	d.o.f.	\tilde{P}	f_{large}	f_{small}
8	8	8	2	135	14.6	90	7,680	2	0.12	0.88
8	8	16	2	135	5.9	90	15,872	2	0.12	0.88
16	16	16	2	67.5	5.9	45	63,488	2	0.12	0.88
8	8	8	3	135	14.6	90	26,496	3	0.29	0.71
8	8	8	3	135	14.6	90	26,496	2	0.03	0.97
8	8	8	3*	135	14.6	90	6,528	3	0.57	0.43
8	8	8	4*	135	14.6	90	9,344	3	0.70	0.30

Table 5.1: Simulation parameters for turbulent channel flow at $\text{Re}_\tau = 180$. The asterisk refers to the serendipity expansion.

5.4 Summary

We have defined an implementation of the variational multiscale method within the current hierarchical finite-element formulation. This formulation was investigated for fully-developed channel flow at $\text{Re}_\tau = 180$. The results obtained from the VMS method were compared to that using no SGS model and that using an SGS model on all the resolved scales as in a traditional LES formulation.

The results obtained using no SGS model suffer from insufficient dissipation, leading to artificially high energy content in the resolved scales, which are clearly observed in the corresponding one-dimensional energy spectra. This leads to an overprediction of the mean velocity profiles as well as rms profiles. These results indicate that explicit SGS modeling is required in order to obtain reasonable results. Applying the SGS model to all the resolved scales directly, structurally leads to excessive dissipation. In this case, the large energy containing scales are directly affected by the model, resulting in poor low-order statistics. On the other hand, in the VMS method the SGS model is only applied to the small resolved scales. Reasonable results can be obtained depending on the partition between the large and small scales and the size of

the large and small scale space. Regarding the large/small partition, it appears to be crucial for reasonable turbulence statistics that scales describing the basic features of the near-wall coherent structures are released from the direct influence of the model. Specifically, modes that are capable of representing the alternating spanwise velocity profile characteristic for near-wall streaks should be contained in the large-scale space.

We investigated the use of the serendipity basis for an hp -resolution that produced reasonable results using the full basis VMS method. Although the mean velocity was of reasonable quality, the rms profiles were in very poor agreement with the DNS [39]. This is due to the reduced number of modes in both the large and small scale spaces. As a result, the large energy-containing structures cannot be described sufficiently accurately, while the reduced small-scale space does not provide sufficient model dissipation.

We note that there are two features of global spectral VMS discretizations which are not available in the present VMS discretization. First, the present hierarchical basis is a non-orthogonal basis as opposed to that employed in global spectral methods. The use of an orthogonal basis leads to less terms which have to be modeled, and consequently modeling might be less complex [10]. Second, a perfect separation in scales is not possible in the present method, in which scale separation is based on polynomial degree. However, the results obtained using the present VMS method demonstrate that this potential disadvantage is not severe enough to prevent the method from producing reasonable turbulence statistics.

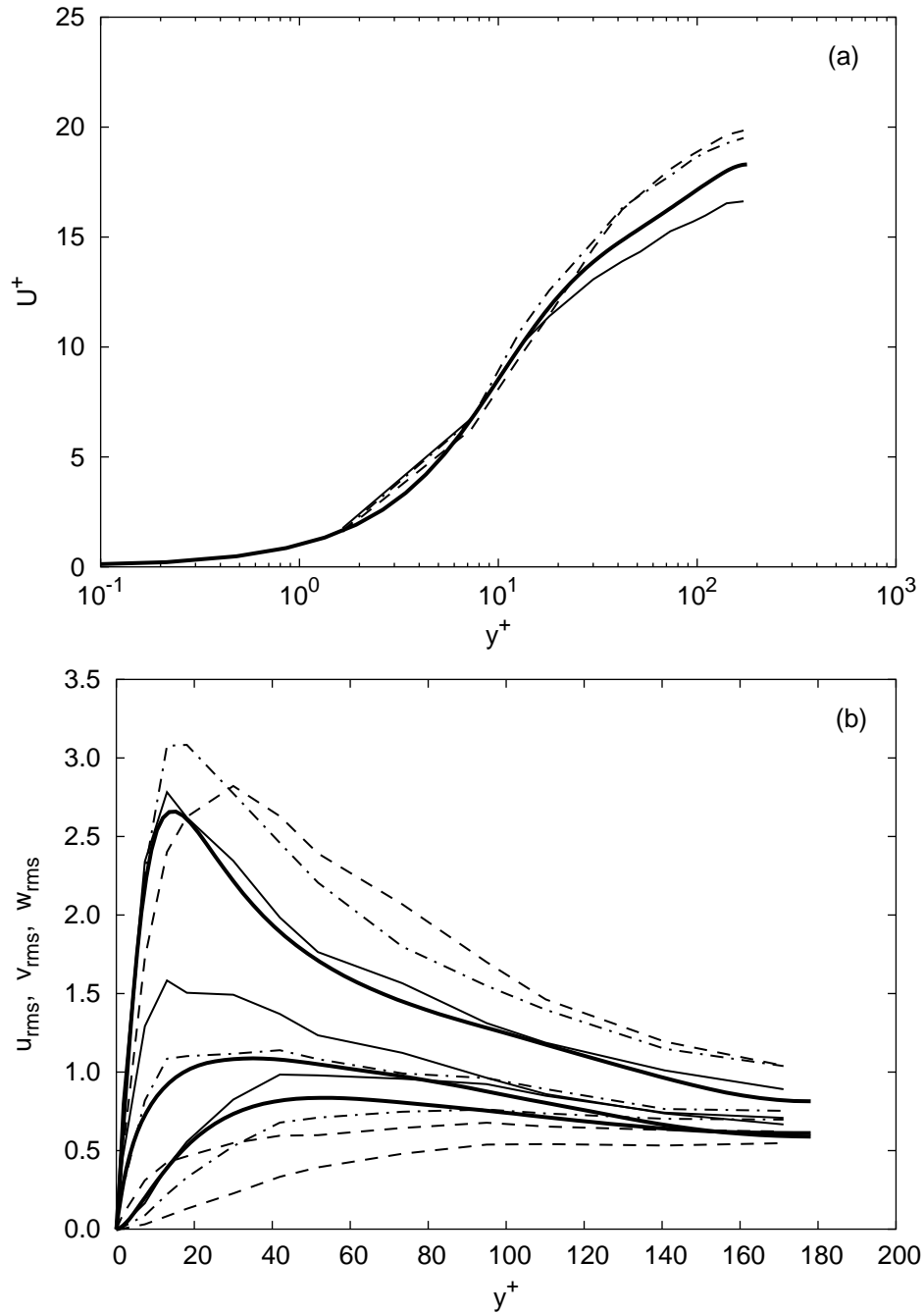


Figure 5.7: Comparison of the mean velocity and rms profiles for $Re_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

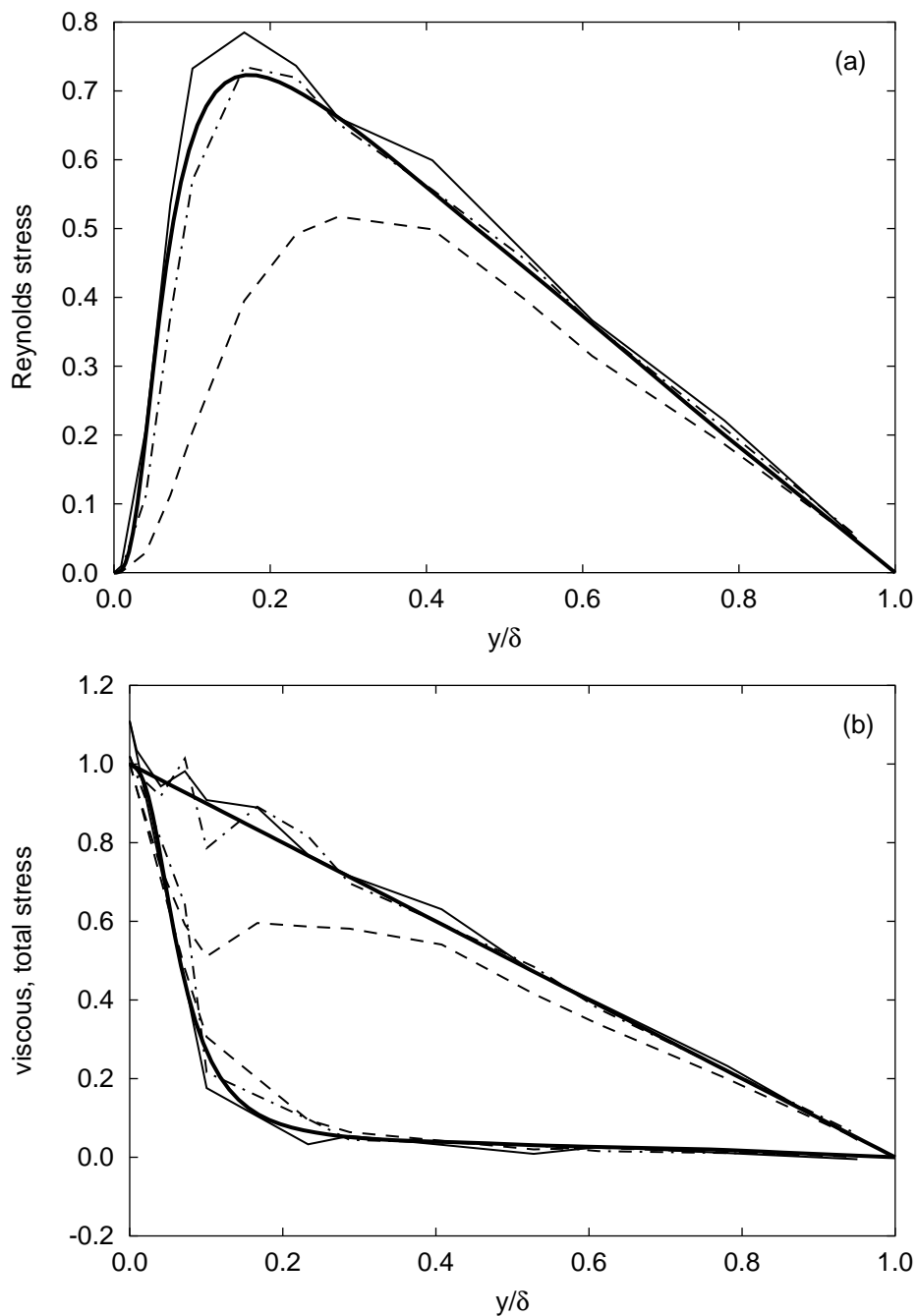


Figure 5.8: Comparison of the Reynolds stress, viscous stress and total stress for $\text{Re}_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

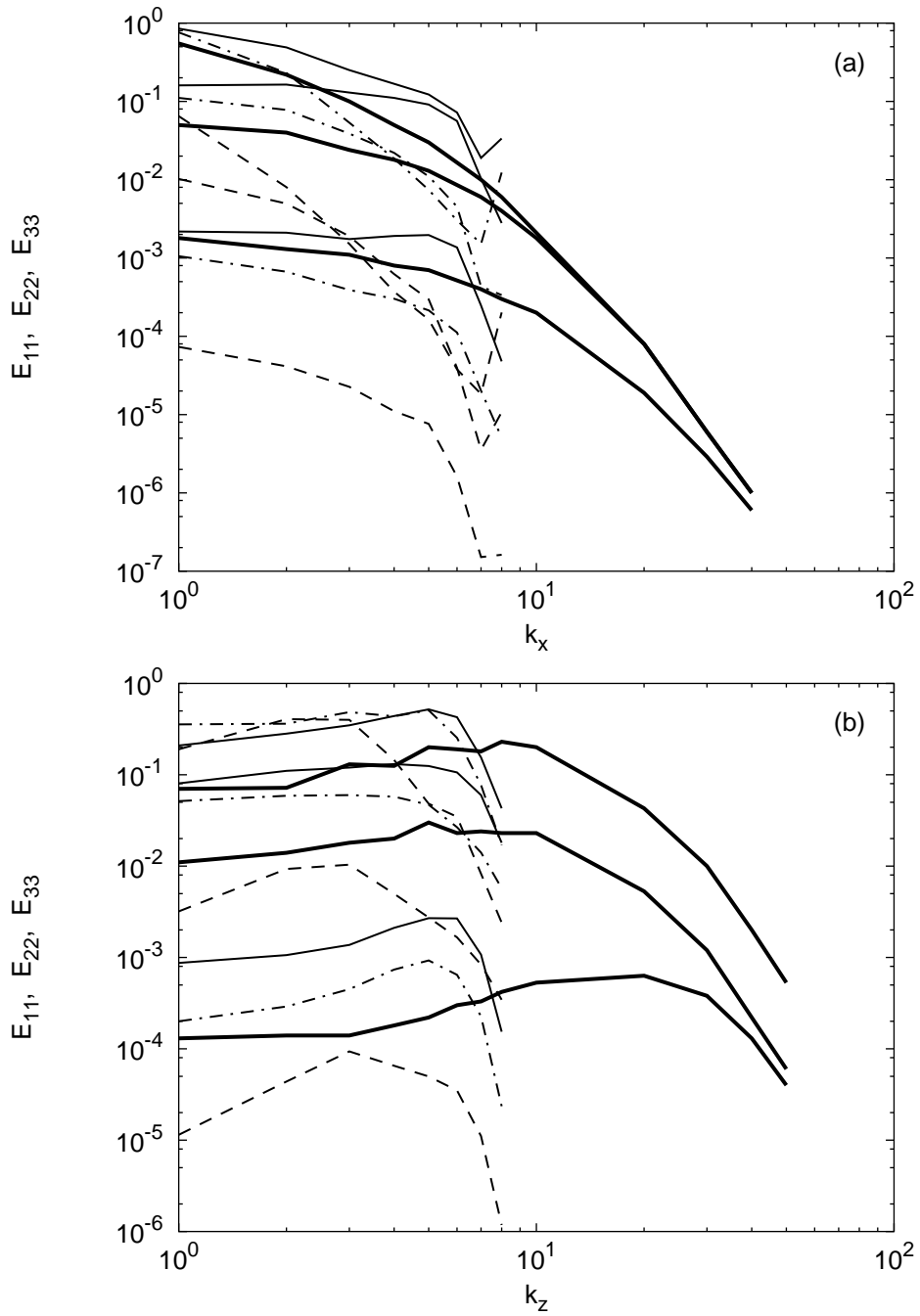


Figure 5.9: Comparison of the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $y^+ = 5.39$ for $\text{Re}_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

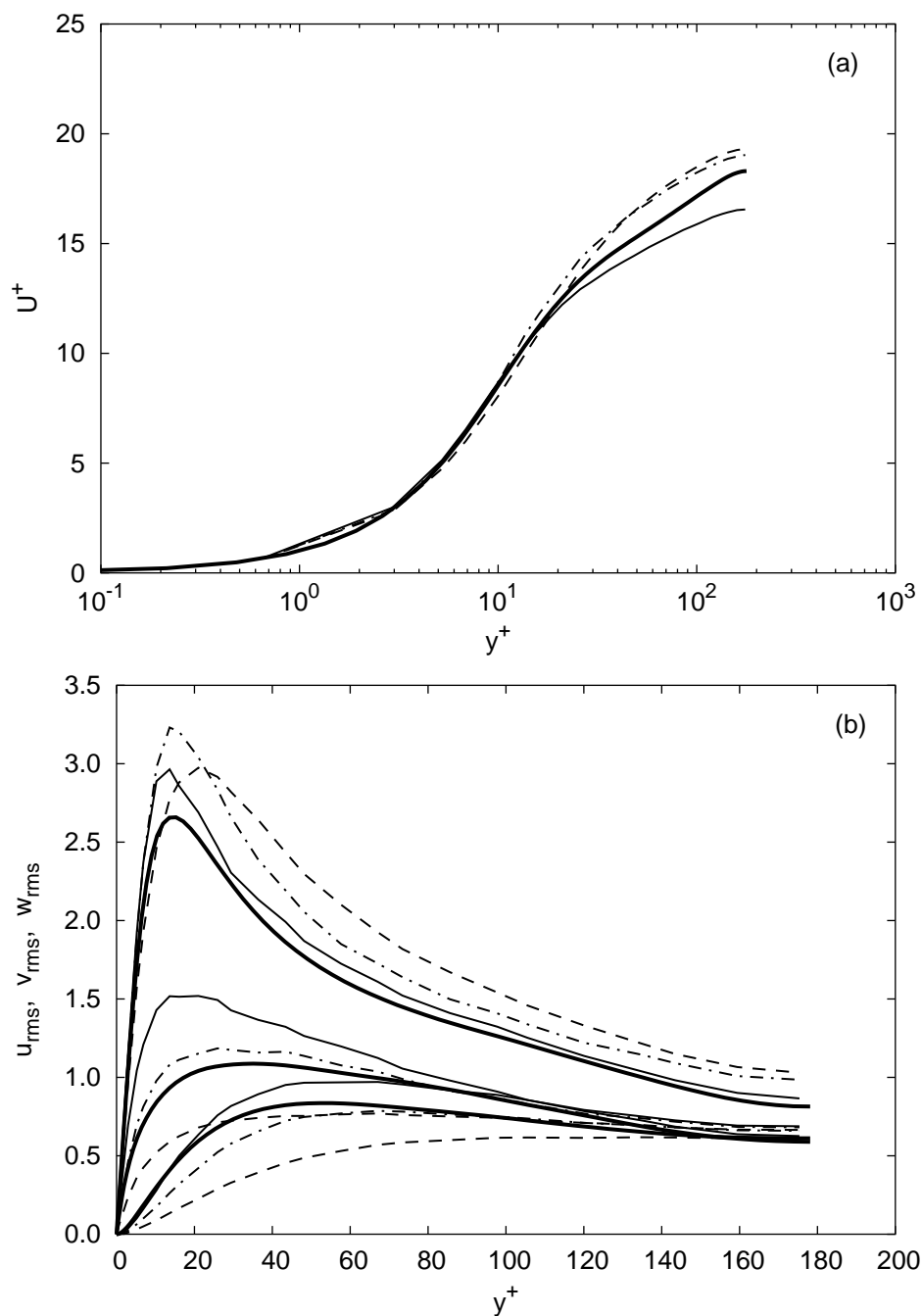


Figure 5.10: Comparison of the mean velocity and rms profiles for $Re_\tau = 180$ obtained using a $(8 \times 16 \times 8)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

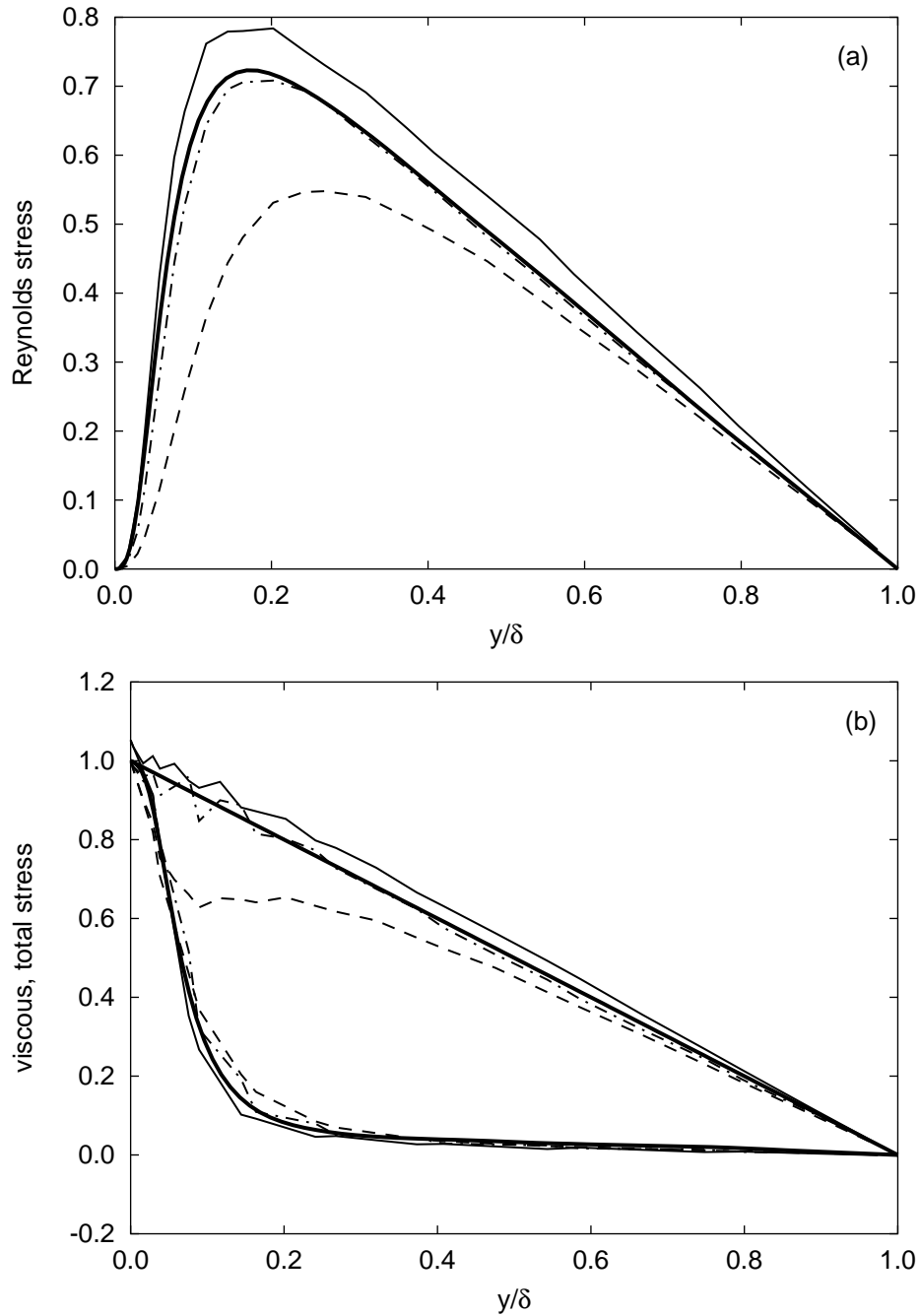


Figure 5.11: Comparison of the Reynolds stress, viscous stress and total stress for $Re_\tau = 180$ obtained using a $(8 \times 16 \times 8)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

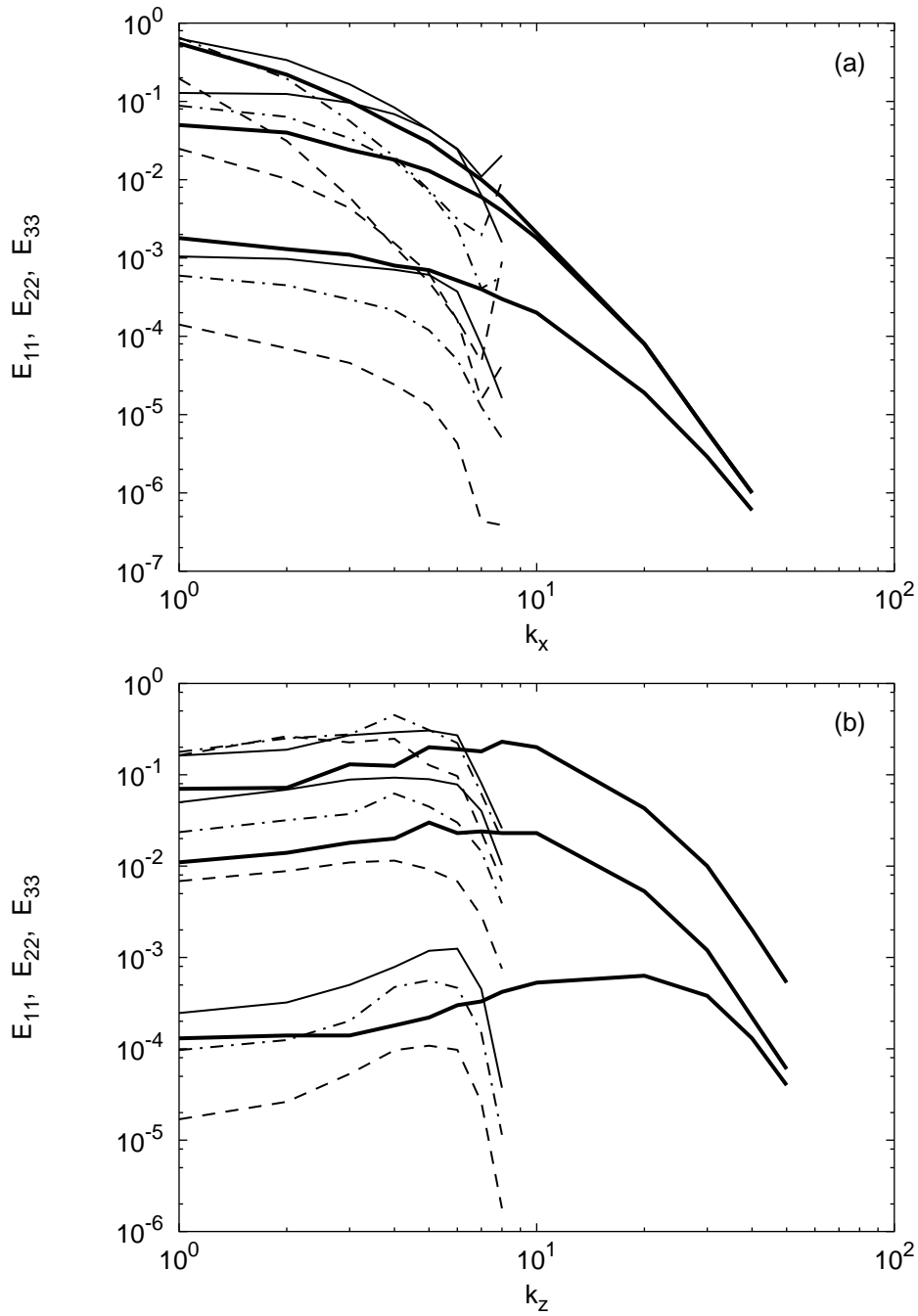


Figure 5.12: Comparison of the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $y^+ = 5.39$ for $\text{Re}_\tau = 180$ obtained using a $(8 \times 16 \times 8)$ mesh with $P = 2$: ——— DNS [39]; ——— no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

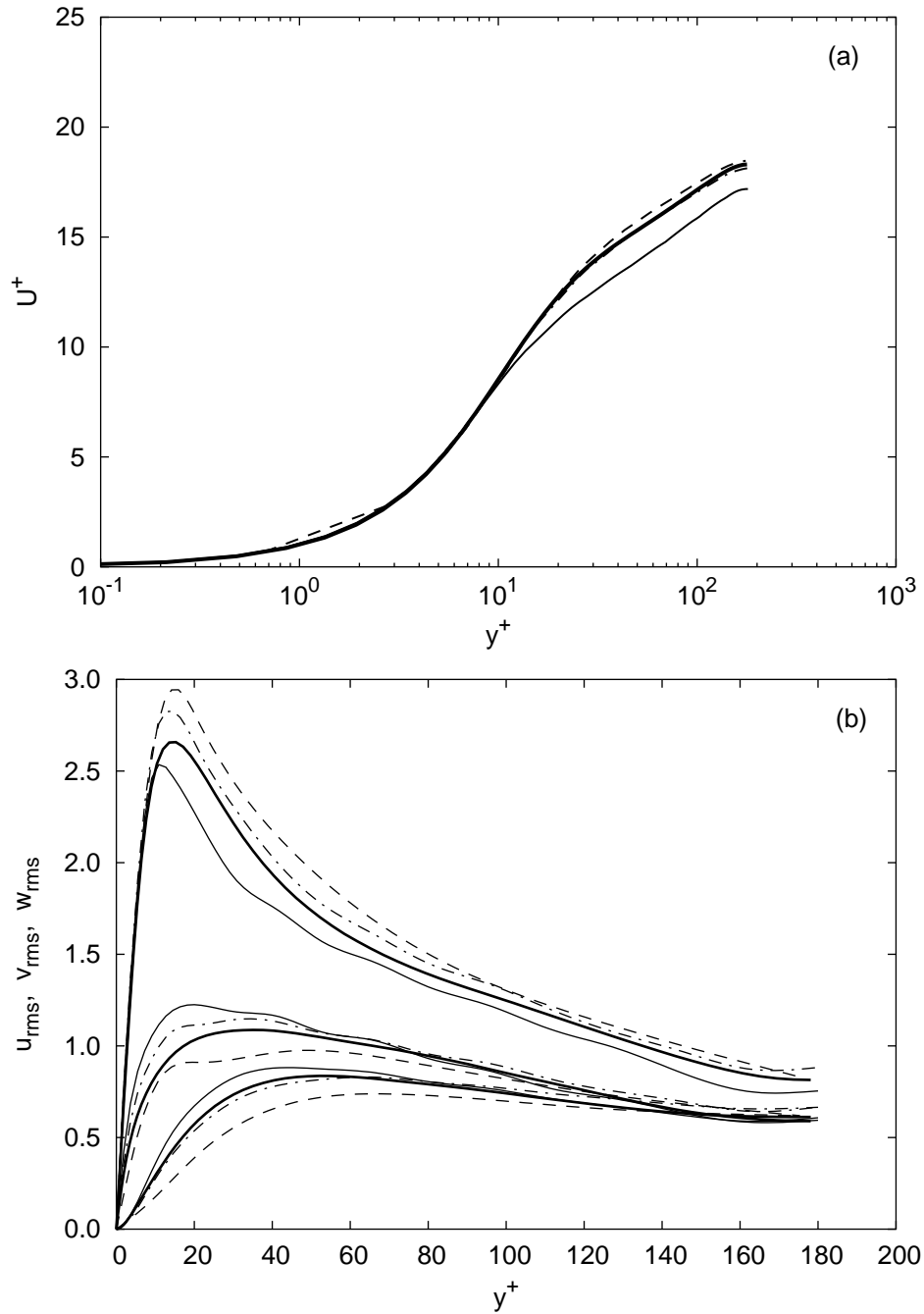


Figure 5.13: Comparison of the mean velocity and rms profiles for $Re_\tau = 180$ obtained using a $(16 \times 16 \times 16)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

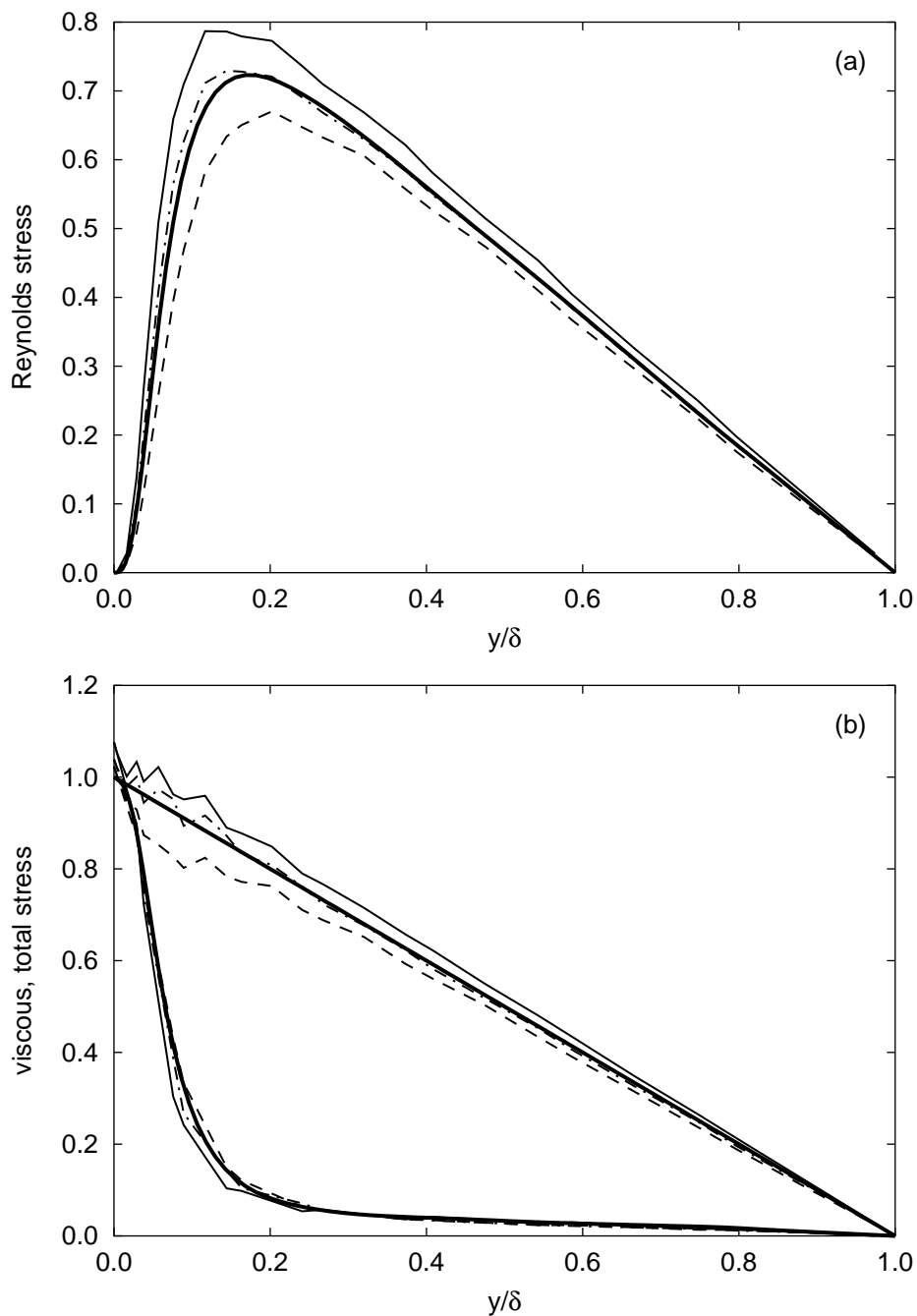


Figure 5.14: Comparison of the Reynolds stress, viscous stress and total stress for $\text{Re}_\tau = 180$ obtained using a $(16 \times 16 \times 16)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

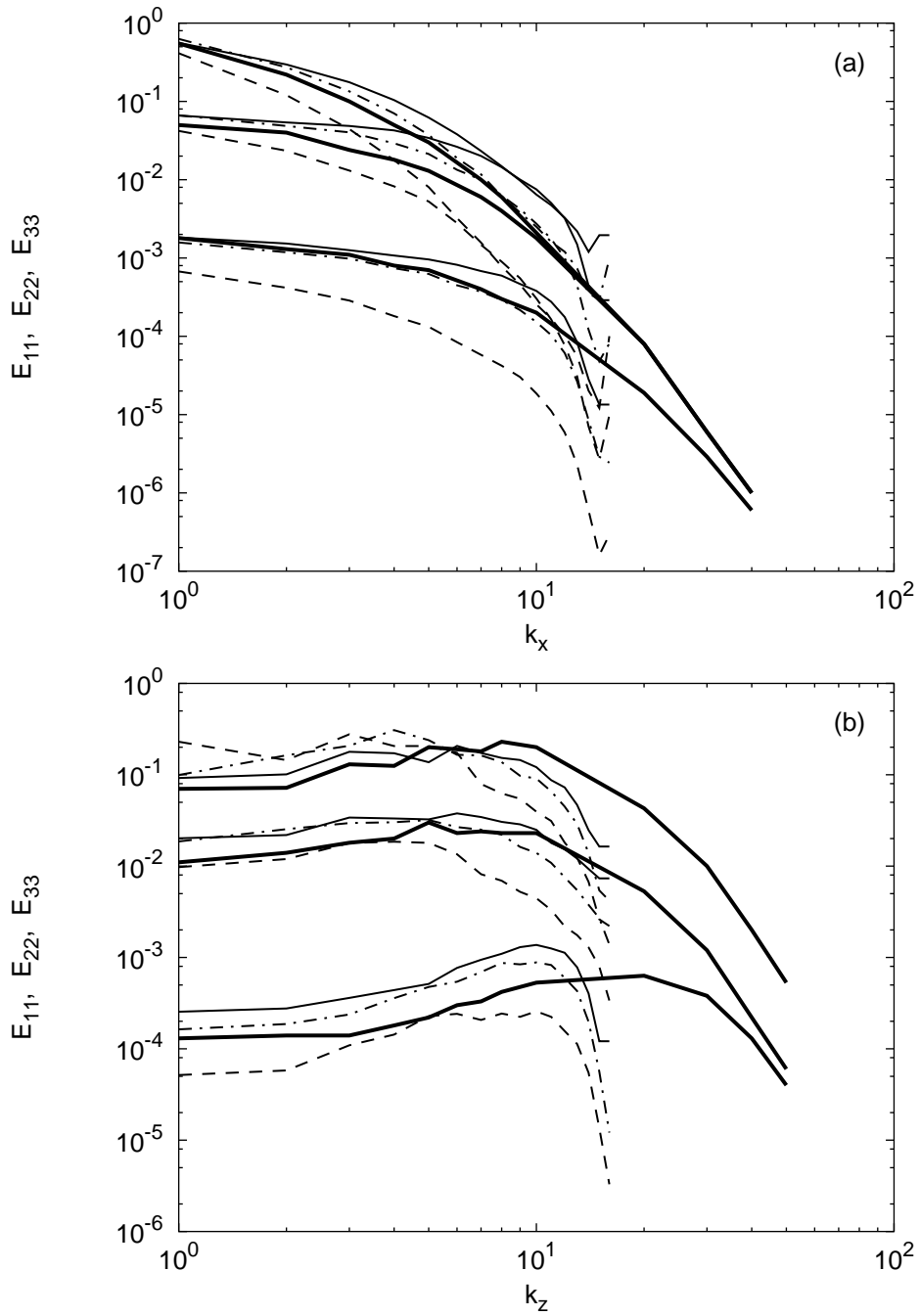


Figure 5.15: Comparison of the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $y^+ = 5.39$ for $\text{Re}_\tau = 180$ obtained using a $(16 \times 16 \times 16)$ mesh with $P = 2$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 2$.

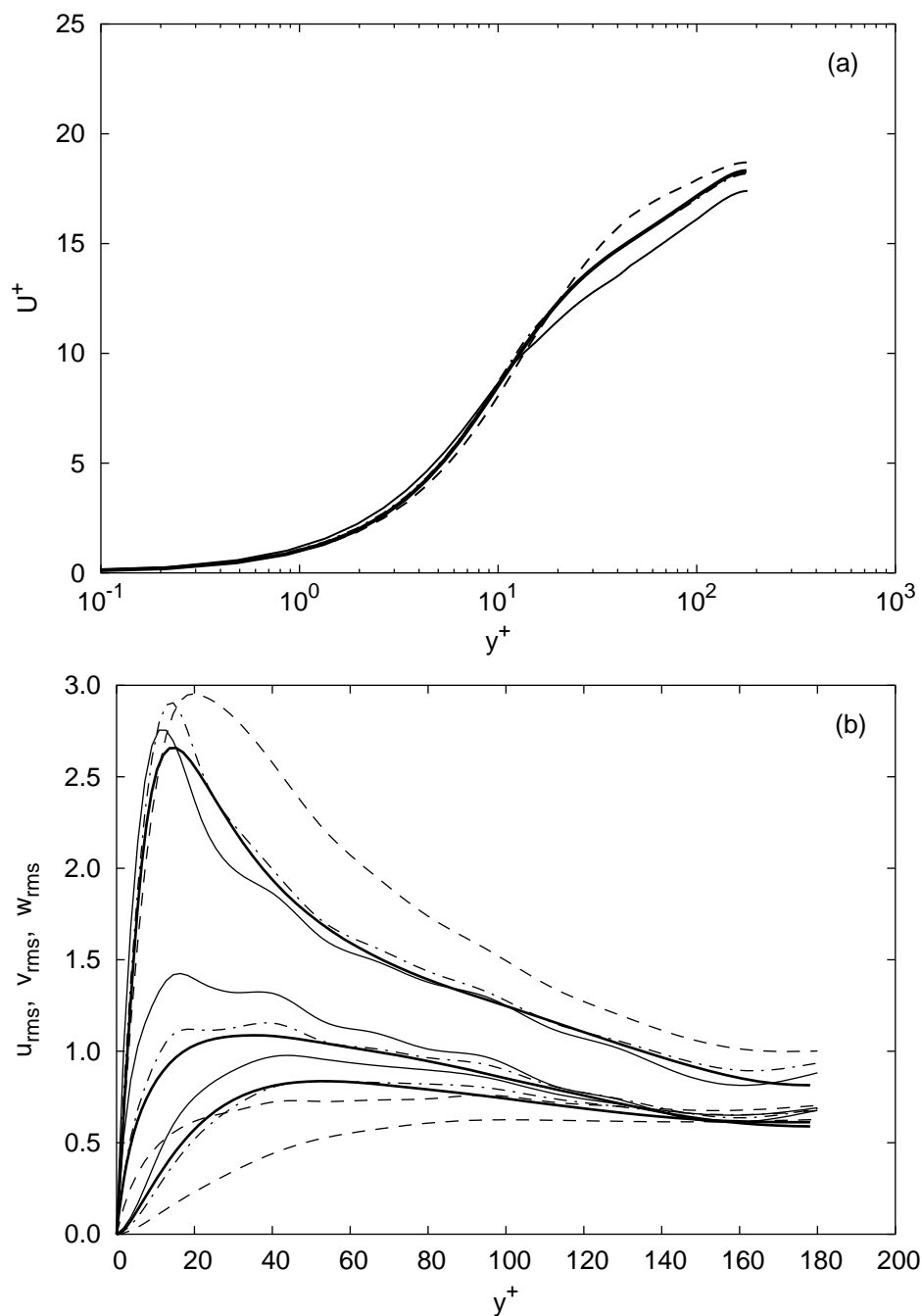


Figure 5.16: Comparison of the mean velocity and rms profiles for $Re_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 3$.

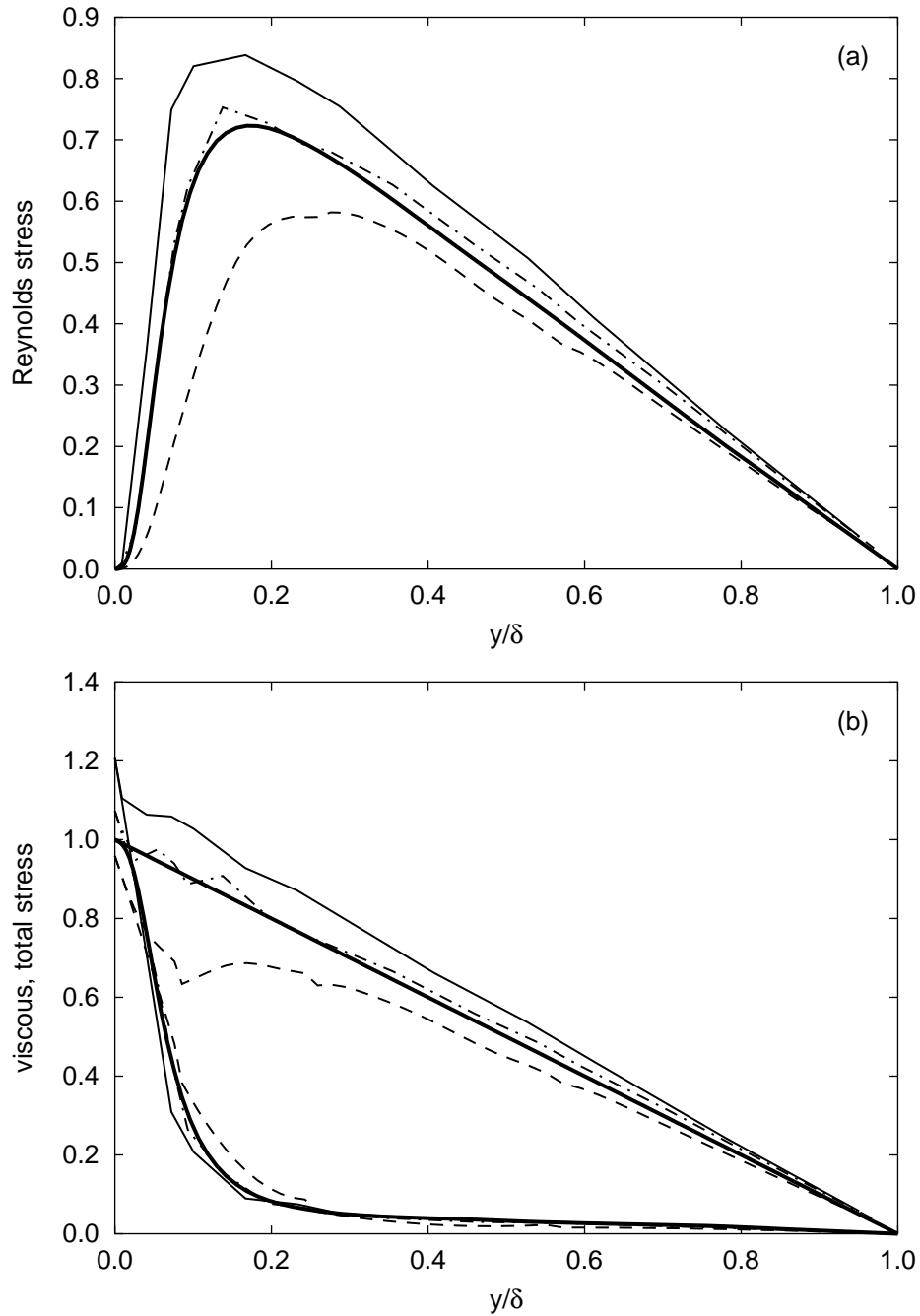


Figure 5.17: Comparison of the Reynolds stress, viscous stress and total stress for $Re_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 3$.

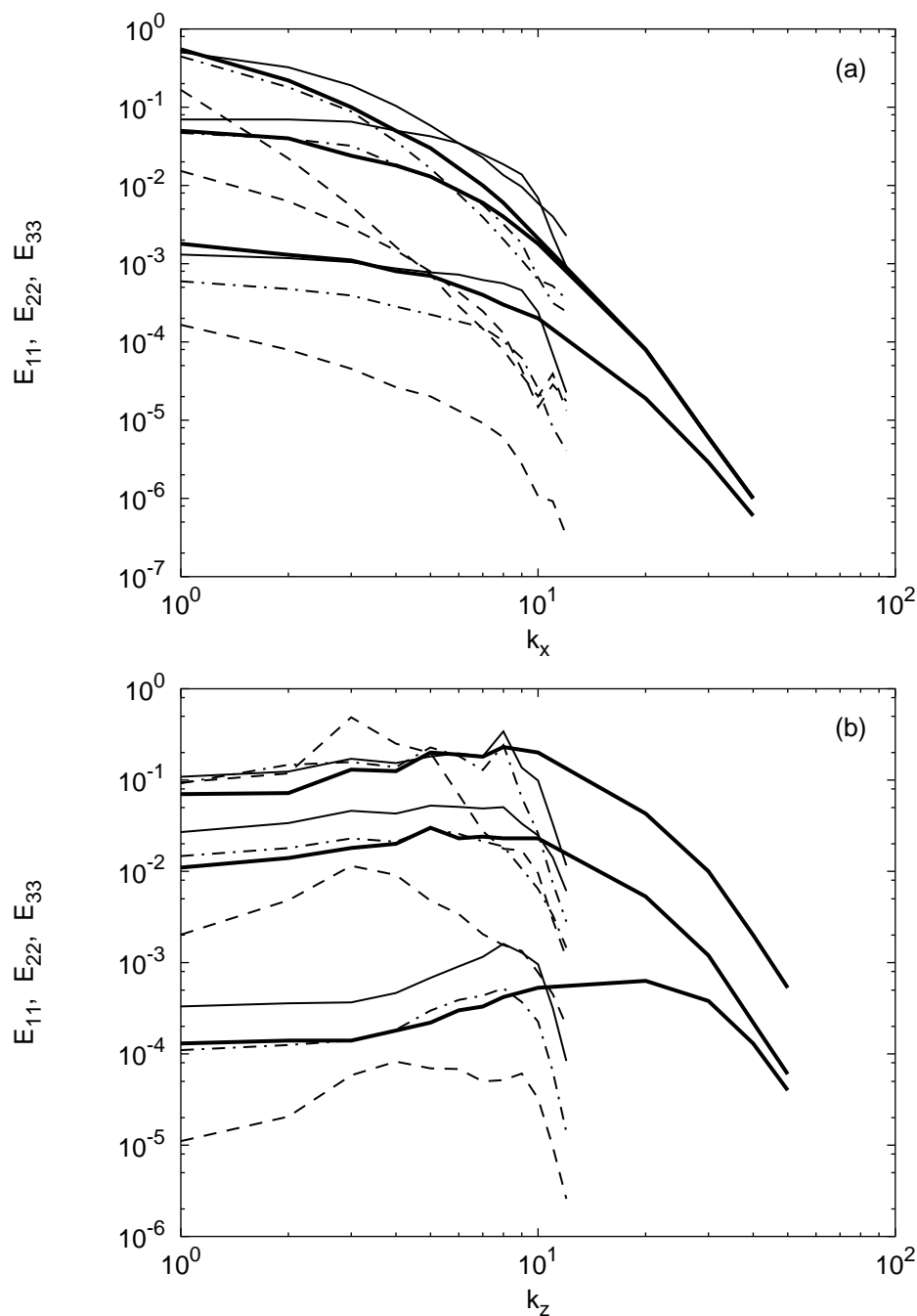


Figure 5.18: Comparison of the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $y^+ = 5.39$ for $\text{Re}_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: — DNS [39]; — no model; - - - LES; - · - VMS with $\tilde{P} = 3$.

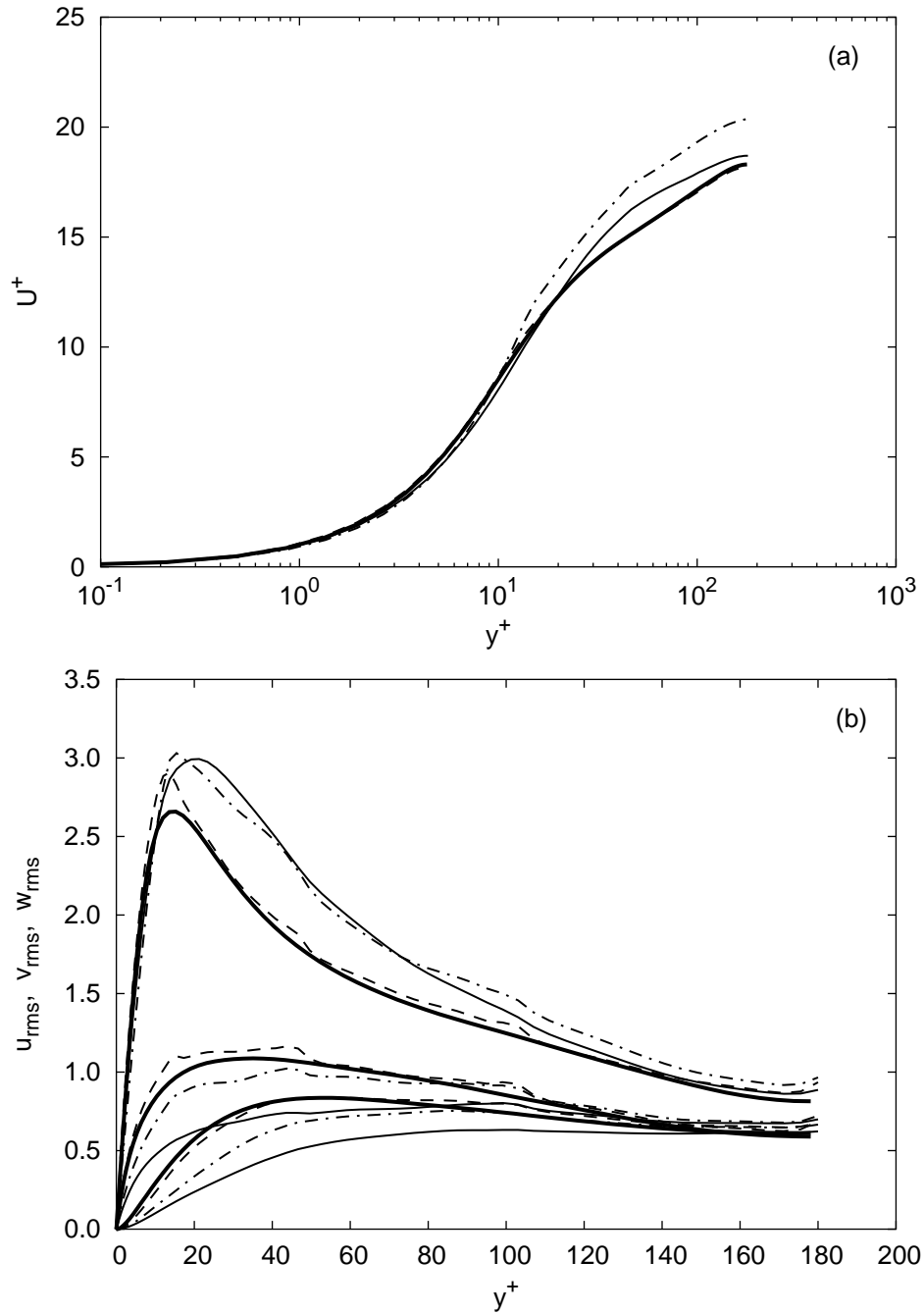


Figure 5.19: Comparison of the mean velocity and rms profiles for $Re_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: — DNS [39]; — LES; - - - VMS with $\tilde{P} = 3$; - · - VMS with $\tilde{P} = 2$.

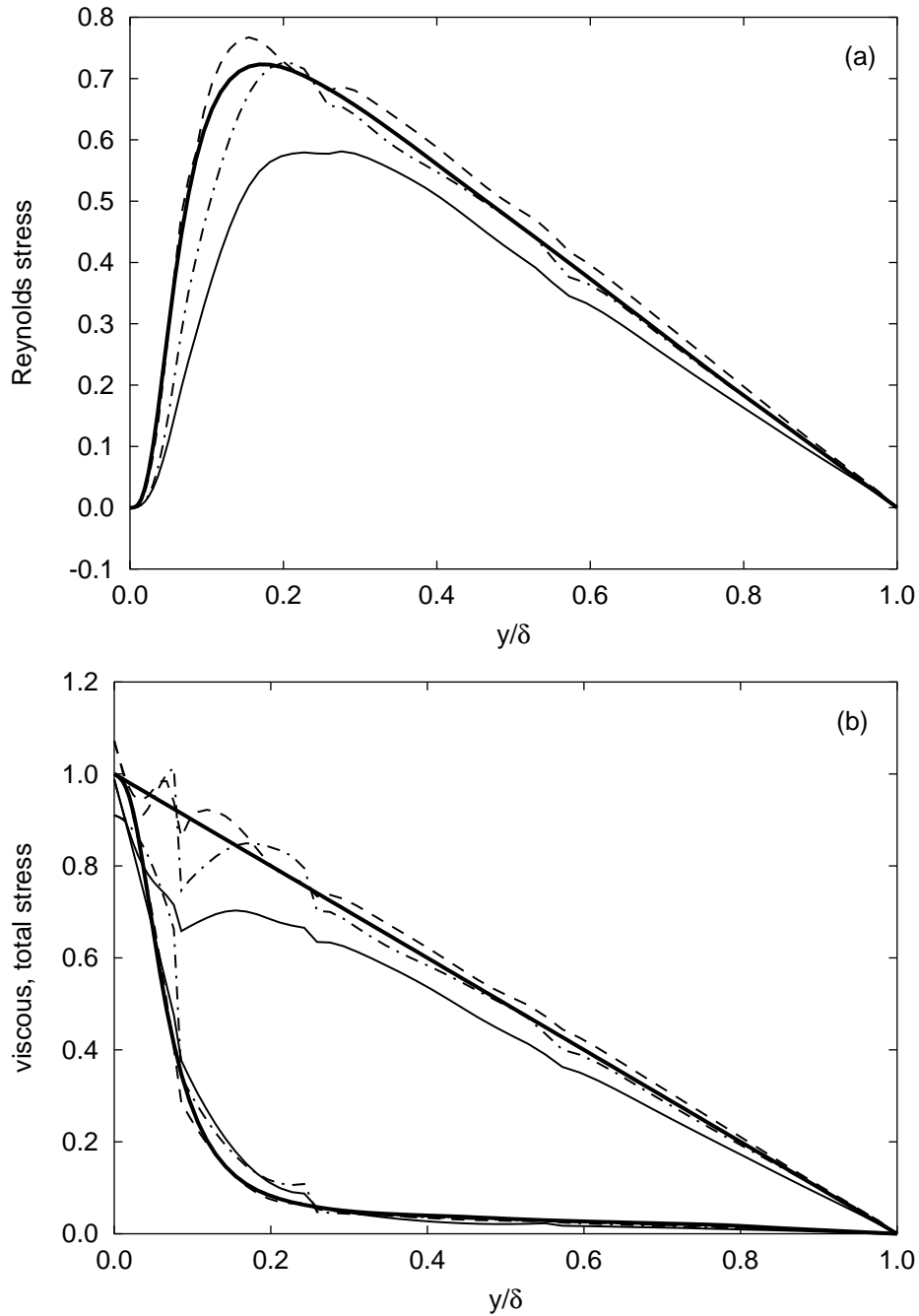


Figure 5.20: Comparison of the Reynolds stress, viscous stress and total stress for $\text{Re}_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: — DNS [39]; — LES; - - - VMS with $\tilde{P} = 3$; - · - VMS with $\tilde{P} = 2$.

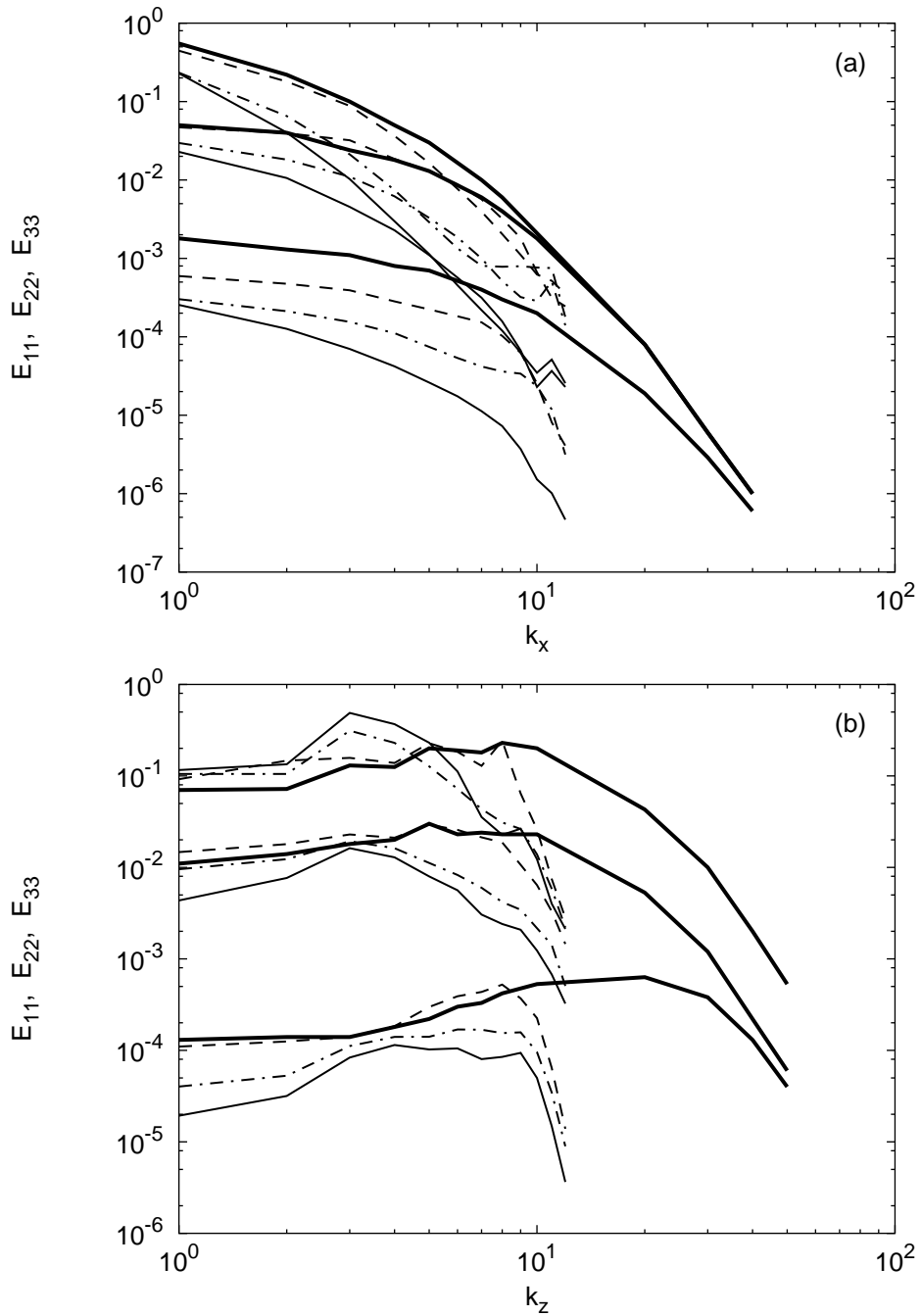


Figure 5.21: Comparison of the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $z^+ = y.39$ for $\text{Re}_\tau = 180$ obtained using a $(8 \times 8 \times 8)$ mesh with $P = 3$: ——— DNS [39]; ——— LES; - - - VMS with $\tilde{P} = 3$; - . - VMS with $\tilde{P} = 2$.

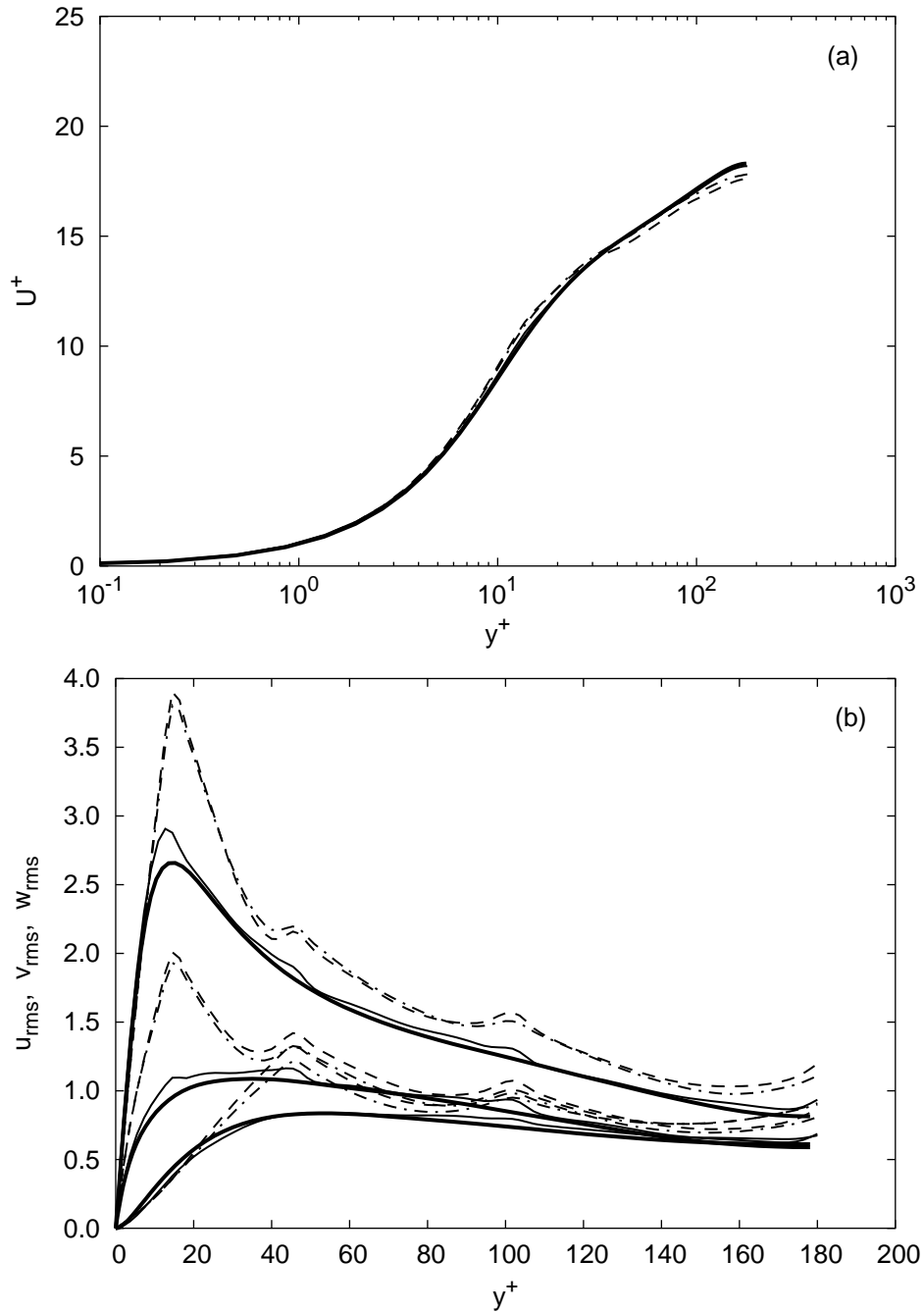


Figure 5.22: Effect of the serendipity expansion on the mean velocity and rms profiles for $Re_\tau = 180$ for an $(8 \times 8 \times 8)$ mesh using the VMS method with $\tilde{P} = 3$:
 — DNS [39]; — full basis for $P = 3$; - - - serendipity basis for $P = 3$;
 - · - serendipity basis for $P = 4$.

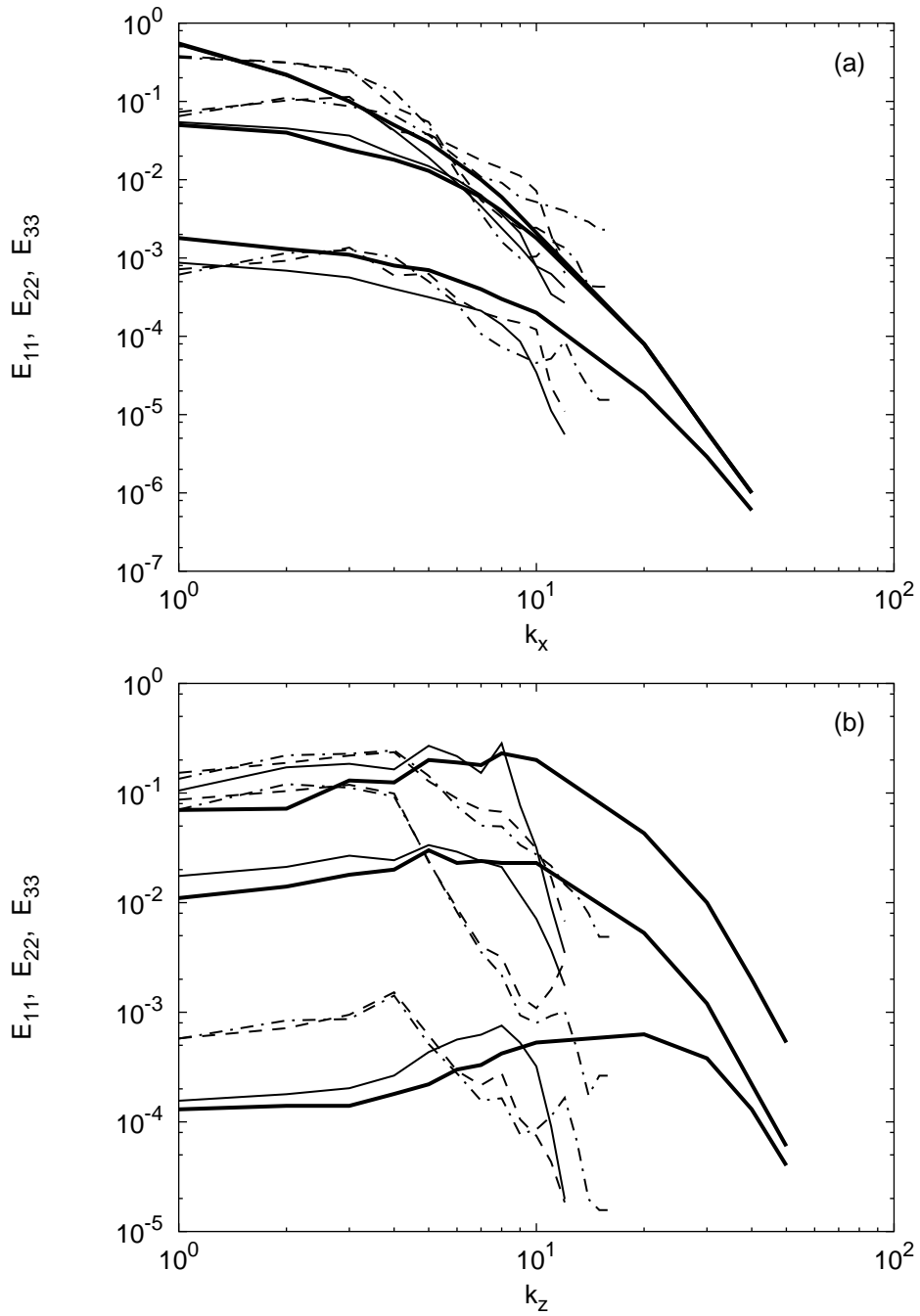


Figure 5.23: Effect of serendipity expansion on the one-dimensional streamwise and spanwise turbulent kinetic energy spectra at $y^+ = 5.39$ for $\text{Re}_\tau = 180$ for an $(8 \times 8 \times 8)$ mesh using the VMS method with $\tilde{P} = 2$: — DNS [39]; — full basis for $P = 3$; - - - serendipity basis for $P = 3$; - · - serendipity basis for $P = 4$.

Chapter 6

Performance analysis

In this chapter we investigate the factors influencing the performance of the proposed discretizations for large eddy simulation. For this purpose, we consider again the turbulent channel flow at $\text{Re}_\tau = 180$ and $\text{M} = 0.3$. In Section 6.1, we consider the effect of the time-step size on the accuracy and computational cost of the time discretizations. Next, in Section 6.2, we compare the accuracy and cost of different spatial resolutions using an equivalent number of degrees of freedom. Additionally, we examine the efficiency of parallel computing. In Section 6.3 we consider the influence of quadrature order, and in Section 6.4 we focus on the effect of convergence criteria on the solution quality.

6.1 Temporal discretization

Here, we consider the effect of the time-step size on the accuracy and computational work involved in the different time discretizations (Section 3.2). The investigation is performed using the VMS method at a fixed spatial resolution of $(16 \times 16 \times 16)$ elements with $P = \tilde{P} = 2$. Combined with the TDG method using $\Delta t = 0.002$, this resolution proved to be a representative candidate for turbulent channel flow (Section 5.3.3), where it is desired to minimize the number of degrees of freedom for a given

level of accuracy.

In the following, we consider three different time-step sizes, viz. $\Delta t = 0.001$, $\Delta t = 0.002$ and $\Delta t = 0.004$. The TCG method proved unable to converge at these time-step sizes, and is therefore considered to be a relatively uncompetitive method for this type of problem. As a result, the TCG method is not considered hereafter. For convenience, we label the TCG method using $\Delta t = 0.001$, $\Delta t = 0.002$ and $\Delta t = 0.004$ as TCG1, TCG2 and TCG3 respectively. Similarly, for the TDG method these cases are referred to as TDG1, TDG2 and TDG3. For all cases, the linear and non-linear convergence criteria (Chapter 4) are $\varepsilon_{gmres} = \varepsilon_{update} = \varepsilon_{res} = 0.001$.

6.1.1 Accuracy

The effects of the time-step sizes on the mean velocity and turbulence intensities obtained using the TDG and TCG method are shown in Figures 6.1 and 6.2. For both methods, the range of time step sizes do not have a visible influence on the mean velocity, while the effect on the rms profiles is very small.

6.1.2 Computational cost

Next, we investigate the effect of the time-step size on the computational cost. For this purpose we consider the parameters involved in the predictor multi-corrector algorithm and their corresponding CPU times. Although the comparison of CPU times between codes can be misleading, it is justified here. In this case, the implementation of the two methods has occurred in the same code, and is virtually identical except for a few details.

The main components of the computational work within a time-step are given by

- Jacobian matrix evaluation,
- residual vector evaluation,

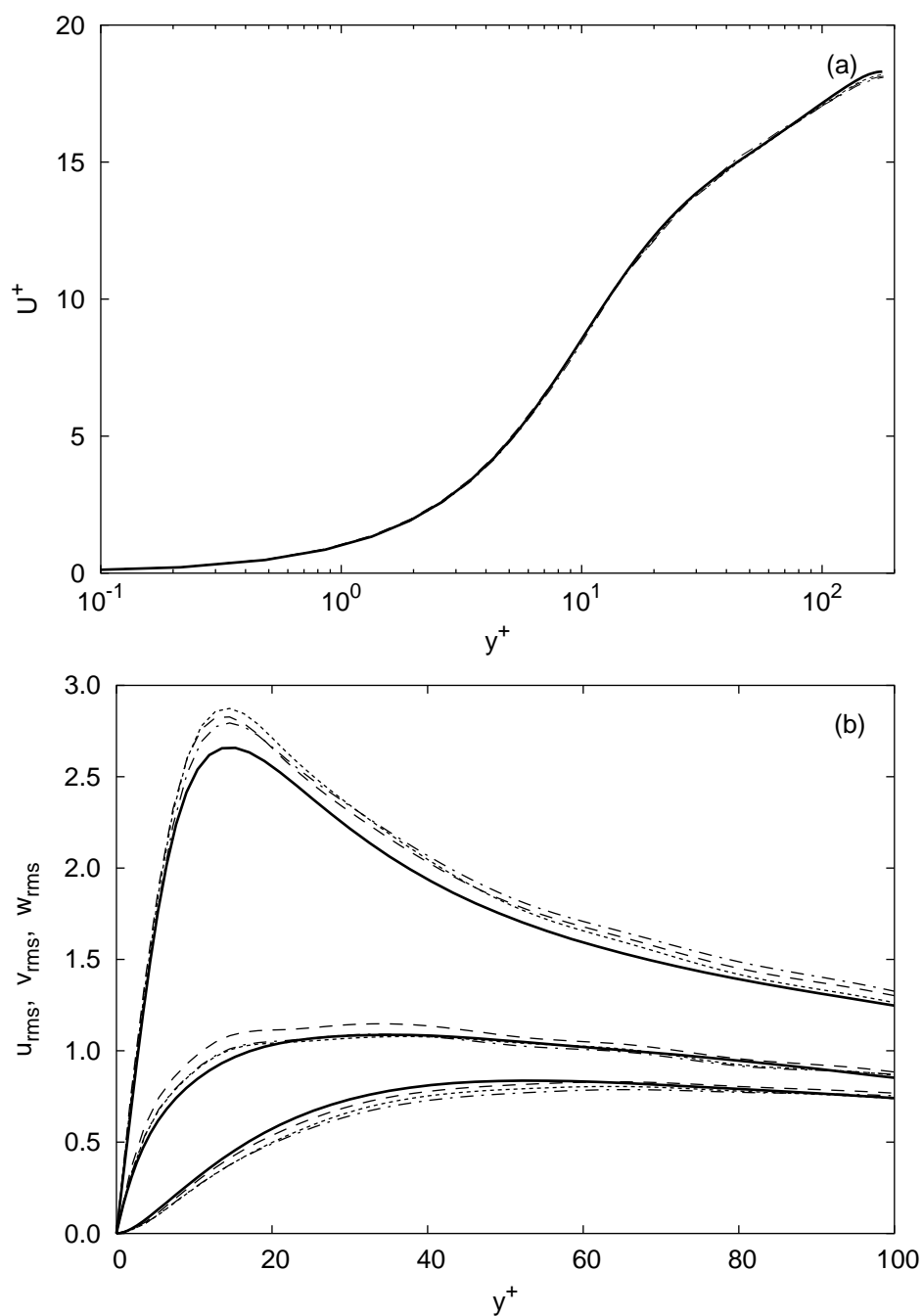


Figure 6.1: Influence of the time-step size on the mean velocity and rms profiles for the TDG method with $(16 \times 16 \times 16)$ elements and $P = \tilde{P} = 2$ for turbulent channel flow at $Re_\tau = 180$: — DNS [39]; - - - $\Delta t = 0.001$; - · - $\Delta t = 0.002$; ···· $\Delta t = 0.004$.

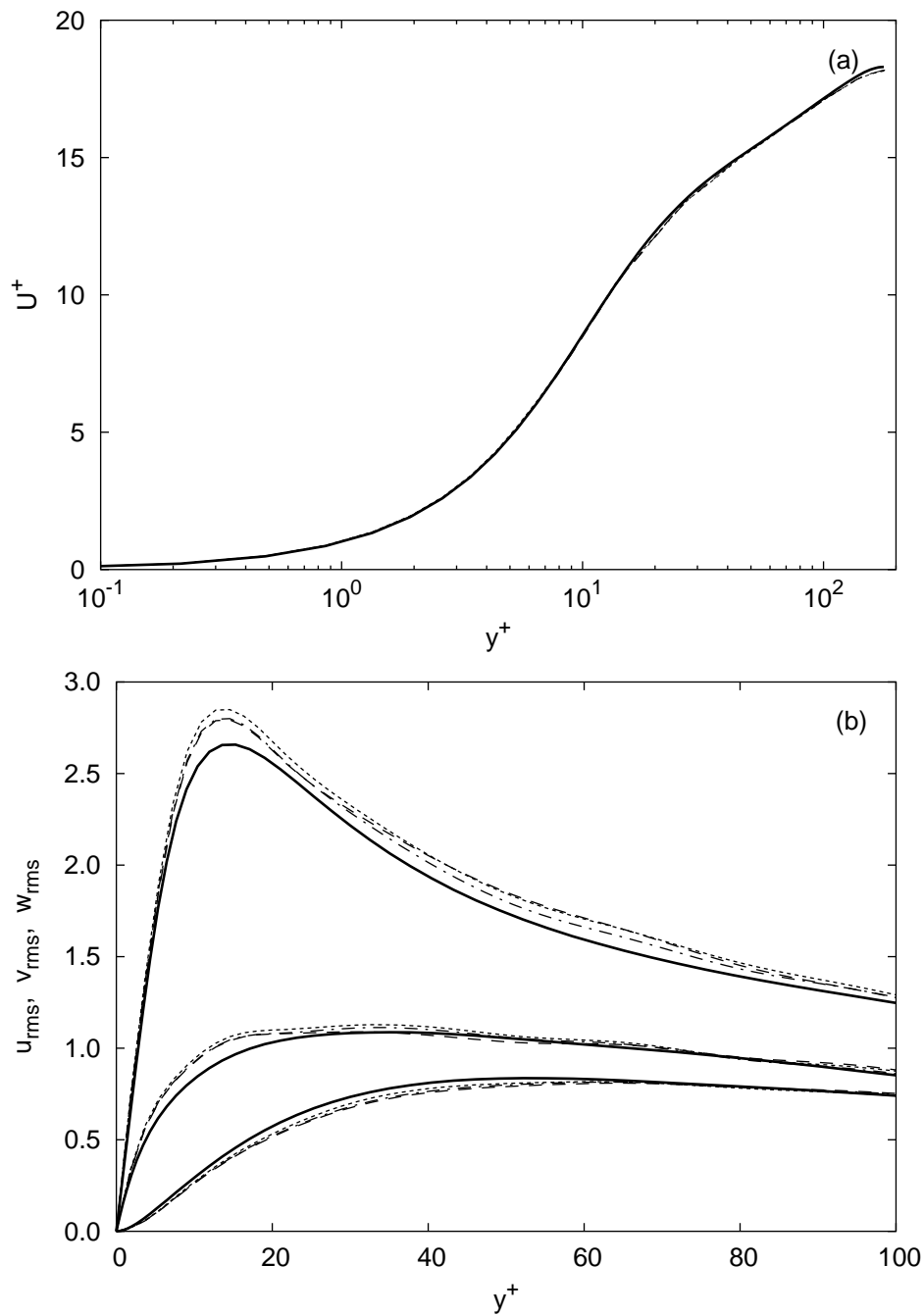


Figure 6.2: Influence of the time-step size on the mean velocity and rms profiles for the TCPG method with $(16 \times 16 \times 16)$ elements and $P = \tilde{P} = 2$ for turbulent channel flow at $Re_\tau = 180$: — DNS [39]; - - - $\Delta t = 0.001$; - · - $\Delta t = 0.002$; ···· $\Delta t = 0.004$.

- linear system solution,
- number of corrector passes,

and the total CPU time per time step can be approximated by

$$\text{cpu}_{\Delta t} \approx n_{\text{corr}}(\text{cpu}_{\text{Jac}} + \text{cpu}_{\text{res}} + n_{\text{gmres}}\text{cpu}_{\text{gmres}}) + \text{cpu}_{\text{res}}, \quad (6.1)$$

where n_{corr} is the number of corrector passes required for non-linear convergence, cpu_{Jac} is the CPU time required for a single Jacobian matrix evaluation, cpu_{res} is the CPU time required for a single residual vector evaluation, n_{gmres} is the number of GMRES iterations required for convergence of the linear system solution, and $\text{cpu}_{\text{gmres}}$ is the CPU time required for a single GMRES iteration. The values for each of the components in (6.1) obtained using the TDG and TCPG methods are shown in Table 6.1. All simulations are performed on an SGI Altix 350 cluster using four 1.4GHz IA-64 Intel Itanium 2 processors in parallel. All simulations are performed on an SGI Altix 350 cluster using four 1.4GHz IA-64 Intel Itanium 2 processors in parallel. As the components in (6.1) vary slightly from one time step or corrector pass to the next, their values are averaged over 1000 time steps. In the following sections we discuss each of these components.

Jacobian matrix evaluation

The assembly of the Jacobian matrix is by far the most time-consuming operation. Fortunately, the Jacobian does not have to be updated at every corrector step for this type of problem. In the present simulations the Jacobian is only updated whenever the drop in any component of the update is less than 20% of that of the previous update. Remarkably, this condition never occurred, meaning the Jacobian remained fixed at its state computed at the first time step. Moreover, the number of corrector passes was found to be approximately constant during the entire simulation, even through the transient region. This advantageous behavior is likely due to the initial

case	cpu_{Jac}	n_{corr}	cpu_{res}	n_{gmres}	cpu_{gmres}	$\text{cpu}_{\Delta t}$
TDG1	-	3	1.07	8	1.14	31.6
TDG2	-	3	1.07	8	1.14	31.6
TDG3	-	4	1.07	14	1.14	69.2
TCPG1	-	4	1.22	7	0.29	14.2
TCPG2	-	4	1.22	8	0.29	15.4
TCPG3	-	5	1.22	11	0.29	23.3

Table 6.1: Contributions per time step for the TDG and TCPG method (CPU times in sec.) for turbulent channel flow using the VMS method on a $(16 \times 16 \times 16)$ mesh with $P = 2$ and $\tilde{P} = 2$; $\varepsilon_{update} = \varepsilon_{res} = 0.001$, $\varepsilon_{gmres} = 0.001$, 4 processors.

condition, which provides a reasonable approximation to the final state. Additionally, the time-dependent behavior of the fully-developed channel flow is apparently not strong enough to draw the solution out of the domain of convergence of the non-linear algorithm. As a consequence of the large number of time steps required for the simulation, the contribution of evaluating the Jacobian matrix to the total execution time is negligible. Therefore, we do not take this contribution into account.

Residual vector evaluation

The residual vector evaluation for both TCPG and TDG methods are very similar. For the results tabulated here, both methods used an equivalent quadrature order. However, since the TCPG method employs a constant weighting function in time, further savings can be realized with a lower quadrature order. This will be discussed in Section 6.3.

Linear system solution

The number of GMRES iterations (n_{gmres}) for both methods is very similar, however, the CPU time required per GMRES iteration for the TDG method is about a factor four larger. The latter is due to the fact that the TDG method has twice as many degrees of freedom compared to the TCPG method. Consequently, the Jacobian matrix has twice the number of rows and columns and therefore four times as many entries. Since the major contribution to a single GMRES iteration is the product of the Jacobian matrix and a (Krylov) vector (Section 4.6.1), the total work per GMRES iteration is approximately a factor four larger for the TDG method.

Note that the number of GMRES iterations increases with the time-step size. This is likely due to the zero initial guess for the GMRES algorithm, which in early Newton iterations becomes increasingly less appropriate as the time-step size increases. Currently, the initial guess for the solution (predictor) is set equal to the final solution of the previous slab. A better initial guess for the solution, e.g. obtained from extrapolation of previous solutions, could be used to reduce the update in the first Newton iteration. This way, the zero initial guess for the GMRES algorithm becomes a more suitable approximation.

Non-linear system solution

The TDG method requires one corrector pass (n_{corr}) less for non-linear convergence compared to the TCPG method. Both methods require an additional corrector step for the largest time step $\Delta t = 0.004$ compares to the lower time steps $\Delta t = 0.001$ and $\Delta t = 0.002$. This is due to the fact that for larger time-step sizes the predictor is further away from the final solution than for a smaller time-step size. Therefore, an improved initial guess for the solution could also reduce the number of corrector passes required. Note that an improved predictor could therefore simultaneously improve both linear and non-linear convergence.

Relative contributions

The total CPU times per time step are shown in Figure 6.3, together with the total contributions of the residual evaluation and linear system solution. The CPU time per time step of the TCPG method is about three times smaller than that of the TDG method. TCPG1 and TCPG2 require a comparable CPU time per time step, while that for TCPG3 is considerably larger. A similar trend is observed for the TDG method, although the CPU times per time step of the TDG methods are significantly larger.

The contributions of the residual evaluation and linear system solution are approximately equal for TCPG1 and TCPG2, while for TCPG3 the linear system solution becomes the dominant factor. The cost of the TDG method, in contrast, is always dominated by linear system solution.

6.1.3 Discussion

Taking into account the fact that fewer time steps are required when using larger time-step sizes, both methods using time-step size $\Delta t = 0.002$ and $\Delta t = 0.004$ require approximately half the total CPU time required by $\Delta t = 0.001$.

The small effect of the time-step sizes considered on the accuracy of the solution implies that potentially even larger time-step sizes can be used without seriously degrading the solution quality. The computational work, however, was approximately equal for $\Delta t = 0.002$ and $\Delta t = 0.004$, due to the decreased convergence rates of both the linear and non-linear system solution associated with increased time-step size. Therefore, there seems to be little motivation for using time-step sizes larger than $\Delta t = 0.002$. At $\Delta t = 0.002$, the TCPG method is significantly less expensive than the TDG method. This is mainly due to the GMRES iterations which require less work since the TCPG method requires only half the number of degrees of freedom compared to TDG. Obviously, since both methods produce very similar results, use

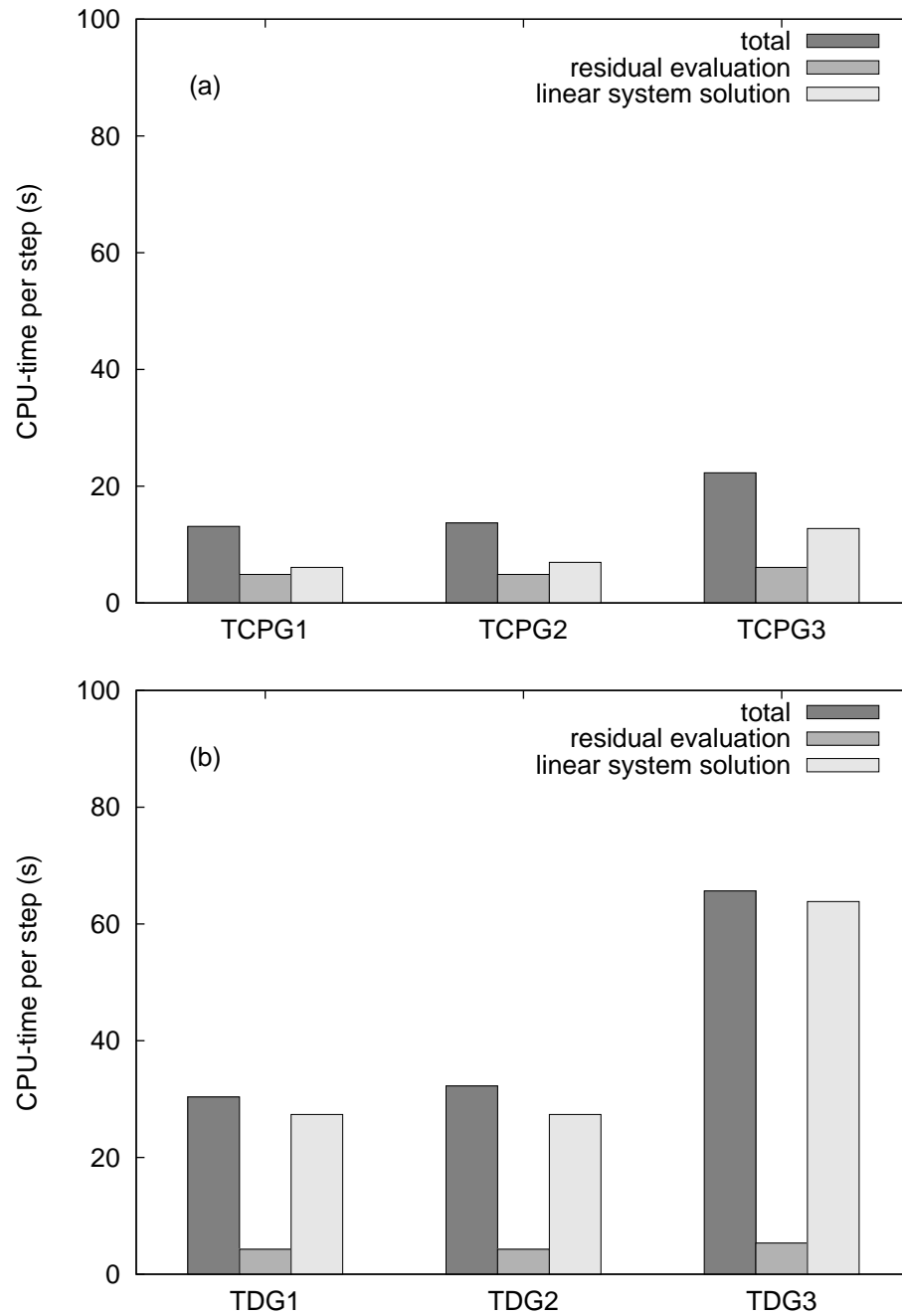


Figure 6.3: CPU times per time step for the TCGP and TDG methods.

of the TCPG method is preferred, except where a weak initial condition is required for reasons of adaptation. This conclusion might be overturned if a procedure for improving iterative convergence rates at larger time-step sizes can be found, such as extrapolation techniques, allowing the higher rate of convergence of the TDG method (i.e. third order as opposed to second order of TCPG) to be exploited, as observed in Section 3.4.4.

6.2 Spatial discretization

In this section we analyze the computational efficiency of different spatial hp -resolutions employing a similar number of degrees of freedom. In particular, we consider the following cases:

- h32p1: $(32 \times 32 \times 32)$ mesh with $P = 1$
- h16p2: $(16 \times 16 \times 16)$ mesh with $P = \tilde{P} = 2$
- h8p4: $(8 \times 8 \times 8)$ mesh with $P = \tilde{P} = 4$

All methods employ the Smagorinsky model with $C_s = 0.1$, and no attempts were made to tune this constant to achieve the highest accuracy. Since h32p1 employs linear basis functions only, it does not lend itself to scale decomposition such as used in the present VMS method. Consequently, the SGS model acts on all scales directly, such as in a traditional LES method. Cases h16p2 and h8p4, on the other hand, employ the VMS method with the SGS model acting on the smallest resolved scales only. All cases employ the TCPG method with $\Delta t = 0.002$, as it was shown to be the most efficient time discretization considered. As before, the convergence criteria are $\varepsilon_{gmres} = \varepsilon_{update} = \varepsilon_{res} = 0.001$.

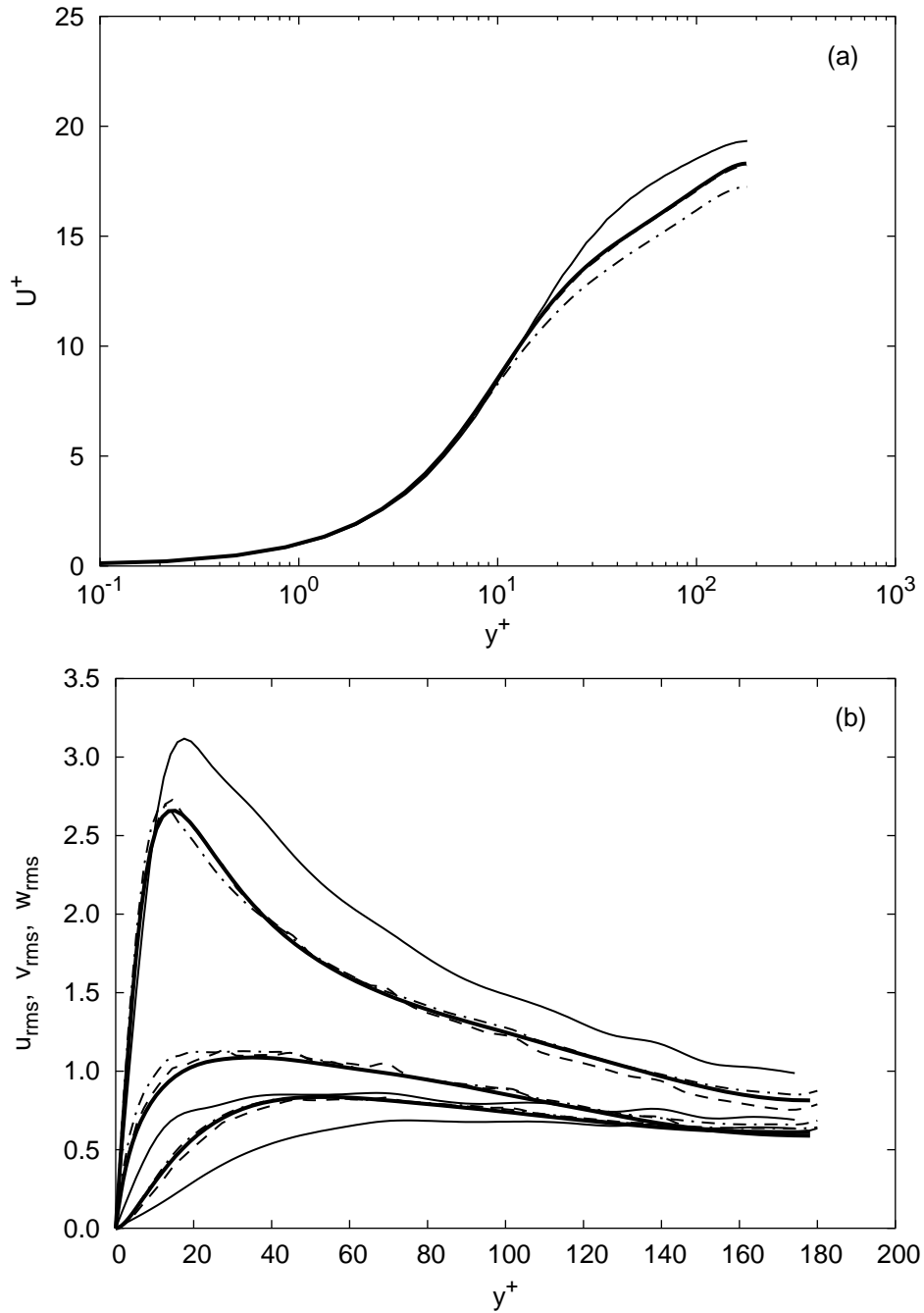


Figure 6.4: Mean velocity and rms profiles for turbulent channel flow at $Re_\tau = 180$ using different hp -resolutions for a similar number of degrees of freedom: — ($32 \times 32 \times 32$) mesh with $P = 1$; — DNS [39]; - - - ($16 \times 16 \times 16$) mesh with $P = 2$ and $\tilde{P} = 2$; - · - ($8 \times 8 \times 8$) mesh with $P = 4$ and $\tilde{P} = 4$.

6.2.1 Accuracy

The mean velocity profiles are shown in Figure 6.4(a). Case h16p2 is clearly the most accurate, h8p4 overpredicts the wall-shear stress while h32p1 underpredicts the wall-shear stress. Moreover, h32p1 does not exhibit the correct logarithmic behavior. The rms profiles, shown in 6.4(b), obtained for h16p2 and h8p4 are of similar quality and in reasonable agreement with the DNS [39], while those of h32p1 are of poor quality.

6.2.2 Computational cost

Next, we examine the computational cost of the different spatial *hp*-resolutions. All simulations are performed on an SGI Altix 350 cluster consisting of 12 1.4GHz IA-64 Intel Itanium 2 processors having 3GB of memory each. Because of its shared-memory NUMA (Non-Uniform Memory Acces) architecture, a single processor can access up to 36GB of memory. In general, the simulations are performed using multiple processors in parallel. The performance obtained on a single processor, however, serves to illustrate the basic differences in the contributions to the computational work for the different spatial *hp*-resolutions. Additionally, it serves as a starting point from which trends can be observed when using multiple processors in parallel.

The contributions constituting the computational work within a time step, as described by (6.1), obtained using one, two, four and eight processors in parallel are shown in Table 6.2. First, we investigate the performance obtained on a single processor. Next, we consider the effect of parallel computing.

Single processor

On a single processor, the lowest total CPU time per time step is obtained for h32p1, while h16p2 requires approximately 25 percent more CPU time. Case h8p4, on the other hand, is significantly more expensive, requiring approximately five times the

case	# proc	n_{corr}	cpu_{res}	n_{gmres}	cpu_{gmres}	$cpu_{\Delta t}$
h32p1	1	3	4.50	4	0.24	20.5
h16p2	1	3	4.61	4	0.58	25.9
h8p4	1	4	13.7	4	1.84	106.0
h32p1	2	3	2.20	5	0.21	11.6
h16p2	2	3	2.34	5	0.50	17.3
h8p4	2	4	7.90	6	2.28	90.5
h32p1	4	3	1.20	7	0.13	7.18
h16p2	4	3	1.18	8	0.45	15.0
h8p4	4	4	5.30	8	2.14	93.4
h32p1	8	3	0.66	10	0.10	5.79
h16p2	8	3	0.61	11	0.27	11.5
h8p4	8	4	4.00	12	2.36	127.5

Table 6.2: Simulation parameters for cases h32p1, h16p2 and h8p4 using 1,2,4 and 8 processors. All simulations employ TCPG scheme with $\Delta t = 0.002$ and $\varepsilon_{gmres} = \varepsilon_{update} = \varepsilon_{res} = 0.001$.

CPU time compared to cases h32p1 and h16p2. To explain this increase in CPU time with polynomial degree P at a fixed number of degrees of freedom, we consider the individual components constituting the work within a time step.

First, we focus on the residual evaluations. Both case h32p1 and h8p4 require 3 corrector passes while the case h8p4 requires 4 corrector passes. The CPU times measured for the residual evaluations for cases h32p1 and h16p2 are very similar, while that for case h8p4 is about three times larger. A simplistic approximation to the computational work involved in a single residual evaluation (CW_{res}) is given by

$$CW_{res} \approx n_{el} \times n_{eb} \times n_{ip}, \quad (6.2)$$

where n_{el} is the number of elements, n_{eb} is the number of element basis functions,

and n_{ip} is the number of integration points per element. It follows from (6.2) that the computational work scales linearly with the number of elements, but quadratically with the polynomial degree of the finite element basis functions. The latter is due to the fact that in each coordinate direction, there is at least one more integration point than element basis functions for Gaussian quadrature (Section 6.3). This is an important reason explaining why p -refinement is generally more expensive than h -refinement.

Next, we consider the contribution associated with the solution of the linear systems. All cases require four GMRES iterations to satisfy the convergence criterion. This implies that the ILU preconditioner is a robust preconditioner for both the low-order and higher-order polynomial bases considered. However, the CPU times per GMRES increase significantly as with the polynomial degree of the solution approximation. In particular, cpu_{gmres} for case h16p2 is approximately two times larger as that for h32p1, while that for case h8p4 is about 8 times larger. As discussed in the previous section, the main contribution to the work within a GMRES iteration is the matrix-vector product, for which the computational work (CW_{gmres}) can be approximated by

$$CW_{gmres} \approx n_{el} \times n_{eb} \times n_{eb}. \quad (6.3)$$

This relation reflects the fact that a higher-order polynomial basis leads to a denser system matrix (more entries per row), compared to a lower-order basis. Consequently, higher-order bases require considerably more work for a matrix-vector product for a given number of degrees of freedom, despite the lower number of elements.

The higher the polynomial order for a given number of degrees of freedom, the more memory is required to store the matrix and its ILU preconditioner. As the total memory required by the simulation exceeds what is locally available to the processor, additional memory has to be accessed which is physically at a different location within the cluster. This results in additional latencies, which depending on the particular infrastructure of the cluster, can significantly slow down the calculation.

Unfortunately, this is the case for h8p4.

Relative contributions

In Figure 6.5(a) we plot the contributions of the residual evaluation and the solution of the linear system to the total CPU time per time step obtained on a single processor. For all three spatial resolutions, the majority of the CPU time step is spent in the residual evaluation. For case h16p2 the time spent for the linear system solution is relatively small compared to that of the residual evaluation. Since the remaining parameters are equivalent to that of case h32p1, case h16p2 is only slightly more expensive.

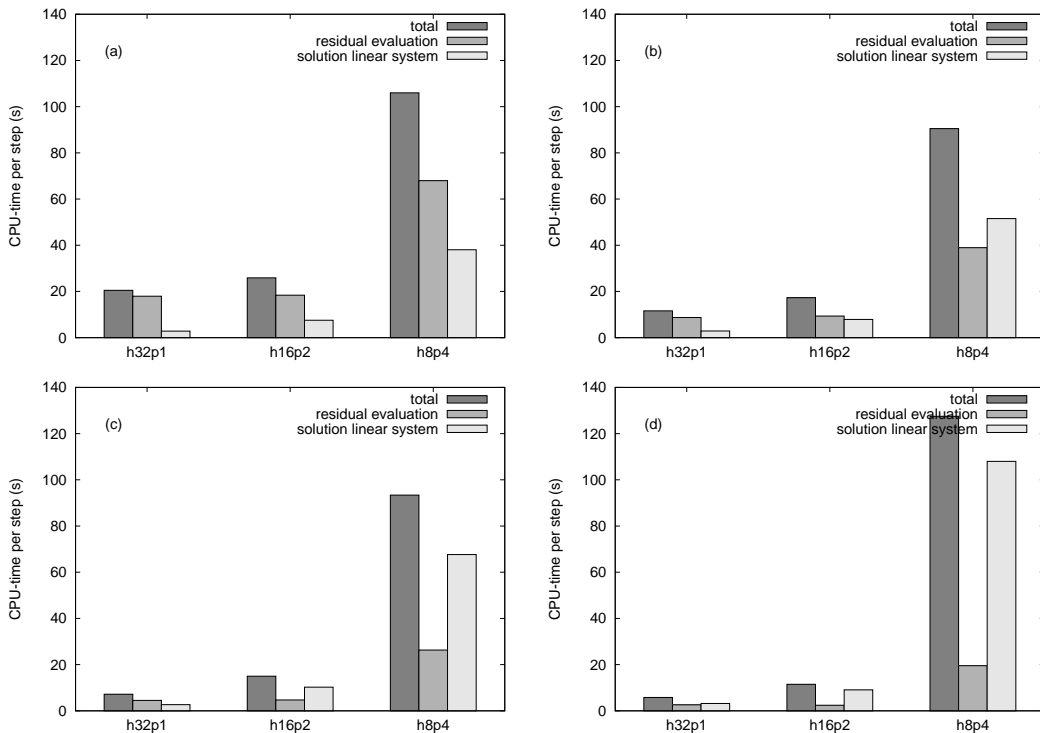


Figure 6.5: CPU times per time step for different hp -resolutions for turbulent channel flow at $\text{Re}_\tau = 180$ obtained using 1 processor (a); 2 processors (b); 4 processors (c); 8 processors (d).

Parallel computing

Next, we consider the performance obtained using multiple processors in parallel, as also shown in Table 6.2. Here, we employ the domain decomposition approach, in which the computational domain is partitioned and distributed over the processors. Subsequently, the problem is discretized and solved on each partition in parallel. An extensive discussion on parallel computing for finite element methods can be found in [47].

Obviously, the CPU time required for the residual evaluation decreases with the number of processors, as it is performed only on a partition of the domain for each processor. Note that when running on multiple processors, additional time is spent due to communication, required to include contributions to the residual vector from neighboring partitions.

As the number of processors increases, the efficiency of the ILU preconditioner decreases. This is due to the ILU decomposition which is then based on the Jacobian matrix corresponding to a partition of the domain rather than the entire domain. As a result, the ILU preconditioner can be viewed as a block ILU, which is a less efficient approximation to the inverse of the Jacobian matrix for the entire domain than the "full" ILU preconditioner. As shown in Figure 6.6, the decreased efficiency of the preconditioner is reflected by an almost linear increase in GMRES iterations with increasing number of processors for the cases under consideration. As follows from (6.3), the CPU time required for a single GMRES iteration scales linearly with the number of elements. Therefore, as the number of processors increases, and consequently the partitions contain a decreasing number of elements, the CPU time for required for the matrix-vector product of a GMRES iteration decrease correspondingly. However, the measured scalings of cpu_{gmres} are less than expected. Although cpu_{gmres} decreases with the number of processors for case h32p1 and h16p2, an increase is found for h8p4. There are a number of reasons for this poor scaling. We use

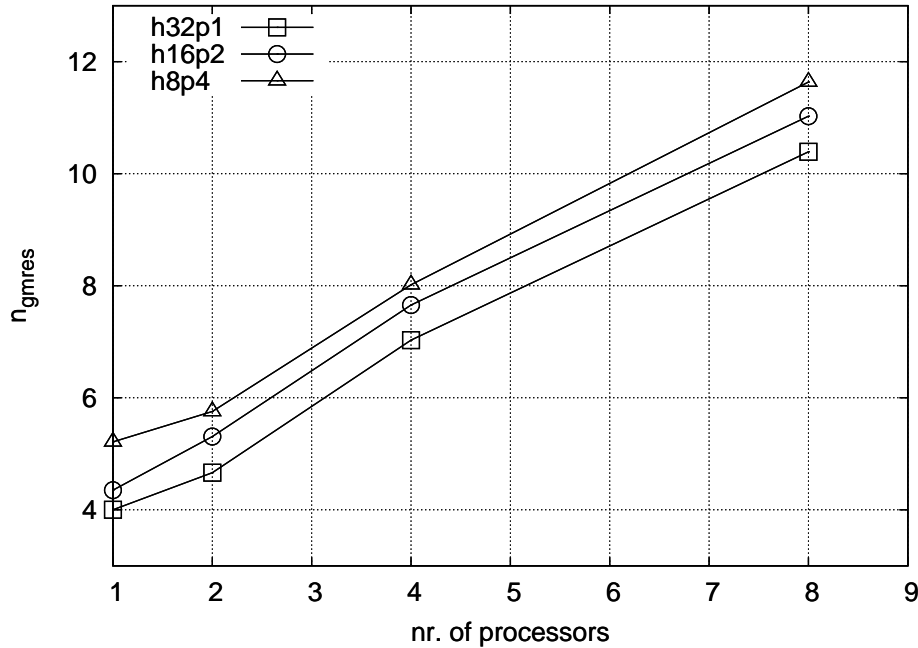


Figure 6.6: Illustration of the decreasing efficiency of the ILU preconditioner resulting in an increasing number of GMRES iterations required for convergence as the number of processors increases.

an extra overlap element at each boundary of the partition to account for contributions from neighboring partitions to the Jacobian matrix. If the number of elements is low and the order is high, such as typically the case for high-order methods, this has a relatively high impact on the work involved in a matrix-vector product. Another reason is the communication which is required after each GMRES iteration to account for contributions of neighboring partitions to the matrix-vector product. This is especially true for h8p4, where latencies are relatively high since this case requires more memory than locally available.

The contributions of the residual evaluation and linear system solution to the total CPU time per time step obtained with parallel computing on two, four and eight processors are shown in Figures 6.5(b)-(d). In all cases, as the number of processors increase, the contribution associated with the residual evaluation decreases, while

that of the solution of the linear system increases. For case h32p1 both contributions become equivalent when running on eight processors. For cases h16p2 and h8p4 this already happens with two processors, while for four and eight processors the linear system solution constitutes the dominant contribution to the total CPU time per time step.

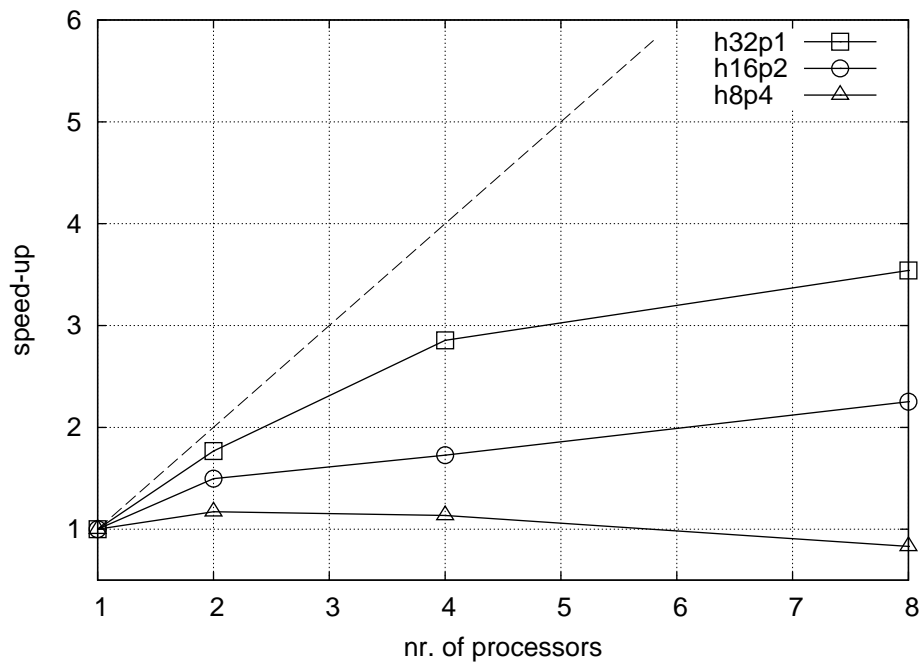


Figure 6.7: Scaling of CPU time per time step for parallel computing.

The total speed-up of the calculations when using parallel computing is shown in Figure 6.7. Case h32p1 has the best scaling with the number of processors, while h8p4 scales very poorly and even becomes slower when running on eight processors. Meanwhile, case h16p2 is somewhere in between, but does not exhibit negative scaling for the number of processors considered.

6.3 Numerical integration

The quadrature order used to integrate the terms that constitute the Jacobian matrix and residual vector is of significant consideration for large-scale problems since it is directly related to the computational efficiency of the method. To minimize the computational work associated with numerical integration, we employ Gauss-Legendre quadrature, which has the highest possible quadrature order [22] for a given number of integration points. The polynomial order that can be integrated exactly by this rule is equal to $2Q - 1$, where Q is the number of integration points in one dimension.

Insufficient quadrature may cause numerical difficulties, such as reduced accuracy, loss of convergence, or singular Jacobian matrices. On the other hand, if the simulation is well resolved, numerical crimes committed by insufficient quadrature are often negligible [40]. To ensure the theoretical rate of convergence for second order elliptic equations Heinrich and Pepper [22] state that the squares of the first derivatives of all basis functions must be integrated exactly by the quadrature rule. Although this criterion might be useful for ensuring convergence, it does not ensure the highest possible accuracy.

Kirby and Karniadakis [40] have investigated the effect of quadrature order on the stability of the solution. They found that insufficient quadrature of the non-linear terms in a Galerkin discretization effectively leads to polynomial aliasing errors. It was shown that these aliasing errors can be eliminated, and therefore stability of the solution can be enhanced, by integrating the non-linear terms in the variational statement with a higher quadrature order than the linear terms. Therefore, this technique is referred to as Polynomial De-aliasing (PD). Ramakrishnan and Collis [64] demonstrated the suitability of PD for enhancing stability for their discontinuous Galerkin method on coarse resolutions. Additionally, PD combined with ℓ VMS produced particularly good turbulence statistics.

When using primitive variables, the cubic non-linear terms in the compressible

Navier-Stokes equations lead to fourfold products of polynomials, which require $(4P+1)/2$ integration points in each spatial direction for exact integration. Additional care must be taken when primitive variables are employed. For example, linear terms including the total energy also leads to fourfold products due to the kinetic energy component $(\frac{1}{2}\rho u_i u_i)$.

We consider the effect of the quadrature order on the results of the TCPG method using $(16 \times 16 \times 16)$ elements with $P = \tilde{P} = 2$ and $\Delta t = 0.002$. The convergence criteria are set equal to $\varepsilon_{gmres} = \varepsilon_{update} = \varepsilon_{res} = 0.001$. So far, the results in this chapter for this resolution employed $(3 \times 3 \times 3 \times 2)$ integration points per element, where the last number corresponds to the temporal integration direction. Note that, although the Navier-Stokes equations are not purely elliptic, this scheme satisfies the criterion that it integrates the squares of the derivatives exactly. Although this scheme employs considerably less points (viz. 3^3 instead of 5^3) in the spatial directions than strictly necessary for exact integration of all terms, the simulation did not suffer from instabilities. In order to investigate the effect of PD, we increase the spatial quadrature order by employing $(4 \times 4 \times 4 \times 2)$ integration points. We have also investigated a reduced spatial quadrature order corresponding to $(2 \times 2 \times 2 \times 2)$ integration points. This integration scheme, however, lead to a singular Jacobian matrix.

Additionally, we examine the effect of reducing the temporal quadrature order by employing $(3 \times 3 \times 3 \times 1)$ integration points. Note that a single integration point in time corresponds to the midpoint rule, which is still second-order accurate (as is the TCPG method).

The mean flow and rms profiles obtained using increased spatial quadrature order as well as reduced temporal quadrature order are shown in Figure 6.8. As opposed to prior studies [64] on the effect of PD, no visible effect of the increased spatial quadrature orders on the mean flow is observed, while the effect on the rms profiles is small. Advantageously, reduced temporal quadrature order does not lead to sig-

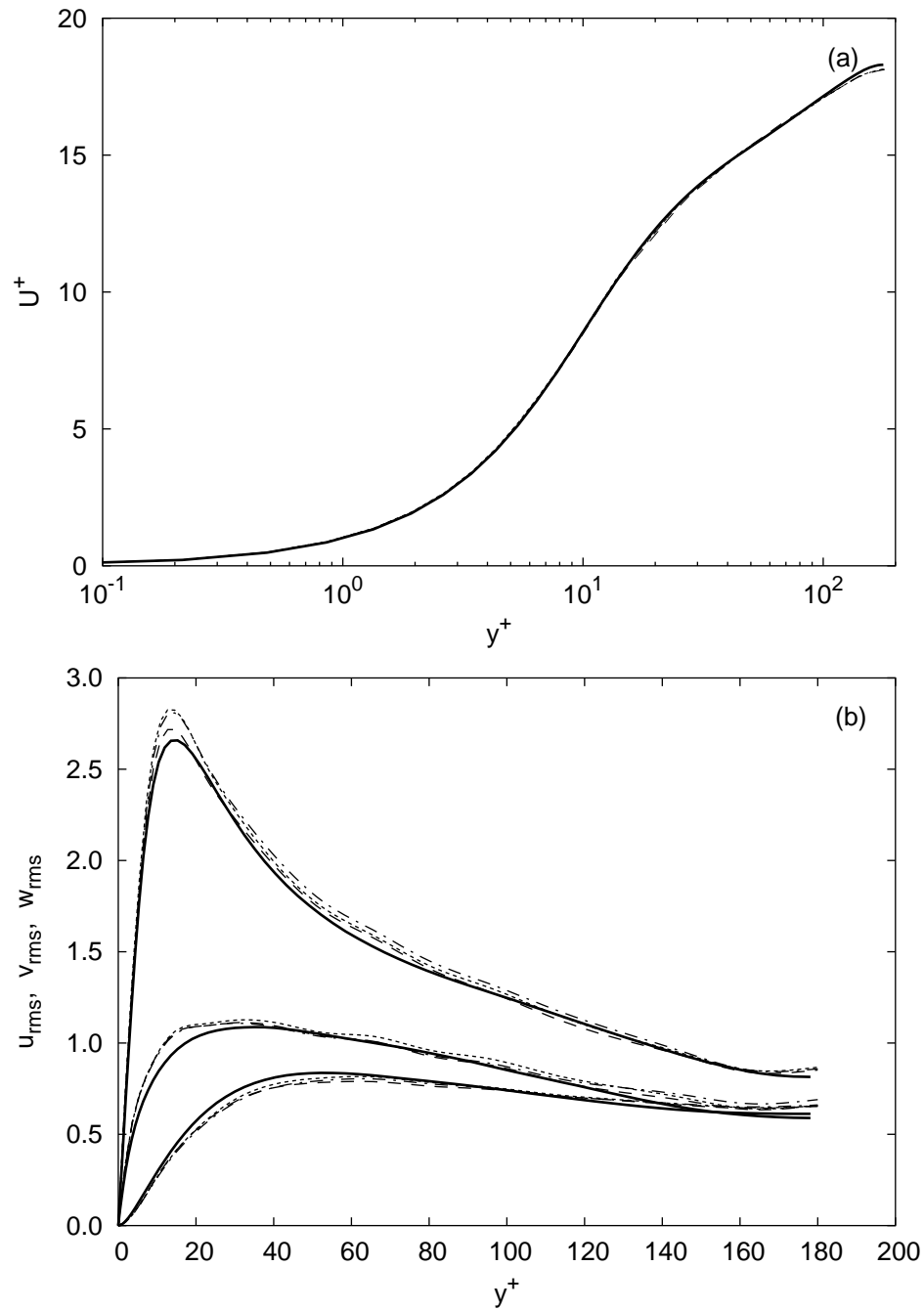


Figure 6.8: Influence of the quadrature order on the mean velocity and rms profiles for the TCPG method with $(16 \times 16 \times 16)$ elements and $P = \tilde{P} = 2$ for turbulent channel flow at $Re_\tau = 180$: — DNS [39]; - - - $(4 \times 4 \times 4 \times 2)$; - · - $(3 \times 3 \times 3 \times 2)$; ···· $(3 \times 3 \times 3 \times 1)$.

nificant loss of accuracy in both mean and rms profiles. Although not shown here, the criterion for the quadrature rule stating that the square of the derivatives of the basis functions should be integrated exactly, also proved to be effective for all results presented in the previous chapter.

From (6.2) it follows that the increased spatial quadrature order increases the CPU time for a residual evaluation by approximately a factor $(4/3)^3 \approx 3$. As can be seen from Figure 6.5, this significantly increases the simulation time, as the residual evaluation then becomes a significant contribution for all number of processors. The reduced temporal quadrature order, on the other hand, decreases the residual-evaluation time by a factor two, while preserving accuracy and stability.

6.4 Convergence criteria

The convergence criteria of the predictor multi-corrector algorithm are directly related to the efficiency of the present method. In general, these should be chosen such that the both linear and non-linear system are not solved with excessive precision. In Section 6.4.1 we examine the implications of convergence criterion for the solution of the linear system, while in Section 6.4.2 we consider the non-linear convergence criteria. We restrict ourselves again to a spatial resolution of $(16 \times 16 \times 16)$ elements with $P = \tilde{P} = 2$ and $\Delta t = 0.002$. The cases considered in this section are all performed using four processors in parallel.

6.4.1 Linear system

We consider the following tolerances for the GMRES algorithm: $\varepsilon_{gmres} = 0.001, 0.01$ and 0.1 , while the non-linear convergence criterion is fixed at $\varepsilon_{update} = \varepsilon_{res} = 0.001$. Note that, although the solution of the linear system is less accurate as the ε_{gmres} is reduced, the non-linear convergence criterion is still satisfied.

The mean velocity, shown in Figure 6.9(a) is not noticeably affected by the range of linear convergence criteria considered, the effect on rms profiles, shown in Figure 6.9(b), is very small.

ε_{gmres}	n_{gmres}	n_{corr}	$\text{cpu}_{\Delta t}$
0.001	8	3	15.0
0.01	5	3	11.5
0.1	3	4	11.4

Table 6.3: Effect of convergence criterion for the linear system solution. The CPU times are obtained using four processors in parallel and $\varepsilon_{update} = \varepsilon_{res} = 0.001$.

The number of GMRES iterations required corresponding to the convergence criteria as well as the number of corrector passes are shown in Table 6.3. As ε_{gmres} is reduced from 0.001 to 0.01 an additional corrector pass is needed for non-linear convergence. Recall that the solution of the linear system represents the Newton update. Obviously, reduced accuracy in the update eventually leads to an increased number of corrector passes to satisfy the non-linear convergence criteria. Therefore, given similar solution quality, one has to consider if cheaper corrector passes are advantageous, compared to a lower number of more expensive corrector passes. This depends on the cost of the residual evaluations relatively to that of the linear system solution, which is a function of the number of processors (see Figure 6.5). Moreover, it should be ensured that the increased number of corrector passes does not lead to a net increase in GMRES iterations per time step. For the specific case of using four processors, the CPU time per time step (see Table 6.3), levels off going from $\varepsilon_{gmres} = 0.01$ to 0.1.

6.4.2 Non-linear system

For the solution of the non-linear system within each time step, we consider the following convergence criteria: $\varepsilon_{update} = \varepsilon_{res} = 0.0001$, 0.001 , and 0.01 . The tolerance for the linear system is set equal to $\varepsilon_{gmres} = 0.01$.

We note that no adverse effect on the stability is observed for the range of non-linear tolerances considered. Moreover, as seen in Figure 6.10, the nonlinear convergence criteria considered do not have a significant influence on both the mean velocity and rms profiles.

$\varepsilon_{res}, \varepsilon_{update}$	n_{corr}	n_{gmres}	$cpu_{\Delta t}$
0.0001	4	8	20.3
0.001	3	8	15.0
0.01	2	8	10.7

Table 6.4: Effect of non-linear convergence criterion. The CPU times are obtained using four processors in parallel and $\varepsilon_{gmres} = 0.001$.

The number of corrector passes and GMRES iterations are shown in Table 6.4, along with the CPU time per time step obtained on four processors. Increasing the tolerance by a factor ten decreases the number of required corrector passes by one. In other words, in each corrector pass the non-linear residual is decreased by an order of magnitude. Moreover, the number of corrector passes is low for all cases. This clearly demonstrates the efficiency of the constant Jacobian matrix. Recall that in Section 4.6.3 it was claimed that the constant Jacobian approach can have much weaker non-linear convergence properties than the Jacobian-free approach. These results imply that this is not the case for the present problem. Although the CPU time per step depends on the number of processors, the trend will be similar for different number of processors since the number corrector passes remains the same. It follows from (6.1) that the CPU time per time step scales approximately linearly with the number

of corrector passes, as confirmed by the CPU times for the present cases. Including the decreased CPU time that can be achieved by increasing the linear convergence tolerance, the computation time can be reduced by at least more than a factor two compared to the finest tolerances considered.

6.5 Summary

We have investigated the main factors that influence the accuracy and computational work for the numerical simulation of turbulent channel flow at $\text{Re}_\tau = 180$.

First, we compared the different time discretizations. The TCG method appeared to be incapable of producing results at time steps of interest, while the TDG and TCPG method produced accurate results of similar quality. The TDG method is significantly more expensive than the TCPG method, however, because it employs twice the number of unknowns. Therefore, for reasons other than adaption, the use of TCPG is preferred.

Next, we compared the accuracy and computational work obtained using different spatial hp -resolutions for a similar number of degrees of freedom. It was shown that the use of higher-order methods ($P > 1$) can lead to increased accuracy as they can exploit the advantages of the VMS method (Chapter 5). However, a major disadvantage is that the computational work scales quadratically with the number of element basis functions. As these increase rapidly with the polynomial degree P , high-order methods are inherently expensive. Additionally, the Jacobian matrix for high-order methods are significantly more dense, resulting in increased memory requirements for its storage. If the required memory exceeds what is locally available to the processor, the simulation is considerably slowed down due to additional latency effects. An additional difficulty with the present implementation is its poor parallel scaling with increased polynomial degree P . This is mainly due to the use of overlap elements used to account for contributions of neighboring partitions to the Jacobian

matrix of each partition. As higher-order methods typically use low numbers of elements, these overlap elements constitute a relatively large contribution to the computational work.

The quadrature order is of significance consideration in large-scale computations. Stable solutions were found if the quadrature order is chosen such that the square of the derivatives can be integrated exactly. Choosing higher quadrature orders did not lead to increased accuracy of the lower-order statistics.

Finally, the influence of the convergence criteria of both the solution of the linear system (ε_{gmres}) and the non-linear system ($\varepsilon_{update}, \varepsilon_{res}$) was investigated. These should be chosen such that the linear and non-linear system are not solved with excessive precision. It was found that $\varepsilon_{gmres} = 0.01$ and $\varepsilon_{update} = \varepsilon_{res} = 0.01$ are sufficient for the turbulent channel flow case considered.

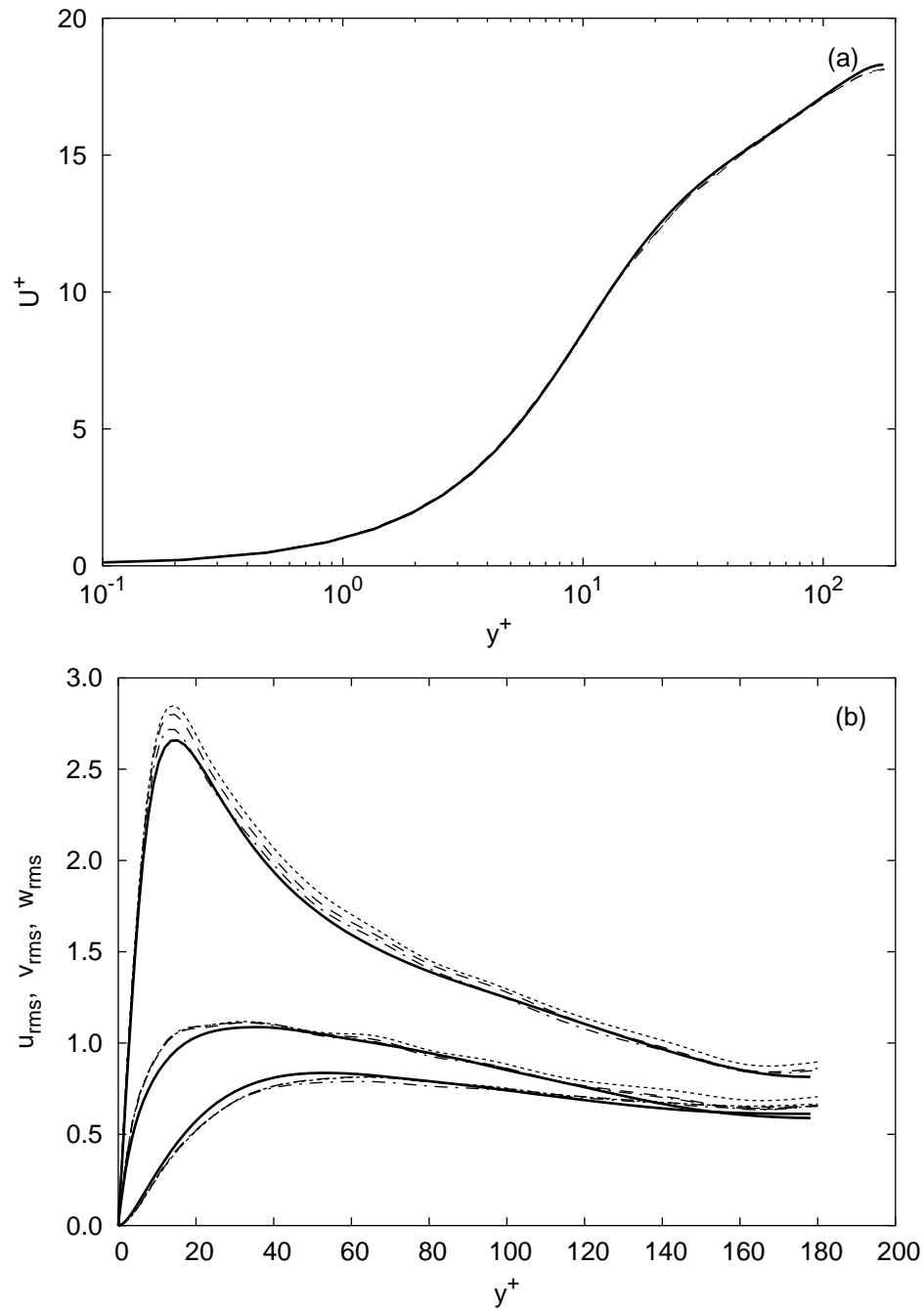


Figure 6.9: Influence of the linear convergence criteria on the mean velocity and rms profiles for the TCPG method with $(16 \times 16 \times 16)$ elements and $P = \tilde{P} = 2$ for turbulent channel flow at $\text{Re}_\tau = 180$: — DNS [39]; - - - $\varepsilon_{gmres} = 0.001$; - · - $\varepsilon_{gmres} = 0.01$; ····· $\varepsilon_{gmres} = 0.1$.

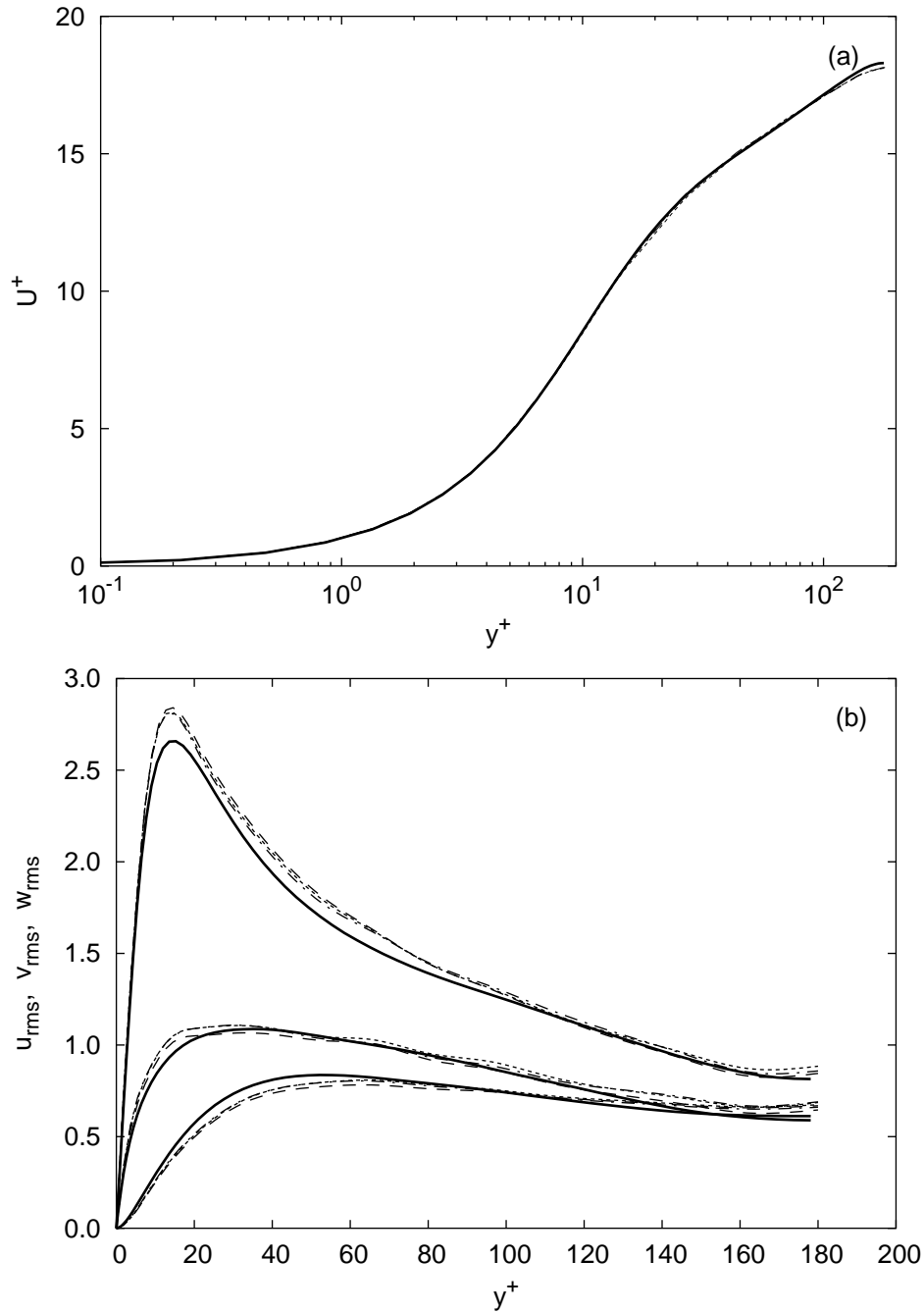


Figure 6.10: Influence of the non-linear convergence criteria on the mean velocity and rms profiles for the TCPG method with $(16 \times 16 \times 16)$ elements and $P = \tilde{P} = 2$ for turbulent channel flow at $Re_\tau = 180$: — DNS [39]; - - - $\epsilon_{update} = \epsilon_{res} = 0.0001$; - · - $\epsilon_{update} = \epsilon_{res} = 0.001$; ···· $\epsilon_{update} = \epsilon_{res} = 0.01$.

Chapter 7

Conclusions and Recommendations

In Section 7.1 we highlight the main conclusions of the present research and we conclude with some recommendations for future research directions inspired by the findings of this work in Section 7.2.

7.1 Conclusions

We have proposed and implemented a space-time VMS discretization for the compressible Navier-Stokes equations, which can be applied to complex flow configurations. The capability of the VMS discretization was examined for turbulent channel flow at $Re_\tau = 180$. The results obtained using the VMS method were compared to those obtained using no SGS model as well as those from an LES formulation using SGS modeling on all scales.

Overall, the results obtained using no SGS modeling suffer from insufficient dissipation. This leads to excess energy content in the large scales, which results in poor-quality turbulence statistics. This implies that SGS modeling is required to enhance energy dissipation. Traditional LES formulations, however, structurally lead to excessive dissipation. These allow the dynamically important structures in the flow to be directly influenced by the SGS model, resulting in poor turbulence statistics.

In contrast, the VMS method was found to be capable of producing good quality low-order turbulence statistics at relatively coarse levels of refinement. In the VMS method, the direct influence of the SGS model is restricted to the small scales, while the large scales are left free from direct modeling effects. The crucial parameter for obtaining reasonable turbulence statistics was found to be the large/small partition combined with the size of the large and small scale spaces. In general, the basic dynamics of the near-wall coherent structures should be released from the direct influence of the SGS model. It was found that to do so, modes capable of representing the spanwise-alternating velocity profile typical of near-wall streaks should be contained in the large-scale space.

Additionally, a serendipity variant of the spatial discretization was examined, in which only those modes are retained that are necessary for a certain order of accuracy. For similar resolutions (i.e. number of elements and polynomial order), the serendipity discretization produced reasonable mean velocity profiles, but the turbulence intensities and Reynolds stress were of poor quality. Due to the reduced number of modes, the large-scale space is too small to allow an adequate description of the large scales, while the small-scale space is too small to provide sufficient dissipation.

A detailed performance analysis of the proposed VMS discretization was conducted, in which we examined the factors influencing the accuracy and computational cost for the simulation of turbulent channel flow.

First, we compared the different time discretizations. The TCG method is not a competitive method as it failed to obtain convergence for time steps of interest. The TDG and TCPG methods, on the other hand, produced accurate and very similar results for relatively large time steps. However, the TDG method is considerably more computationally expensive than the TCPG method, since it used twice the number of degrees of freedom. Therefore, if no adaptation is required, the TCPG

method is preferred.

Next, we compared the accuracy and cost of different spatial resolution employing a similar total number of degrees of freedom. It is shown that the spatially higher-order methods employing the VMS approach lead to increased accuracy compared to a standard low-order (linear) method, where all scales are necessarily directly affected by the SGS model. Higher-order methods, however, are inherently expensive as the computational work scales quadratically with the number of finite-element basis functions. In addition, higher-order methods lead to denser system matrices, resulting in rapidly increasing memory requirements with the order of the scheme.

The quadrature order used for numerical integration is of great importance in large-scale simulations as it is directly related to computational efficiency. It was shown that, even at coarse resolutions, considerably lower quadrature orders can be used than strictly necessary for exact integration, without seriously degrading accuracy.

In conclusion, it appears that the present discretization, with correctly chosen parameters, is a suitable candidate for the large eddy simulation of turbulent flows over complex deforming domains. The VMS method has shown clear advantages as it leads to reduced resolution requirements compared to the traditional LES approach using a similar SGS model. As the computational cost associated with higher-order methods is still relatively high, additional research has to be conducted in order to increase efficiency.

7.2 Recommendations

The proposed VMS discretization has proven its capability of reducing the resolution requirements for reasonable turbulence statistics compared to traditional LES formulations. There is considerable evidence [3, 64] that the resolution requirements can be

further reduced by a weak enforcement of the boundary conditions. These techniques may therefore be expected to further improve the computational efficiency.

The parallel scaling of the present implementation was found to be poor. This is partly due to the decreasing efficiency of the preconditioner with increasing number of processors. Currently, deflation techniques are under development that counteract this loss of efficiency. Moreover, alternative implementations should be considered that do not require overlap elements.

The present implementation also requires large amounts of memory if the matrix and its preconditioner are stored. Excessive memory use can significantly increase the length of the simulation. To avoid this problem, pseudo-time formulations, potentially coupled with multigrid techniques [77], are promising candidates. These allow the implicit problem to be solved in a fully explicit way, and therefore completely avoid the need to solve large linear systems of equations.

Bibliography

- [1] S.K. Aliabadi, T.E. Tezduyar. *Space-time finite element computation of compressible flows involving moving boundaries and interfaces*. Computer Methods in Applied Mechanics and Engineering, **107** (1993), 209–223.
- [2] A.K. Aziz, P. Monk. *Continuous finite elements in space and time for the heat equation*. Mathematics of Computation, **52** (1989), 255–274.
- [3] Y. Bazilevs, T.J.R. Hughes. *Weak imposition of Dirichlet boundary conditions in fluid mechanics*. ICES Report, 05-25, 2005.
- [4] G.J. Le Beau, S.E. Ray, S.K. Aliabadi, T.E. Tezduyar. *SUPG finite element computation of compressible flows with the entropy and conservation variables formulations*. Computer Methods in Applied Mechanics and Engineering, **104** (1993), 397–422.
- [5] G.L. Brown, A. Roshko. *On density effects and large structure in turbulent mixing layers*. Journal of Fluid Mechanics, **64** (1974), 775–816.
- [6] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988.
- [7] Y. Chang. *Approximate models for optimal control of turbulent boundary layers*. Ph.D. Thesis, Rice University, 2000.

-
- [8] Y. Chang, S.S. Collis, S. Ramakrishnan. *Viscous effects in control of near-wall turbulence*. *Physics of Fluids*, **14** (2002), 4069–4080.
- [9] T. Chisholm, D. W. Zing. *A fully coupled Newton-Krylov solver for turbulent aerodynamic flows*. ICAS Congress, 2002.
- [10] S.S. Collis. *Monitoring unresolved scales in multiscale turbulence modeling*. *Physics of Fluids*, **13** (2001), 1800–1806.
- [11] S.S. Collis. *Multiscale Methods for Turbulence Simulation and Control*. 32nd Computational Fluid Dynamics Lecture Series, von Karman Institute, Belgium. (MEMS preprint no. 2002-034), 2002.
- [12] S.S. Collis. *The DG/VMS method for unified turbulence simulation*. AIAA paper 2002-3124, 2002.
- [13] S.S. Collis, K. Ghayour. *Discontinuous Galerkin methods for compressible DNS*. Proceedings of FEDSM, ASME/JSME Joint Fluids Engineering Conference, 2003.
- [14] S.S. Collis, S. Ramakrishnan. *The local variational multiscale method*. Third MIT Conference on Computational Fluid and Solid Dynamics, June 14-17, 2005.
- [15] J. Donea, A. Huerta, J.-Ph. Ponthot, A. Rodríguez-Ferran. *Arbitrary Lagrangian-Eulerian methods*. Encyclopedia of Computational Mechanics, Volume 1: Fundamentals. John Wiley & Sons Ltd., 2004.
- [16] C. Farhat, P. Geuzaine, C. Grandmont. *The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids*. *Journal of Computational Physics*, **174** (2001), 669–694.

- [17] C. Farhat, A. Rajasekharan, B. Koobus. *A dynamic variational multiscale method for large eddy simulations on unstructured grids*. Computer Methods in Applied Mechanics and Engineering, **195** (2006), 1667–1691.
- [18] M. Germano, U. Piomelli, P. Moin, W.H. Cabot. *A dynamic subgrid-scale model*. Physics of Fluids, **3** (1991), 1760–1765.
- [19] P. Geuzaine. *Newton-Krylov Strategy for compressible turbulent flows on unstructured meshes*. AIAA Journal, **39** (2000), 528–531.
- [20] P. Hansbo. *A crank-Nicolson type space-time finite element method for computing on moving meshes*. Journal of Computational Physics, **159** (2000), 274–289.
- [21] G. Hauke, T.J.R. Hughes. *A comparative study of different sets of variables for solving compressible and incompressible flows*. Computer Methods in Applied Mechanics and Engineering, **153** (1998), 1–44.
- [22] J.C. Heinrich, D.W. Pepper. *Intermediate Finite Element Method: Fluid Flow and Heat Transfer Applications*. Taylor & Francis, 1999.
- [23] J. Holmen, T.J.R. Hughes, A.A. Oberai, G.N. Wells. *Sensitivity of the scale partition for variational multiscale LES of channel flow*. Physics of Fluids, **16** (2004), 824–827.
- [24] T.J.R. Hughes, G.M. Hulbert. *Space-time finite element methods for elastodynamics: formulations and error estimates*. Computer Methods in Applied Mechanics and Engineering, **66** (1988), 305–328.
- [25] T.J.R. Hughes, L.P. Franca, G.M. Hulbert. *A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations*. Computer Methods in Applied Mechanics and Engineering, **73** (1989), 173–189.

-
- [26] T.J.R. Hughes. *Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods*. Computer Methods in Applied Mechanics and Engineering, **127** (1995), 387–401.
- [27] T.J.R. Hughes. *A space-time formulation for multiscale phenomena*. Journal of Computational and Applied Mathematics, **74** (1996), 217–229.
- [28] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.B. Quincy. *The variational multiscale method – a paradigm for computational mechanics*. Computer Methods in Applied Mechanics and Engineering, **166** (1998), 3–24.
- [29] T.J.R. Hughes, L. Mazzei, K.E. Jansen. *Large Eddy Simulation and the Variational Multiscale Method*. Computing and Visualization in Science, **3** (2000), 47–59.
- [30] T.J.R. Hughes, L. Mazzei, A.A. Oberai. *The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence*. Physics of Fluids, **13** (2001), 505–512.
- [31] T.J.R. Hughes, A.A. Oberai, L. Mazzei. *Large eddy simulation of turbulent channel flow by the variational multiscale method*. Physics of Fluids, **13** (2001), 1784–1799.
- [32] G.M. Hulbert, T.J.R. Hughes. *Space-time finite element methods for second-order hyperbolic equations*. Computer Methods in Applied Mechanics and Engineering, **84** (1990), 327–348.
- [33] B.L. Hulme. *One-step piecewise polynomial Galerkin methods for initial value problems*. Mathematics of Computation, **26** (1972), 415.

- [34] K.E. Jansen, S.S. Collis, C.H. Whiting, F. Shakib. *A better consistency for low-order stabilized finite element methods* Computer Methods in Applied Mechanics and Engineering, **174** (1999), 153–170.
- [35] K.E. Jansen, A.E. Tejada-Martínez. *An evaluation of the variational multiscale method for large-eddy simulation while using a hierarchical basis*. AIAA paper 2002-0283, 2002.
- [36] Z. Johan, T.J.R. Hughes, K.K. Mathur, S.L. Johnsson. *A data parallel finite element method for computational fluid dynamics on the Connection Machine system*. Computer Methods in Applied Mechanics and Engineering, **99** (1991), 113-134.
- [37] C. Johnson, U. Navert, J. Pitkaranta. *Finite element methods for linear hyperbolic problems*. Computer Methods in Applied Mechanics and Engineering, **45** (1984), 285-312.
- [38] E.M. Karniadakis, S.J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.
- [39] J. Kim, P. Moin, R. Moser. *Turbulence statistics in fully developed channel flow at low Reynolds number*. Journal of Fluid Mechanics, **177** (1987), 133-166.
- [40] R.M. Kirby, G.E. Karniadakis. *De-aliasing on non-uniform grids: algorithms and applications*. Journal of Computational Physics, **191** (2003), 249–264.
- [41] S.J. Kline, W.C. Reynolds, F.A. Schraub, P.W. Rundstadtler. *The structure of turbulent boundary layers*. Journal of Fluid Mechanics, **30** (1967), 741–773.
- [42] D.A. Knoll, D.E. Keyes. *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*. Journal of Computational Physics, **193** (2004), 357–397.

- [43] B. Koobus, C. Farhat. *A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes – application to vortex shedding*. Computer Methods in Applied Mechanics and Engineering, **193** (2004), 1367–1383.
- [44] M. Lesoinne, C. Farhat. *Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations*. Computer Methods in Applied Mechanics and Engineering, **134** (1996), 71–90.
- [45] V. Levasseur, P. Sagaut, F. Chalot, A. Davroux. *An entropy-variable-based VMS/GLS method for the simulation of compressible flows on unstructured grids*. Computer Methods in Applied Mechanics and Engineering, **195** (2006), 1154–1179.
- [46] D.K. Lilly. *The length scale for subgrid-scale parametrization with anisotropic resolution*. Center for Turbulence Research, Stanford University/NASA Ames Research Center, 1988. CTR Annual Research Briefs, 3–9.
- [47] F.J. Lingen. *Design of an object oriented finite element package for parallel computers*. Ph.D. Thesis, Delft University of Technology, 2000.
- [48] H. Lomax, T.H. Pulliam, D.W. Zingg. *Fundamentals of Computational Fluid Dynamics*. Scientific Computation, Springer-Verlag, 2001.
- [49] N.P.C. Marques, J.C.F. Pereira. *Comparison of matrix-free acceleration techniques in compressible Navier-Stokes equations*. International Journal for Numerical Methods in Engineering, **61** (2004), 455–474.
- [50] M.P. Martin, U. Piomelli, G.V. Candler. *Subgrid-scale models for compressible large-eddy simulations*. Theoretical and Computational Fluid Dynamics, **13** (2000), 361–376.

- [51] A. Masud, T.J.R. Hughes. *A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems*. Computer Methods in Applied Mechanics and Engineering, **146** (1997), 91–126.
- [52] J. Mathieu, J. Scott. *An Introduction to Turbulent Flow*. Cambridge University Press, 2000.
- [53] P. Moin, J. Kim. *Numerical investigation of turbulent channel flow*. Journal of Fluid Mechanics, **118** (1982), 341–371.
- [54] F.T.M. Nieuwstadt. *Turbulentie. Theorie en toepassingen van turbulente stromingen*. Epsilon Uitgaven, 1998.
- [55] A.A. Oberai, T.J.R. Hughes. *The variational multiscale formulation of LES: Channel flow at $Re=590$* . AIAA paper, 2002-1056, 2002.
- [56] A. Pinelli, C. Benocci. *Large eddy simulation of a fully developed turbulent plane channel flow*. Von Karman Institute for Fluid Dynamics, Technical Note 171, 1989.
- [57] U. Piomelli, E. Balaras. *Wall-layer models for large-eddy simulations*. Annual Review of Fluid Mechanics, **34** (2002), 349-374.
- [58] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [59] N. Qin, D.K. Ludlow, S.T. Shaw. *A matrix-free preconditioned Newton/GMRES method for unsteady Navier-Stokes solutions*. International Journal for Numerical Methods in Fluids, **33** (2000), 223–248.
- [60] S. Ramakrishnan, S.S. Collis. *Variational multiscale modeling for turbulence control*. AIAA paper 2002-3280, 2002.
- [61] S. Ramakrishnan, S.S. Collis. *Multiscale modeling for turbulence simulation in complex geometries*. AIAA paper 2004-0241, 2004.

-
- [62] S. Ramakrishnan, S.S. Collis. *Turbulence control using the variational multiscale method*. AIAA Journal, **42** (2004), 745–753.
- [63] S. Ramakrishnan, S.S. Collis. *Partition selection in the multi-scale turbulence modeling*. Submitted to Physics of Fluids, 2004.
- [64] S. Ramakrishnan, S.S. Collis. *The Local Variational Multi-Scale Method for Turbulence Simulation*. Sandia Report, SAND2005-2733, 2005.
- [65] P.J.Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.
- [66] Y. Saad, M.H Schultz. *GMRES: a Generalized Minimal Residual algorithm for solving nonsymmetric linear systems*. SIAM Journal on Scientific and Statistical Computing, **7** (1986), 856–869.
- [67] Y. Saad. *Iterative solution techniques for large linear systems*. PWS Publishing Company, 1996.
- [68] P. Sagaut. *Large eddy simulation for incompressible flows: An introduction*. Scientific Computing, Springer-Verlag, 1998.
- [69] H. Schlichting, K. Gersten. *Boundary Layer Theory*. 8th Revised and Enlarged Edition, Springer-Verlag, 2000.
- [70] G. Scovazzi. *Multiscale methods in science and engineering*. Ph.D. Thesis, Stanford University, 2004.
- [71] F. Shakib. *Finite element analysis of the compressible euler and navier-Stokes equations*. Ph.D. Thesis, Stanford University, 1988.
- [72] F. Shakib, T.J.R. Hughes, Z. Johan. *A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations*. Computer Methods in Applied Mechanics and Engineering, **89** (1991), 141–219.

-
- [73] J. Smagorinsky. *General circulation experiments with the primitive equations. I. The basic experiment*. Monthly Weather Review, **91** (1963), 99–164.
- [74] A.E. Tejada-Martínez. *Dynamic subgrid-scale modeling for large-eddy simulation of turbulent flows with a stabilized finite element method*. Ph.D. Thesis, Rensselaer Polytechnic Institute, 2002.
- [75] T.E. Tezduyar, M. Behr, J. Liou. *A new strategy for finite element computations involving moving boundaries and interfaces – The Deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests*. Computer Methods in Applied Mechanics and Engineering, **94** (1992), 339–351.
- [76] M.D. Tidriri. *Preconditioning techniques for the Newton-Krylov solution of compressible flow*. Journal of Computational Physics, **132** (1997), 51–61.
- [77] J.J.W. van der Vegt, H. van der Ven. *Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows*. Journal of Computational Physics, **182** (2002), 546–585.
- [78] A.W. Vreman. *The filtering analog of the variational multiscale method in large-eddy simulation*. Physics of Fluids, **15** (2003), L61–L64.
- [79] W. Wang. *Coupled compressible and incompressible finite volume formulations for the large eddy simulation of turbulent flow with and without heat transfer*. Ph.D. Thesis, Iowa State University, Aerospace Engineering and Engineering Mechanics, 1995.
- [80] F.M. White *Viscous Fluid Flow*. McGraw-Hill, Second Edition, 1991.
- [81] D.C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc., Third Edition, 1998.

- [82] R. Winther. *A stable finite element method for initial-boundary value problems for first order hyperbolic systems*. Mathematics of Computation, **36** (1981), 45.
- [83] A. Yoshizawa. *Statistical theory for compressible turbulent shear flows, with the application to subgrid scale modeling*. Physics of Fluids A, **29** (1986), 2152–2164.
- [84] O.C. Zienkiewicz. *The Finite Element Method*. McGraw-Hill, Third Edition, 1977.

Appendix A

Coefficient matrices for the Navier-Stokes equations

The compressible Navier-Stokes equations in conservation form can be written as

$$\mathbf{U}_{,t} + \mathbf{F}_{i,i} - \mathbf{F}_{i,i}^v = \mathbf{S},$$

where \mathbf{U} is the vector of conservative variables, $\mathbf{F}_i(\mathbf{U})$ and $\mathbf{F}_i^v(\mathbf{U})$ are the inviscid and viscous flux vectors defined as

$$\mathbf{U} = \rho \begin{pmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{pmatrix}, \quad \mathbf{F}_i(\mathbf{U}) = u_i \mathbf{U} + p \begin{pmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{pmatrix}, \quad \mathbf{F}_i^v(\mathbf{U}) = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ \tau_{ij} u_j - q_i \end{pmatrix},$$

and \mathbf{S} the vector containing external sources. The quasi-linear form of the Navier-Stokes equations is given by

$$\mathbf{U}_{,t} + \mathbf{A}_i \mathbf{U}_{,i} - (\mathbf{K}_{ij} \mathbf{U}_{,j})_{,i} = \mathbf{S},$$

where $\mathbf{A}_i = \mathbf{F}_{i,\mathbf{U}}$ are the inviscid flux Jacobians and \mathbf{K}_{ij} are the diffusivity matrices, satisfying $\mathbf{K}_{ij} \mathbf{U}_{,j} = \mathbf{F}_i^v$. Introducing the transformation of variables $\mathbf{U} = \mathbf{U}(\mathbf{Y})$,

the Navier-Stokes equations can be solved in any set of variables \mathbf{Y} [21]. Using this transformation, the quasi linear form can be written in terms of \mathbf{Y} as

$$\tilde{\mathbf{A}}_0 \mathbf{Y}_{,t} + \tilde{\mathbf{A}}_i \mathbf{Y}_{,i} - (\tilde{\mathbf{K}}_{ij} \mathbf{Y}_{,j})_{,i} = \mathbf{S},$$

where

$$\begin{aligned}\tilde{\mathbf{A}}_0 &= \mathbf{U}_{,\mathbf{Y}}, \\ \tilde{\mathbf{A}}_i &= \mathbf{A}_i \tilde{\mathbf{A}}_0, \\ \tilde{\mathbf{K}}_{ij} &= \mathbf{K}_{ij} \tilde{\mathbf{A}}_0.\end{aligned}$$

In the remainder of this appendix, these coefficient matrices are given for conservative variables, and primitive density and pressure variables.

A.1 Conservative variables

Flux Jacobians

The flux Jacobians for conservative variables, i.e. $\mathbf{Y} = \mathbf{U}$ are given by

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -u_1^2 + \frac{\gamma-1}{2} \|\mathbf{u}\|^2 & (3-\gamma)u_1 & (1-\gamma)u_2 & (1-\gamma)u_3 & \gamma-1 \\ -u_1 u_2 & u_2 & u_1 & 0 & 0 \\ -u_1 u_3 & u_3 & 0 & u_1 & 0 \\ -u_1[\gamma e - (\gamma-1)\|\mathbf{u}\|^2] & \gamma e - \frac{\gamma-1}{2}(\|\mathbf{u}\|^2 + 2u_1^2) & (1-\gamma)u_1 u_2 & (1-\gamma)u_1 u_3 & \gamma u_1 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -u_1 u_2 & u_2 & u_1 & 0 & 0 \\ -u_2^2 + \frac{\gamma-1}{2} \|\mathbf{u}\|^2 & (1-\gamma)u_1 & (3-\gamma)u_2 & (1-\gamma)u_3 & \gamma-1 \\ -u_2 u_3 & 0 & u_3 & u_2 & 0 \\ -u_2[\gamma e - (\gamma-1)\|\mathbf{u}\|^2] & (1-\gamma)u_1 u_2 & \gamma e - \frac{\gamma-1}{2}(\|\mathbf{u}\|^2 + 2u_2^2) & (1-\gamma)u_2 u_3 & \gamma u_2 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -u_1 u_3 & u_3 & 0 & u_1 & 0 \\ -u_2 u_3 & 0 & u_3 & u_2 & 0 \\ -u_3^2 + \frac{\gamma-1}{2} \|\mathbf{u}\|^2 & (1-\gamma)u_1 & (1-\gamma)u_2 & (3-\gamma)u_3 & \gamma-1 \\ -u_3[\gamma e - (\gamma-1)\|\mathbf{u}\|^2] & (1-\gamma)u_1 u_3 & (1-\gamma)u_2 u_3 & \gamma e - \frac{\gamma-1}{2}(\|\mathbf{u}\|^2 + 2u_3^2) & \gamma u_3 \end{bmatrix}$$

Diffusivity matrices

Let $\chi = \lambda + 2\mu$, then the diffusivity matrices for conservative variables are given by

$$\mathbf{K}_{11} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\chi u_1 & \chi & 0 & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 & 0 \\ \frac{\kappa}{c_v}(-e + \|\mathbf{u}\|^2) - \chi u_1^2 - \mu(u_2^2 + u_3^2) & (\chi - \frac{\kappa}{c_v})u_1 & (\mu - \frac{\kappa}{c_v})u_2 & (\mu - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} & 0 \end{bmatrix}$$

$$\mathbf{K}_{12} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\lambda u_2 & 0 & \lambda & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -(\mu + \lambda)u_1 u_2 & \mu u_2 & \lambda u_1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{K}_{13} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\lambda u_3 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ -(\mu + \lambda)u_1 u_3 & \mu u_3 & 0 & \lambda u_1 & 0 \end{bmatrix}$$

$$\mathbf{K}_{21} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -\lambda u_1 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -(\mu + \lambda)u_1 u_2 & \lambda u_2 & \mu u_1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{K}_{22} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 & 0 \\ -\chi u_2 & 0 & \chi & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 & 0 \\ \frac{\kappa}{c_v}(-e + \|\mathbf{u}\|^2) - \chi u_2^2 - \mu(u_1^2 + u_3^2) & (\mu - \frac{\kappa}{c_v})u_1 & (\chi - \frac{\kappa}{c_v})u_2 & (\mu - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} & 0 \end{bmatrix}$$

$$\mathbf{K}_{23} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\lambda u_3 & 0 & 0 & \lambda & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -(\mu + \lambda)u_2 u_3 & 0 & \mu u_3 & \lambda u_2 & 0 \end{bmatrix}$$

$$\mathbf{K}_{31} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\lambda u_1 & \lambda & 0 & 0 & 0 \\ -(\mu + \lambda)u_1 u_3 & \lambda u_3 & 0 & \mu u_1 & 0 \end{bmatrix}$$

$$\mathbf{K}_{32} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\mu u_3 & 0 & 0 & \mu & 0 \\ -\lambda u_2 & 0 & \lambda & 0 & 0 \\ -(\mu + \lambda)u_2 u_3 & 0 & \lambda u_3 & \mu u_2 & 0 \end{bmatrix}$$

$$\mathbf{K}_{33} = 1/\rho \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\mu u_1 & \mu & 0 & 0 & 0 \\ -\mu u_2 & 0 & \mu & 0 & 0 \\ -\chi u_3 & 0 & 0 & \chi & 0 \\ \frac{\kappa}{c_v}(-e + \|\mathbf{u}\|^2) - \chi u_3^2 - \mu(u_1^2 + u_2^2) & (\mu - \frac{\kappa}{c_v})u_1 & (\mu - \frac{\kappa}{c_v})u_2 & (\chi - \frac{\kappa}{c_v})u_3 & \frac{\kappa}{c_v} \end{bmatrix}$$

A.2 Primitive density variables

Primitive density variables are defined by

$$\mathbf{Y} = \begin{Bmatrix} \rho \\ u_1 \\ u_2 \\ u_3 \\ T \end{Bmatrix}.$$

Flux Jacobians

The inviscid flux Jacobians $\tilde{\mathbf{A}}_i$ for primitive density variables are given by

$$\tilde{\mathbf{A}}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u_1 & \rho & 0 & 0 & 0 \\ u_2 & 0 & \rho & 0 & 0 \\ u_3 & 0 & 0 & \rho & 0 \\ e & \rho u_1 & \rho u_2 & \rho u_3 & \rho c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_1 = \begin{bmatrix} u_1 & \rho & 0 & 0 & 0 \\ u_1^2 + (\gamma - 1)c_v T & 2\rho u_1 & 0 & 0 & (\gamma - 1)\rho c_v \\ u_1 u_2 & \rho u_2 & \rho u_1 & 0 & 0 \\ u_1 u_3 & \rho u_3 & 0 & \rho u_1 & 0 \\ u_1[e + (\gamma - 1)c_v T] & \rho(u_1^2 + e) + p & \rho u_1 u_2 & \rho u_1 u_3 & \rho u_1 \gamma c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_2 = \begin{bmatrix} u_2 & 0 & \rho & 0 & 0 \\ u_1 u_2 & \rho u_2 & \rho u_1 & 0 & 0 \\ u_2^2 + (\gamma - 1)c_v T & 0 & 2\rho u_2 & 0 & (\gamma - 1)\rho c_v \\ u_2 u_3 & 0 & \rho u_3 & \rho u_2 & 0 \\ u_2[e + (\gamma - 1)c_v T] & \rho u_1 u_2 & \rho(u_2^2 + e) + p & \rho u_2 u_3 & \rho u_2 \gamma c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_3 = \begin{bmatrix} u_3 & 0 & 0 & \rho & 0 \\ u_1 u_3 & \rho u_3 & 0 & \rho u_1 & 0 \\ u_2 u_3 & 0 & \rho u_3 & \rho u_2 & 0 \\ u_3^2 + (\gamma - 1)c_v T & 0 & 0 & 2\rho u_3 & (\gamma - 1)\rho c_v \\ u_3[e + (\gamma - 1)c_v T] & \rho u_1 u_3 & \rho u_2 u_3 & \rho(u_3^2 + e) + p & \rho u_3 \gamma c_v \end{bmatrix}$$

Diffusivity matrices

The diffusivity matrices $\tilde{\mathbf{K}}_{ij}$ for primitive density variables are given by

$$\tilde{\mathbf{K}}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \chi & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & \chi u_1 & \mu u_2 & \mu u_3 & \kappa \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu u_2 & \lambda u_1 & 0 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & \mu u_3 & 0 & \lambda u_1 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda u_2 & \mu u_1 & 0 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \chi & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & \mu u_1 & \chi u_2 & \mu u_3 & \kappa \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & \mu u_3 & \lambda u_2 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{31} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda u_3 & 0 & \mu u_1 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda u_3 & \mu u_2 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \chi & 0 \\ 0 & \mu u_1 & \mu u_2 & \chi u_3 & \kappa \end{bmatrix}$$

A.3 Primitive pressure variables

Primitive pressure variables are defined by

$$\mathbf{Y} = \begin{Bmatrix} p \\ u_1 \\ u_2 \\ u_3 \\ T \end{Bmatrix}.$$

Flux Jacobians

The inviscid flux Jacobians $\tilde{\mathbf{A}}_i = \mathbf{F}_{i,\mathbf{Y}}$ for primitive pressure variables are given by

$$\tilde{\mathbf{A}}_0 = \begin{bmatrix} \frac{\rho}{p} & 0 & 0 & 0 & -\frac{\rho}{T} \\ \frac{\rho u_1}{p} & \rho & 0 & 0 & -\frac{\rho u_1}{T} \\ \frac{\rho u_2}{p} & 0 & \rho & 0 & -\frac{\rho u_2}{T} \\ \frac{\rho u_3}{p} & 0 & 0 & \rho & -\frac{\rho u_3}{T} \\ \frac{\rho e}{p} & \rho u_1 & \rho u_2 & \rho u_3 & -\frac{\rho e}{T} + \rho c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_1 = \begin{bmatrix} \frac{\rho u_1}{p} & \rho & 0 & 0 & -\frac{\rho u_1}{T} \\ \frac{\rho u_1^2}{p} + 1 & 2\rho u_1 & 0 & 0 & -\frac{\rho u_1^2}{T} \\ \frac{\rho u_1 u_2}{p} & \rho u_2 & \rho u_1 & 0 & -\frac{\rho u_1 u_2}{T} \\ \frac{\rho u_1 u_3}{p} & \rho u_3 & 0 & \rho u_1 & -\frac{\rho u_1 u_3}{T} \\ \frac{\rho u_1 e}{p} + u_1 & \rho(u_1^2 + e) + p & \rho u_1 u_2 & \rho u_1 u_3 & -\frac{\rho u_1 e}{T} + \rho u_1 c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_2 = \begin{bmatrix} \frac{\rho u_2}{p} & 0 & \rho & 0 & -\frac{\rho u_2}{T} \\ \frac{\rho u_1 u_2}{p} & \rho u_2 & \rho u_1 & 0 & -\frac{\rho u_1 u_2}{T} \\ \frac{\rho u_2^2}{p} + 1 & 0 & 2\rho u_2 & 0 & -\frac{\rho u_2^2}{T} \\ \frac{\rho u_2 u_3}{p} & 0 & \rho u_3 & \rho u_2 & -\frac{\rho u_2 u_3}{T} \\ \frac{\rho u_2 e}{p} + u_2 & \rho u_1 u_2 & \rho(u_2^2 + e) + p & \rho u_2 u_3 & -\frac{\rho u_2 e}{T} + \rho u_2 c_v \end{bmatrix}$$

$$\tilde{\mathbf{A}}_3 = \begin{bmatrix} \frac{\rho u_3}{p} & 0 & 0 & \rho & -\frac{\rho u_3}{T} \\ \frac{\rho u_1 u_3}{p} & \rho u_3 & 0 & \rho u_1 & -\frac{\rho u_1 u_3}{T} \\ \frac{\rho u_2 u_3}{p} & 0 & \rho u_3 & \rho u_2 & -\frac{\rho u_2 u_3}{T} \\ \frac{\rho u_3^2}{p} + 1 & 0 & 0 & 2\rho u_3 & -\frac{\rho u_3^2}{T} \\ \frac{\rho u_3 e}{p} + u_3 & \rho u_1 u_3 & \rho u_2 u_3 & \rho(u_3^2 + e) + p & -\frac{\rho u_3 e}{T} + \rho u_3 c_v \end{bmatrix}$$

Diffusivity matrices

The diffusivity matrices $\tilde{\mathbf{K}}_{ij}$ for primitive pressure variables are given by

$$\tilde{\mathbf{K}}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \chi & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & \chi u_1 & \mu u_2 & \mu u_3 & \kappa \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu u_2 & \lambda u_1 & 0 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & \mu u_3 & 0 & \lambda u_1 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda u_2 & \mu u_1 & 0 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \chi & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & \mu u_1 & \chi u_2 & \mu u_3 & \kappa \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{23} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & \mu u_3 & \lambda u_2 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{31} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & \lambda u_3 & 0 & \mu u_1 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda u_3 & \mu u_2 & 0 \end{bmatrix}$$

$$\tilde{\mathbf{K}}_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \chi & 0 \\ 0 & \mu u_1 & \mu u_2 & \chi u_3 & \kappa \end{bmatrix}$$

Appendix B

Jacobi polynomials

An important property of Jacobi polynomials $P_p^{\alpha,\beta}(x)$ is their orthogonality relationship given by

$$\int_{-1}^1 (1-x)^\alpha (1+x)^\beta P_p^{\alpha,\beta}(x) P_q^{\alpha,\beta}(x) dx = C \delta_{pq} \quad (\text{B.1})$$

where $\alpha, \beta > -1$ and C is computed from

$$C = \frac{2^{\alpha+\beta+1}}{2p + \alpha + \beta + 1} \frac{\Gamma(p + \alpha + 1) \Gamma(p + \beta + 1)}{p! \Gamma(p + \alpha + \beta + 1)} \quad (\text{B.2})$$

with $\Gamma(n) = (n-1)!$. This relation implies that $P_p^{\alpha,\beta}(x)$ is orthogonal to all Jacobi polynomials $P_q^{\alpha,\beta}(x)$ of order q less than p with respect to $(1-x)^\alpha(1+x)^\beta$. If $\alpha = \beta = 0$ these polynomials are known as *Legendre* polynomials and if $\alpha = \beta = -\frac{1}{2}$ they are called *Chebyshev* polynomials.

The Jacobi polynomials can be constructed using the following recursive formula

$$\begin{aligned} P_0^{\alpha,\beta}(x) &= 1 \\ P_1^{\alpha,\beta}(x) &= [\alpha - \beta + (\alpha + \beta + 2)x] \\ P_{n+1}^{\alpha,\beta}(x) &= (a_n^2 + a_n^3 x) P_n^{\alpha,\beta}(x) - a_n^4 P_{n-1}^{\alpha,\beta}(x) \end{aligned} \quad (\text{B.3})$$

where

$$\begin{aligned}
 a_n^1 &= 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta) \\
 a_n^2 &= (2n+\alpha+\beta+1)(\alpha^2-\beta^2) \\
 a_n^3 &= (2n+\alpha+\beta)(2n+\alpha+\beta+1)(2n+\alpha+\beta+2) \\
 a_n^4 &= 2(n+\alpha)(n+\beta+1)(2n+\alpha+\beta+2)
 \end{aligned}$$

The derivatives can be computed using either the recursive formula

$$b_n^1(x) \frac{d}{dx} P_{n+1}^{\alpha,\beta}(x) = b_n^2(x) P_n^{\alpha,\beta}(x) + b_n^3(x) P_{n-1}^{\alpha,\beta}(x) \quad (\text{B.4})$$

where

$$\begin{aligned}
 b_n^1(x) &= (2n+\alpha+\beta)(1-x^2) \\
 b_n^2(x) &= n[\alpha-\beta-(2n+\alpha+\beta)x] \\
 b_n^3(x) &= 2(n+\alpha)(n+\beta)
 \end{aligned}$$

or using

$$\frac{d}{dx} P_n^{\alpha,\beta}(x) \frac{1}{2}(\alpha+\beta+n+1) P_{n-1}^{\alpha+1,\beta+1}(x). \quad (\text{B.5})$$

where $P_{n-1}^{\alpha+1,\beta+1}$ follows from B.3.

Appendix C

Modal connectivity and boundary condition enforcement

Here, we illustrate the procedure that is used to obtain \mathcal{C}^0 -continuity as well as the enforcement of boundary conditions. To do so, we define the vector of local degrees of freedom as all the elemental expansion coefficients of each element. For a given element, these are contained within the vector \mathbf{u}^e . The ordering of the degrees of freedom in \mathbf{u}^e is illustrated in Figure C.1 for a two-dimensional expansion. The vector of all the local degrees of freedom can then be written as

$$\mathbf{u}_l = \left\{ \begin{array}{c} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \vdots \\ \mathbf{u}^{n_{el}} \end{array} \right\} \quad (\text{C.1})$$

Additionally, in order to handle Dirichlet boundary conditions we define the vectors \mathbf{u}_D^e containing the values for the local degrees of freedom, where prescribed and zero

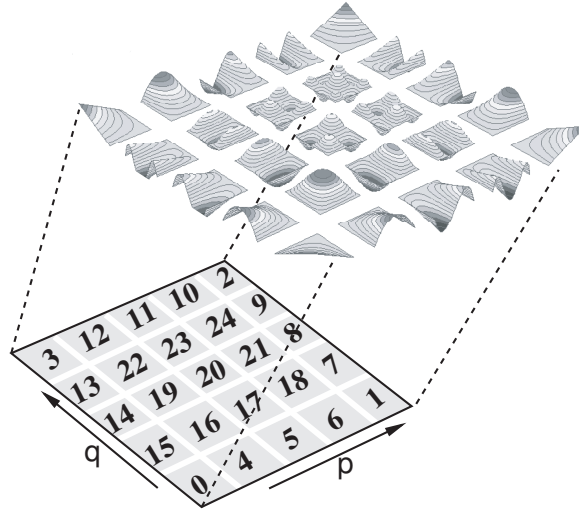


Figure C.1: Illustration of the local numbering scheme on a two-dimensional elemental expansion for $P=5$.

elsewhere. The vector of all Dirichlet conditions can then be written as

$$\mathbf{u}_D = \begin{Bmatrix} \mathbf{u}_D^1 \\ \mathbf{u}_D^2 \\ \vdots \\ \mathbf{u}_D^{n_{el}} \end{Bmatrix} \quad (\text{C.2})$$

If we define \mathbf{u}_g to denote the vector of global degrees of freedom, the vector of all local degrees of freedom can be written as

$$\mathbf{u}_l = \mathbf{u}_D + \mathcal{A}\mathbf{u}_g \quad (\text{C.3})$$

where \mathcal{A} is a very sparse non-square matrix which defines the mapping from global to local degrees of freedom. Depending on the orientation of the local element axes, the non-zero entries of \mathcal{A} are either -1 or 1 . To illustrate the form of the assembly matrix \mathcal{A} we consider the case shown in Figure C.2, where the domain consists of two quadrilateral elements with a Dirichlet condition at the left boundary. In this example the expansion with $P_1 = P_2 = 2$ contains only boundary modes, so that the

number of local degrees of freedom in each element is $n_m = 8$, and the total number of local degrees of freedom is $2 \times n_m = 16$. The local numbering is shown in the left-hand plot of Figure C.2. Note that the edge mode degrees of freedom are listed at the center of the element's edges along which they have non-zero magnitude. To enforce continuity between the two elements, we must couple the boundary modes (1,2,5) of element 0 to (0,3,7) of element 2. This is achieved by assigning the a global numbering scheme as shown in the right-hand plot of Figure C.2. Additionally, the Dirichlet boundary condition is obtained by constraining boundary modes (0,3,7) of element 0 to their prescribed values. Equation C.3 can then be written as

$$\begin{pmatrix} u_0^0 \\ u_1^0 \\ u_2^0 \\ u_3^0 \\ u_4^0 \\ u_5^0 \\ u_6^0 \\ u_7^0 \\ \dots \\ u_0^1 \\ u_1^1 \\ u_2^1 \\ u_3^1 \\ u_4^1 \\ u_5^1 \\ u_6^1 \\ u_7^1 \end{pmatrix} = \begin{pmatrix} u_0^0 \\ 0 \\ 0 \\ u_3^0 \\ 0 \\ 0 \\ 0 \\ u_7^0 \\ \dots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_4 \\ u_5 \\ u_6 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \end{pmatrix} \quad (\text{C.4})$$

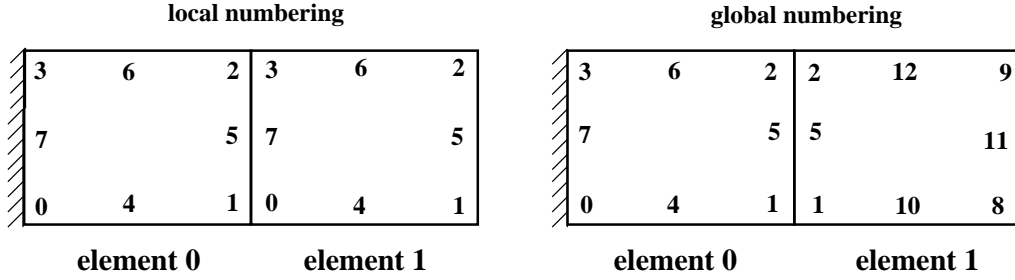


Figure C.2: Local and global degrees of freedom numbering scheme for a two-dimensional expansion of order $P_1 = P_2 = 2$. The edge modes are listed along the center of the edge on which they have nonzero magnitude.

Note that every column of \mathcal{A} contains at least one entry, and the summation of each column equals the number of modes that contribute to the corresponding global mode. This is called the multiplicity of the global mode [38]. The mapping from local to global (assembly) is obtained by taking the transpose of \mathcal{A} , so that

$$\mathbf{v}_g = \mathcal{A}^T \mathbf{v}_l \quad (\text{C.5})$$

where \mathbf{v}_g and \mathbf{v}_l are associated with integral operations. Here we use \mathbf{v} rather than \mathbf{u} to stress that \mathcal{A} and \mathcal{A}^T are not the inverse of each other, i.e. $\mathbf{u}_g \neq \mathcal{A}^T \mathcal{A} \mathbf{u}_g$. The operations defined by \mathcal{A} (scatter from global to local) and \mathcal{A}^T (assembly from local to global) are all that is necessary to construct the global system. Consider the finite-element variational formulation for a given problem that produces, at the element level, the following system of equations

$$\mathbf{M}^e \mathbf{u}^e = \mathbf{f}^e \quad (\text{C.6})$$

where \mathbf{M}^e is the element matrix and \mathbf{f}^e the corresponding right-hand vector. The elemental systems of equations of all elements can be represented by the following system

$$\mathbf{M} \mathbf{u}_l = \mathbf{f}_l \quad (\text{C.7})$$

where \mathbf{u}_l is defined by C.1, and

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}^1 & 0 & 0 & 0 \\ 0 & \mathbf{M}^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{M}^{n_{el}} \end{bmatrix}, \text{ and } \mathbf{f}_l = \begin{Bmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \\ \vdots \\ \mathbf{f}^{n_{el}} \end{Bmatrix}. \quad (\text{C.8})$$

Up to now, there is no notion of continuity or boundary conditions in system (C.8). To enforce these, (C.3) is substituted into (C.8), so that

$$\mathbf{M}(\mathbf{u}_D + \mathcal{A}\mathbf{u}_g) = \mathbf{f}_l \quad (\text{C.9})$$

$$\Rightarrow \mathbf{M}\mathcal{A}\mathbf{u}_g = \mathbf{f}_l - \mathbf{M}\mathbf{u}_D \quad (\text{C.10})$$

Note that the known Dirichlet values contained in \mathbf{u}_D are transferred to the right-hand side. The resulting system is not square, since \mathcal{A} is a non-square matrix. Therefore, (C.10) is pre-multiplied by \mathcal{A}^T to give the square system of equations

$$\mathcal{A}^T\mathbf{M}\mathcal{A}\mathbf{u}_g = \mathcal{A}^T(\mathbf{f}_l - \mathbf{M}\mathbf{u}_D) \quad (\text{C.11})$$

This system must be solved using solution techniques for linear systems of equations, as described in Section 4.6. In practice, the matrix \mathbf{M} is never formed explicitly, as this would require much more memory than the final system matrix $\mathcal{A}^T\mathbf{M}\mathcal{A}$. Instead, the required operations are performed at the element level, that is $\mathcal{A}_e^T\mathbf{M}^e\mathcal{A}_e$ and $\mathcal{A}_e^T(\mathbf{f}^e - \mathbf{M}^e\mathbf{u}_D^e)$.

Samenvatting

De Variational Multiscale (VMS) methode heeft zich ontwikkeld als een veel belovende nieuwe aanpak voor Large Eddy Simulatie (LES) van turbulente stromingen. Het voordeel van de VMS methode is de mogelijkheid om verschillende aannames te maken voor het effect van de subgrid-schalen op de verschillende schalen in de berekening. Een typische keuze in de VMS methode is de expliciete modelering van het effect van de subgrid-schalen te beperken tot de kleinste berekende schalen, zodat de dynamisch belangrijke grote berekende schalen vrij blijven van de directe invloed van het subgrid-schaal (SGS) model. Voorgaande implementaties van de VMS methode zijn beperkt gebleven tot incompressible formulaties op simpele geometrieën en/of kleine tijdstappen vanwege het gebruik van expliciete tijdsdiscretizatie methoden. In dit onderzoek stellen we een ruimte-tijd VMS methode voor die tracht de hierboven beschreven beperkingen te overkomen. In het bijzonder, we beogen een methode te ontwikkelen die toepasbaar is voor complexe geometrieën met een minimaal aantal vrijheidsgraden en die tijdstappen toelaat die gekozen zijn op basis van de van belang zijnde fysische fenomenen in plaats van stabiliteits criteria.

De ruimtelijke discretizatie van de voorgestelde methode is een hogere orde nauwkeurige continue Galerkin methode, die vanwege zijn hierarchische aard een natuurlijk raamwerk voor schaal decompositie vormt, hetgeen cruciaal is voor de VMS methode. Ruimte-tijd eindige elementen methoden kunnen zowel continu als discontinu in de tijd zijn. Discontinue methoden in de tijd bieden grote flexibiliteit

voor adaptatie, maar kunnen leiden tot hoge berekeningskosten. Continue methoden in de tijd daarentegen, bieden mogelijk een goed compromis tussen nauwkeurigheid en berekeningskosten. In dit onderzoek beschouwen we drie verschillende tijdsdiscretizaties, namelijk een eerste orde continue Galerkin methode (TCG), een tweede orde continue Petrov-Galerkin methode (TCPG) en een derde orde discontinue Galerkin methode (TDG).

We beschouwen de prestaties van de ruimtelijke VMS discretizatie voor de simulatie van een volledig ontwikkelde turbulente kanaalstroming. We laten zien dat de methode leidt tot verminderde resolutie eisen vergeleken met een traditionele methode waarin hetzelfde SGS model direct werkt op alle berekende schalen in de simulatie. De cruciale parameter voor het verkrijgen van nauwkeurige lage orde turbulentie statistieken met de VMS methode is de partitie tussen de grote en kleine berekende schalen. In het bijzonder, we laten zien dat wanneer simpele "eddy-viscosity" SGS modellen worden gebruikt, de basis functies die in staat zijn de fundamentele dynamica van de coherente structuren aan de wand te representeren, vrij moeten zijn van de directe invloed van het SGS model.

Aangezien ruimte-tijd methoden noodzakelijk tot een impliciete tijdsdiscretizatie leiden, is het een uitdaging de berekingen uit voeren binnen een acceptabele tijd met acceptabele berekeningskosten. Daarom hebben we een gedetailleerde prestatie analyse uitgevoerd om de factoren te onderzoeken die de nauwkeurigheid en de simulatie kosten van de voorgestelde methoden beïnvloeden. Voor dit doeleinde gebruiken we weer de turbulente kanaalstroming.

Eerst beschouwen we de verschillende tijdsdiscretizaties. Het blijkt dat de TCG methode geen competitieve methode voor tijdsstap grootte waarin we geïnteresseerd zijn. In tegenstelling, de TDG en TCPG methode produceren nauwkeurige en vergelijkbare resultaten voor relatief grote tijdsstappen. Echter, de berekeningskosten van de TDG methode zijn aanzienlijk hoger dan de TCPG methode, aangezien het twee maal zo veel vrijheidsgraden gebruikt. Daarom, in het geval dat geen adaptatie

vereist is, geniet de TCPG methode de voorkeur.

Vervolgens vergelijken we de nauwkeurigheid en berekeningskosten voor verschillende ruimtelijke hp -resoluties voor een gelijk totaal aantal vrijheidsgraden. We laten zien dat de hogere orde methoden tot een hogere nauwkeurigheid leiden vergeleken met een standaard lagere orde (lineaire) eindige elementen methode, die niet geschikt is voor de huidige VMS formulatie. Echter, de hogere orde methoden zijn inherent meer kostbaar aangezien de te verrichten operaties gedurende de simulatie kwadratisch schaalt met het aantal basisfuncties, terwijl dit slechts linear schaalt met het aantal elementen. Hogere orde methoden leiden ook tot significant vollere matrices, hetgeen resulteert in snel toenemende geheugen eisen met de orde van de methode. Aangezien de berekeningskosten van hogere orde methoden nog steeds relatief hoog zijn, doen we suggesties voor aanvullende onderzoeksrichtingen om de efficiëntie van de voorgestelde methode te verbeteren.

Publications

E.A. Munts, S.J. Hulshoff, G.N. Wells, H. Bijl, R. de Borst. *A multiscale method for linear convection-diffusion using partition of unity*. Fifth World Congress on Computational Mechanics (WCCM V), 2002, Vienna, Austria.

E.A. Munts, S.J. Hulshoff, R. de Borst. *The partition-of-unity method for linear diffusion and convection problems: accuracy, stabilization and multiscale interpretation*. International Journal for Numerical Methods in Fluids 2003; **43**:199-213.

S.J. Hulshoff, E.A. Munts, R. de Borst. *A Comparison of Space-Time Variational Multiscale Discretizations*. Third International Conference on Computational Fluid Dynamics, 2004, Toronto, Canada.

E.A. Munts, S.J. Hulshoff, R. de Borst. *A Space-Time Variational Multiscale Discretization for LES*. AIAA paper 2004-2132. 34th AIAA Fluid Dynamics Conference, 2004, Portland, OR, USA.

E.A. Munts, S.J. Hulshoff. *Space-time multiscale methods for turbulence simulation*. Submitted to Physics of Fluids, 2006.

Curriculum Vitae

2001 – 2006 PhD student at the Delft University of Technology,
Faculty of Aerospace Engineering.

1994 – 2001 Master degree in Aerospace Engineering,
Delft University of Technology.

1988 – 1994 Grammar School.

Oct. 6th 1975 Born in Dordrecht, The Netherlands.